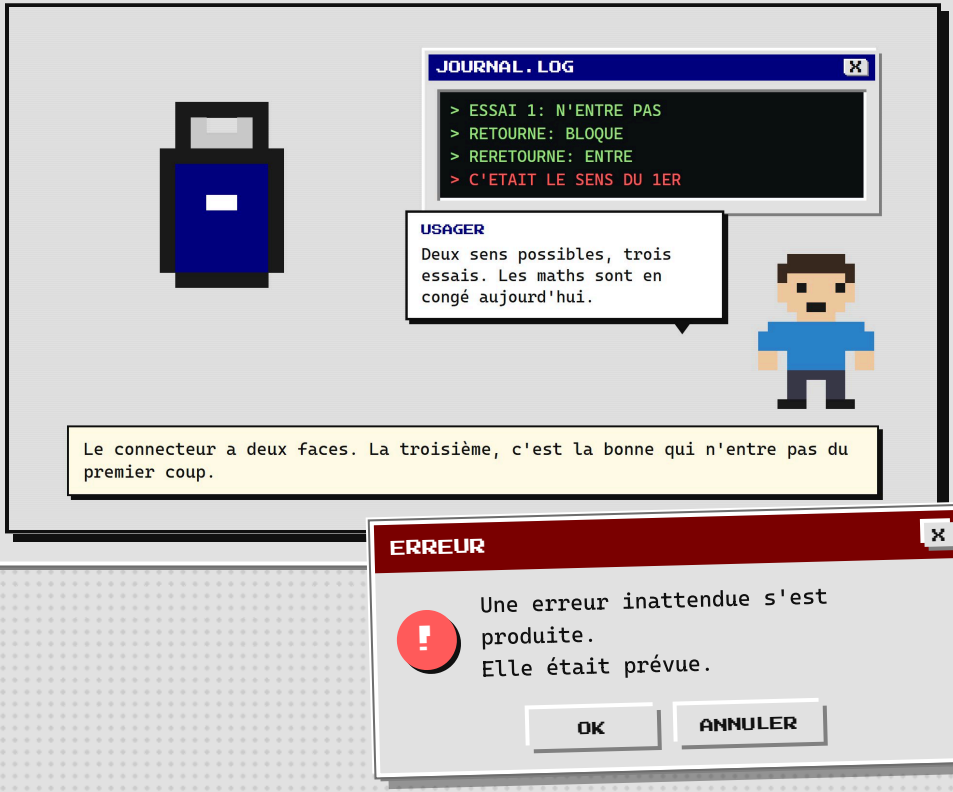
LEGGI_DI_MURPHY.EXE

MURPHY.EXE

Les lois de Murphy de l'informatique, commentées par un admin système qui les a vues à l'œuvre



C:\MURPHY> lire.exe /toutes /commentees
133 lois trouvées. 0 erreur. (pour l'instant)_

SILVANO DE POLI

133 LOIS + COROLLAIRES MAISON

MURPHY.EXE

Les lois de Murphy de l'informatique, commentées par un admin système qui les a vues à l'œuvre

SILVANO DE POLI

À toutes les imprimantes qui m'ont
appris l'humilité.
Et à ceux qui ont fait leur sauvegarde
hier soir : vous savez pourquoi.

SOMMAIRE

Préface 5
Mode d'emploi 6
1. Les lois fondamentales 7
2. Ferraille : imprimantes, câbles et disques qui écoutent 23
3. Helpdesk : utilisateurs, mots de passe, mystères 45
4. Réseaux et Internet : le fil invisible 63
5. La chasse au bug 81
6. L'art d'écrire du code (et de le regretter) 101
7. Projets, réunions et clients 121
8. Progrès, mises à jour et intelligence artificielle 143
Les corollaires maison 167
Envoi 179
Colophon 180

PREFACE

Je m'appelle Silvano De Poli et mon métier, c'est de réparer des choses qui se sont cassées toutes seules. Je suis administrateur système et technicien PC, je construis des sites, je travaille avec l'intelligence artificielle et je dirige l'agence web RetroAI. En clair : quand quelque chose cesse de marcher au bureau, le téléphone qui sonne, c'est en général le mien.

Au fil des années, j'ai appris que les pannes ne sont pas dues au hasard. Elles en ont l'air, mais elles ont leurs habitudes. Elles préfèrent le vendredi. Elles adorent les démonstrations aux clients. Elles se cachent quand le technicien arrive et réapparaissent dès que la machine redémarre. Un jour, devant une imprimante qui s'était coincée pile sur la dernière feuille de papier à en-tête, j'ai compris que je n'étais pas malchanceux. J'étais dans une loi.

Cette loi, et bien d'autres, existaient déjà. Edward Murphy, ingénieur, a donné son nom à la plus célèbre : si une chose peut mal tourner, elle tournera mal. Après lui, programmeurs, économistes, écrivains et chercheurs ont ajouté leurs observations, certaines très sérieuses, d'autres nées comme des blagues et devenues proverbes. L'informatique les a toutes recueillies, avec le soin qu'elle met à garder les câbles dans le tiroir : on ne sait jamais.

Ce recueil est né sur mon site, une loi à la fois. Au début, c'étaient des notes : la loi, son auteur, un commentaire. Puis les commentaires ont grandi, les thèmes sont arrivés, certaines lois sont même devenues de petites vidéos en pixel art. À un moment, il y en avait cent trente-trois, et je me suis aperçu qu'ensemble elles racontaient une histoire : celle de la relation, tendre et compliquée, entre les humains et les machines.

Le livre contient trois sortes de lois. Il y a celles **avec un nom propre**, liées à une personne réelle : Moore, Brooks, Parkinson, Hofstadter et bien d'autres. Pour celles-là, j'ai essayé d'expliquer ce qu'elles disent vraiment avant de m'en moquer. Il y a les **anonymes**, transmises de bureau en bureau comme des recettes de famille, que personne n'a signées parce que tout le monde les a vécues. Et il y a quelques **corollaires de mon cru**, surtout sur l'intelligence artificielle, que je signe avec un certain embarras et beaucoup de fierté.

Chaque loi a son commentaire. Le ton est celui du site : ironique, parfois un peu surréaliste, toujours affectueux. Les anecdotes les plus absurdes sont, je vous le garantis, inventées. Les plus crédibles aussi, mais je le dis avec moins d'assurance. Les explications techniques, elles, sont vraies, et j'ai essayé de les rendre compréhensibles même pour qui n'a jamais ouvert un ordinateur, sauf pour y chercher une vis.

À la fin de chaque chapitre, il y a *Le conseil du sysadmin*. C'est la partie sérieuse, celle que j'aimerais que vous gardiez en poche : des choses pratiques, éprouvées, vraiment utiles. Elles ne battent pas Murphy, parce que Murphy ne se bat pas. Mais elles l'obligent à travailler davantage.

Le livre se lit dans l'ordre ou au hasard. On peut le garder sur le bureau comme un bréviaire et l'ouvrir avant une mise à jour importante. On peut le citer en réunion pour paraître sage, ou l'offrir au collègue qui répète sans cesse « ça marche sur mon PC ». Ça marche aussi. Sur son PC.

Un dernier avertissement. Pendant que je mettais ces pages en ordre, l'ordinateur s'est figé deux fois, le Wi-Fi a disparu au milieu d'une recherche importante et l'imprimante a refusé de collaborer à l'épreuve d'impression. Je ne le raconte pas pour me plaindre. Je le joins comme certificat d'authenticité. Les lois fonctionnent. Bonne lecture, et enregistrez souvent.

MODE D'EMPLOI

Mode d'emploi, modèle MURPHY.EXE. Lire attentivement avant la mise sous tension.

- **Démarrage.** Le livre s'ouvre à n'importe quelle page. Aucun redémarrage requis.
- **Ordre de lecture.** Les chapitres vont du plus universel au plus technique. Qui ne programme pas peut sauter le cinquième et le sixième sans dommages permanents, mais en les lisant, il comprendra enfin son programmeur.
- **Posologie.** Une loi par jour, de préférence avant de toucher à un serveur. En cas de surdosage, des citations involontaires peuvent apparaître en réunion.
- **Conseils du sysadmin.** À la fin de chaque chapitre. Ce sont les seules parties sérieuses du livre : à appliquer, pas seulement à lire.
- **Compatibilité.** Fonctionne à l'écran, sur papier et à voix haute. Testé aussi à l'impression, avec des résultats variables.
- **Garantie.** Le livre n'empêche pas les pannes. Il les rend seulement plus reconnaissables. En cas de dysfonctionnement, essayer de relire.

01

CHAPITRE 1

CHAP 01 — FONDATIONS

Les lois fondamentales

Chaque religion a ses textes sacrés, et l'informatique a ces lois. Personne ne les a gravées dans la pierre : elles ont été tirées du terrain, un fiasco à la fois, par des ingénieurs, des programmeurs, des économistes et des gens qui voulaient juste imprimer un formulaire.

On commence par la loi mère, celle de Murphy, qui à l'origine était un conseil de conception et qui est devenue notre horoscope quotidien. Suivent les proches parentes : celle qui ajoute l'heure du désastre, celle qui rappelle que l'ordinateur obéit à la lettre, celle qui soutient qu'un programme qui marche du premier coup prend juste son élan.

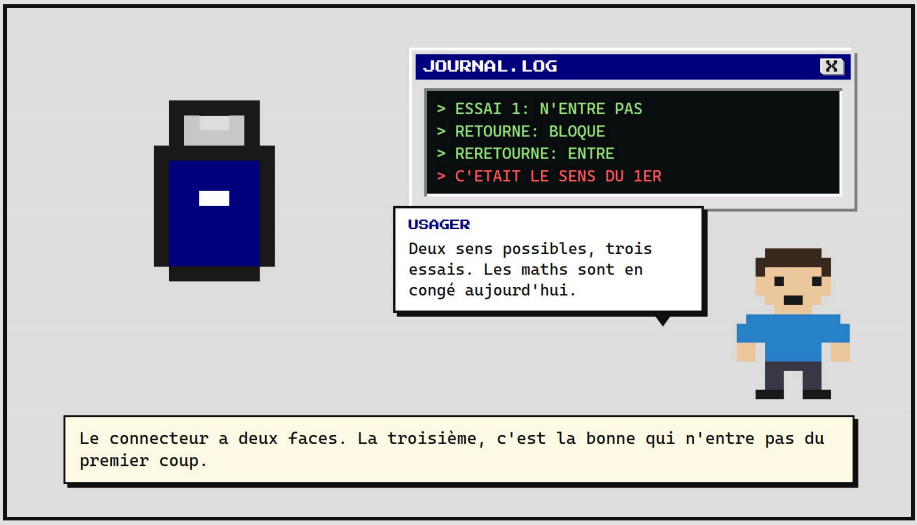
Au milieu, il y a aussi des lois nées hors de l'informatique, dans la bureaucratie, l'économie, la science-fiction. Nous les avons adoptées parce qu'elles nous décrivaient trop bien : le travail qui s'étire jusqu'à remplir le temps, le collègue promu jusqu'au mauvais niveau, les quatre-vingts pour cent des ennuis qui viennent toujours des mêmes vingt pour cent.

C'est le chapitre à lire en premier et à citer en réunion. Ces lois fonctionnent comme les locutions latines : qui les prononce paraît aussitôt plus sage, même si le serveur est toujours en panne.

CONTENU.DIR

01 Loi de Murphy
02 Loi de Finagle
03 Loi de l'ordinateur littéral
04 Loi du bug résiduel
05 Loi du succès suspect
06 Loi de Hofstadter
07 Loi de Parkinson
08 Principe de Peter
09 Principe de Pareto (80/20)
10 Rasoir d'Hanlon
11 Troisième loi de Clarke
12 Loi de Sturgeon
13 Loi de Moore

LOI DE MURPHY



Si une chose peut mal tourner, tôt ou tard elle tournera mal. Cela vaut surtout pour les connecteurs qui se branchent dans les deux sens.

— Edward A. Murphy Jr., *Base aérienne d'Edwards, projet MX981, vers 1949 ; formulation exacte et circonstances contestées*

Tout part d'un ingénieur, de capteurs montés à l'envers et d'une expérience ratée. À l'origine, ce n'était pas du pessimisme : c'était un conseil de conception. Si une pièce peut être mal montée, quelqu'un le fera, donc concevez-la pour que ce soit impossible.

Nous, les informaticiens, l'avons prise comme une prophétie. Le câble USB a trois faces : la bonne, la mauvaise, et la bonne qui pourtant n'entre pas au premier essai. C'est la physique quantique des ports qui le dit.

Moi, je l'applique avec une rigueur bureaucratique. Avant chaque mise à jour, je remplis un formulaire : « Choses qui peuvent mal tourner ». Le formulaire fait quarante pages. La mise à jour plante à la page quarante et un, que je n'avais pas prévue.

LOI DE FINAGLE



Tout ce qui peut mal tourner tournera mal, et choisira le pire moment possible pour le faire.

— Anonyme, *Variante de la loi de Murphy répandue dans les milieux de la science-fiction américaine dans les années 50-60 ; attribution incertaine*

Murphy dit que la catastrophe arrivera. Finagle ajoute l'horaire : vendredi à 17 h 58, pendant que vous éteignez déjà l'écran. C'est la version avec l'agenda synchronisé.

Le serveur ne tombe jamais en panne le mardi matin, quand vous êtes reposé et que vous avez votre café. Il tombe pendant la démo au client, le jour de la clôture des comptes, ou quand vous êtes au mariage de votre cousine avec l'ordinateur portable fermé dans la voiture. La panne a un agenda, et le vôtre est toujours consultable.

J'ai essayé de la piéger : j'ai fait semblant de partir en vacances. Le serveur est resté impassible. Puis, le jour où je suis vraiment rentré, l'alimentation a pris feu. Je ne sais pas comment il le savait. Je soupçonne l'agenda partagé.

LOI DE L'ORDINATEUR LITTERAL

NETTOYAGE_CLIENTS

```
supprime clients où:
   derniere_commande avant 2020
   ou 1 = 1

> 4 218 clients supprimés
> Autre chose ? :)
```

Une erreur d'une ligne. L'ordinateur a exécuté avec un zèle de fonctionnaire.

L'ordinateur fait exactement ce que vous lui avez dit de faire, jamais ce que vous vouliez qu'il fasse.

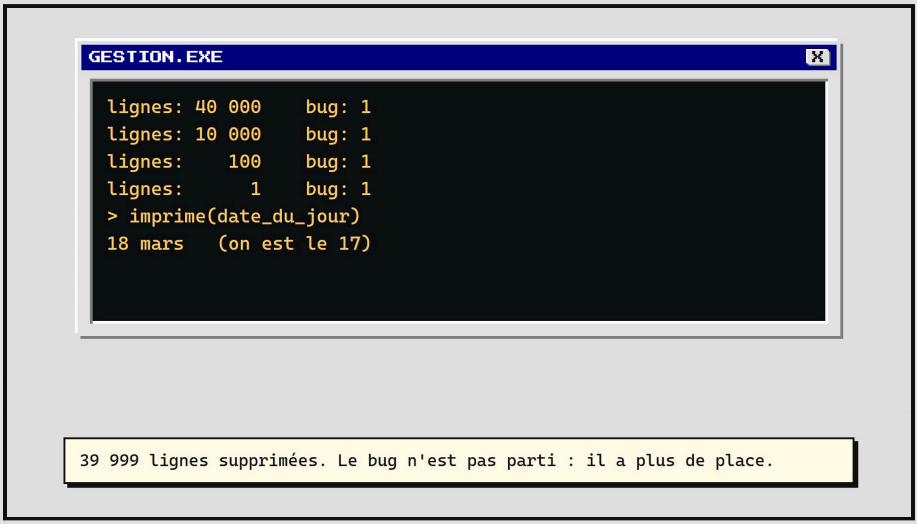
— Anonyme, *folklore informatique*

L'ordinateur est l'employé le plus obéissant de l'histoire et, justement pour cela, le plus dangereux. Il n'interprète pas, il ne devine pas, il ne vous regarde pas dans les yeux pour comprendre ce que vous vouliez dire. Il exécute. À la lettre. Avec un zèle bureaucratique.

Si vous lui dites « supprime les clients inactifs » et que, à cause d'une virgule mal placée, les clients inactifs sont tous les clients, il les supprime tous, avec une efficacité admirable. Puis il vous demande si vous désirez autre chose. Charmant.

C'est comme avoir un génie de la lampe qui a fait son droit. Chaque vœu doit être formulé avec des clauses, des exceptions et des notes en marge. Pour les non-programmeurs, voilà la véritable essence du métier : non pas expliquer à l'ordinateur quoi faire, mais prévoir toutes les façons créatives dont il pourrait vous mal comprendre. Moi, par sécurité, je parle désormais comme ça à mon chien aussi.

LOI DU BUG RESIDUEL



Tout programme contient au moins un bug, et tout programme peut être raccourci d'une instruction. Il s'ensuit, par récurrence, que tout programme peut être réduit à une seule instruction qui ne marche pas.

— Anonyme, *folklore informatique*

C'est le théorème fondamental de la programmation et, comme tous les théorèmes sérieux, personne ne l'a jamais démontré, parce que tout le monde était occupé à corriger autre chose. La logique est implacable : on enlève une ligne à la fois, et le bug est toujours là, accroché comme une moule à son rocher.

J'ai essayé de le vérifier en laboratoire. J'ai pris un logiciel de gestion de quarante mille lignes et j'ai commencé à enlever. Arrivé à une seule ligne, le programme affichait la date. Fausse. D'un jour. Seulement les mois en « r ».

Pour ceux qui ne programment pas, la morale est simple : le logiciel parfait existe, mais seulement tant que personne ne l'écrit. Dès que vous appuyez sur la première touche, l'erreur s'installe au premier rang et prend des notes.

LOI DU SUCCES SUSPECT

TERMINAL

```
> compile projet
0 erreur, 0 alerte
> lance
tout ok. temps: 0,01 seconde

(trop rapide. trop ok.)
```

Le programmeur aguerri ne jubile pas. Il ferme la porte et revérifie tout.

Si un programme marche du premier coup, ce n'est pas bon signe : cela veut dire que l'erreur est plus profonde et qu'elle se manifestera au pire moment.

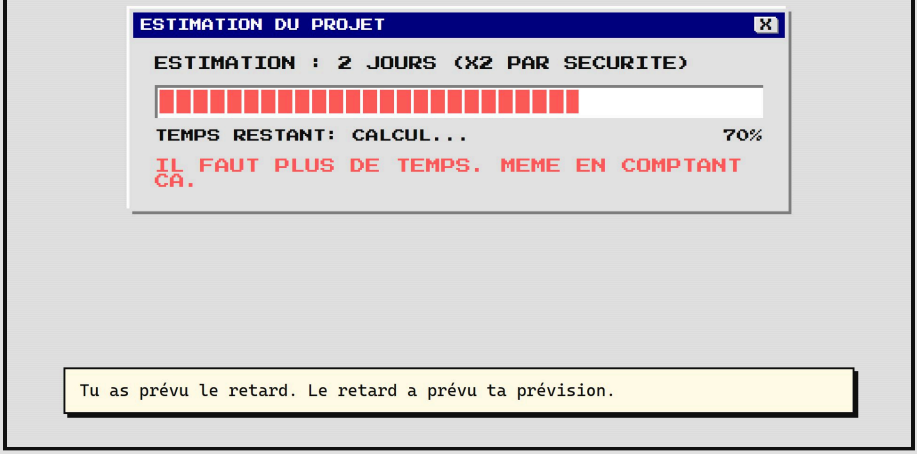
— Anonyme, *folklore informatique*

Un programmeur expérimenté, quand le code tourne tout de suite sans erreur, ne se réjouit pas. Il pâlit. Il se lève, ferme la porte et revérifie tout, comme un démineur devant un colis qui fait un peu trop peu tic-tac.

La raison est statistique. Le logiciel est un champ de mines où chaque pas est faux au moins une fois. Si rien n'a explosé, cela signifie seulement que la mine est plus loin, plus grosse, et qu'elle attend le client.

Ça m'est vraiment arrivé une fois : compilé, lancé, parfait. J'ai fêté ça avec un café de la machine. Trois mois plus tard, j'ai découvert que le programme marchait parce qu'il ne faisait rien. Il sautait toute la partie importante avec une grande élégance. Depuis, quand quelque chose réussit du premier coup, je le traite comme un invité trop aimable : avec affection, et en comptant les petites cuillères.

LOI DE HOFSTADTER



Il faut toujours plus de temps que prévu, même quand on a tenu compte de la loi de Hofstadter dans ses prévisions.

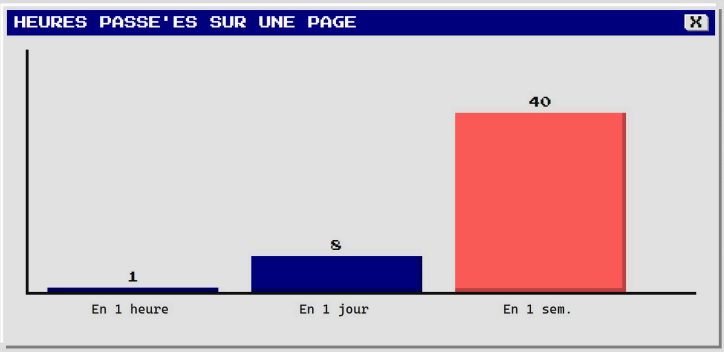
— Douglas Hofstadter, *Gödel, Escher, Bach*, 1979

C'est une loi récursive, et donc très appréciée des informaticiens. Vous estimez deux jours. Vous savez que vous vous trompez toujours, alors vous écrivez quatre. Mais la loi sait que vous le savez, et il en faut huit.

C'est la barre de progression de la vie : « Temps restant : 2 minutes. » Puis 47 minutes. Puis 3 secondes. Puis « Calcul du temps restant en cours », qui est la façon élégante de dire « je n'en ai aucune idée ».

J'ai demandé à une intelligence artificielle d'estimer la durée de mon projet en appliquant la loi de Hofstadter. Elle a répondu : « Cela prendra plus de temps que prévu. » J'ai demandé combien de plus. Elle réfléchit encore. Je lui ai fixé un délai de dix minutes, en tenant compte de la loi. Je reviens mardi.

LOI DE PARKINSON



La page est identique. Celle d'une semaine a changé de police quatorze fois.

Le travail s'étend jusqu'à occuper tout le temps disponible pour l'accomplir. En informatique s'ajoute un corollaire : les données s'étendent jusqu'à remplir tout l'espace disponible.

— Cyril Northcote Parkinson, *The Economist, novembre 1955 (corollaire sur les données : anonyme, attribution incertaine)*

Si vous avez une semaine pour écrire une page, vous y mettrez une semaine. Si vous avez une heure, vous y mettrez une heure. La page sera identique. Dans la version d'une semaine, la police aura juste été changée quatorze fois.

Né comme une satire de la bureaucratie britannique, le principe est devenu le manuel officieux des projets informatiques. Les délais généreux ne produisent pas de meilleurs logiciels : ils produisent de meilleures réunions, c'est-à-dire plus longues. Et ça vaut pour les choses autant que pour les personnes. Vous achetez un disque de quatre téraoctets et en six mois il est plein, comme un garage neuf.

Dans mon bureau, nous avons essayé de vaincre la loi en donnant de faux délais, plus courts que les vrais. Ça a marché un mois. Puis tout le monde a commencé à calculer le vrai délai, et nous avons maintenant un tableur qui l'estime. Nous y travaillons depuis trois semaines.

PRINCIPE DE PETER

CARRIE'RE D'UN PROGRAMMEUR

1. programmeur : excellent
2. chef de projet : assez bon
3. chef de service : incompétent
4. promotions : suspendues

Le système a trouvé son équilibre.

Dans une hiérarchie, chaque personne tend à être promue jusqu'à atteindre son niveau d'incompétence, et elle s'y arrête.

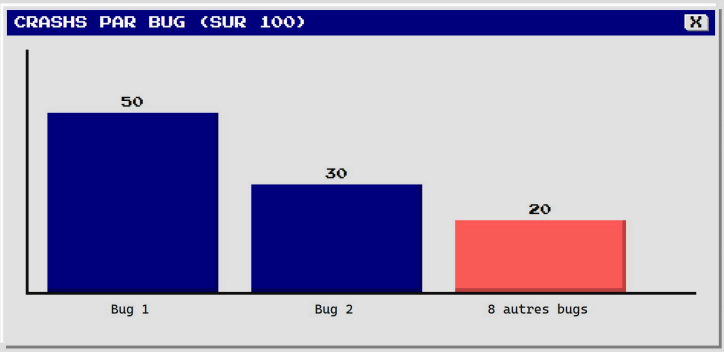
— Laurence J. Peter (avec Raymond Hull), *« The Peter Principle », 1969*

Il était une fois un programmeur excellent. Si excellent qu'on le promut chef de projet. Là, il était moins bon, mais assez pour être promu responsable de service. Et là, enfin, il était totalement incapable. Alors on cessa de le promouvoir. Le système avait trouvé son équilibre.

Le principe de Peter explique pourquoi, dans beaucoup d'entreprises, celui qui écrivait le meilleur code passe aujourd'hui ses journées à faire des tableaux sur les tableaux. Savoir faire une chose ne signifie pas savoir diriger ceux qui la font. Mais la promotion récompense la première compétence pour attribuer un travail qui demande la seconde.

J'ai évité le problème de façon géniale : je me suis mis à mon compte. Ainsi, je suis à la fois programmeur, admin sys, commercial et directeur général. J'ai atteint mon niveau d'incompétence dans quatre rôles à la fois. C'est une optimisation.

PRINCIPE DE PARETO (80/20)



Deux bugs sur dix font huit plantages sur dix. Les vingt pour cent bossent dur.

Dans de nombreux phénomènes, environ 80 % des effets proviennent de 20 % des causes : par exemple, 80 % des problèmes viennent d'une petite partie des bugs ou des clients.

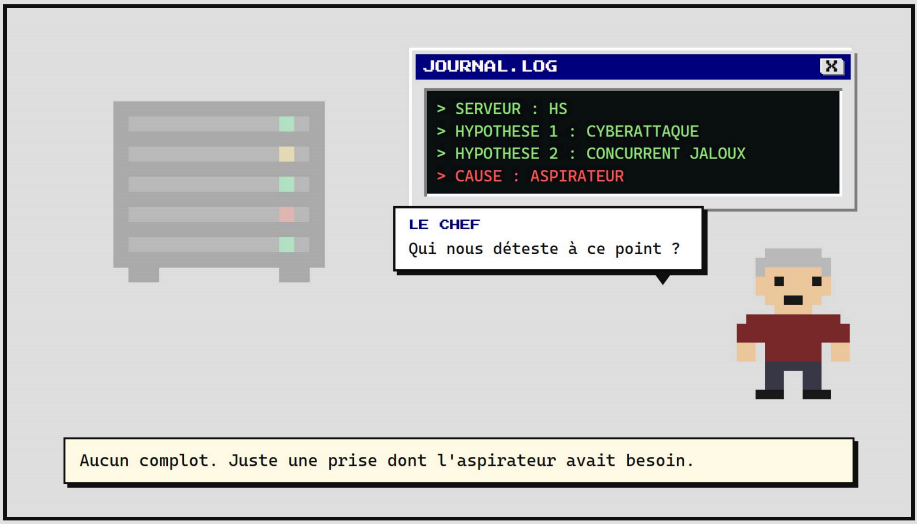
— Vilfredo Pareto, généralisé par Joseph M. Juran, *Observation de Pareto sur la répartition de la richesse (1896) ; nom et généralisation de Juran (années 40-50)*

Un économiste italien remarqua que peu de gens possédaient une grande partie de la terre. Des décennies plus tard, un expert de la qualité généralisa : peu de facteurs causent la majeure partie des effets. Depuis, le quatre-vingts et le vingt forment le couple le plus célèbre du conseil en entreprise.

En informatique, ça marche de façon presque gênante. Quatre-vingts pour cent des plantages viennent d'une poignée de bugs. Quatre-vingts pour cent des appels à l'assistance viennent des mêmes vingt pour cent d'utilisateurs, et vous savez très bien comment ils s'appellent. Quatre-vingts pour cent de la valeur d'un programme tient dans quelques fonctions, justement celles cachées dans le troisième sous-menu.

J'ai appliqué Pareto à mon bureau. Vingt pour cent des câbles produisent quatre-vingts pour cent des nœuds. Le problème, c'est que je ne sais pas quels vingt. Je les ai tous débranchés. Maintenant, c'est rangé, mais plus rien ne s'allume.

RASOIR D'HANLON



N'attribuez pas à la malveillance ce qui s'explique suffisamment par la distraction ou l'incompétence.

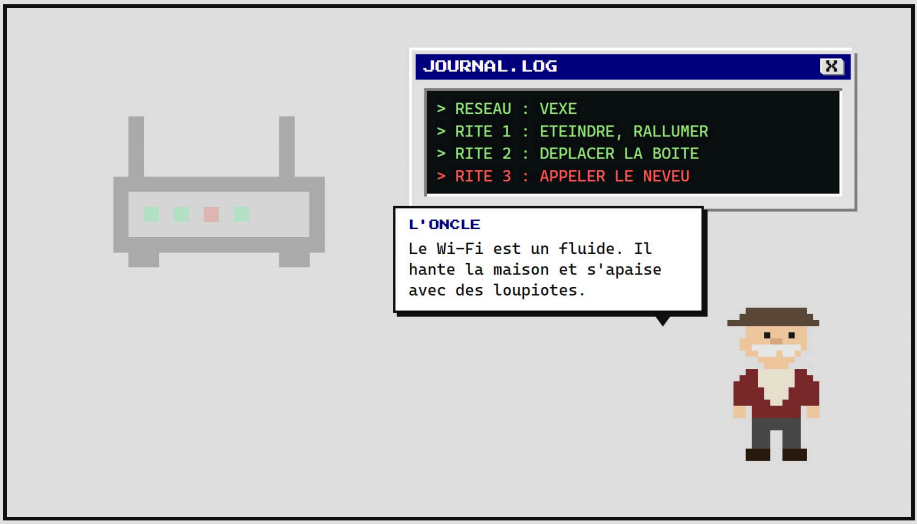
— Robert J. Hanlon, *Arthur Bloch, « Murphy's Law Book Two », 1980 (concepts similaires dans des textes bien antérieurs)*

Le serveur est en panne. Première pensée : cyberattaque, espions étrangers, un concurrent jaloux. Puis on découvre que la dame du ménage a débranché la prise pour brancher l'aspirateur. Aucun complot. Juste une prise libre en moins.

Le rasoir d'Hanlon est le meilleur ami du technicien d'assistance. Quand un utilisateur supprime le dossier partagé, il ne sabote pas l'entreprise. Il a seulement cliqué avec détermination sur quelque chose qu'il ne comprenait pas. Ce qui est pire, mais au moins ce n'est pas personnel. Le garder à l'esprit préserve les relations humaines et la tension artérielle.

Un conseil : le rasoir marche aussi dans l'autre sens. Quand c'est vous qui vous trompez, espérez que les autres le connaissent. Je le cite toujours aux clients avant de leur dire ce que j'ai fabriqué. Ça marche comme une sorte d'antivirus diplomatique.

TROISIEME LOI DE CLARKE



Toute technologie suffisamment avancée est indiscernable de la magie.

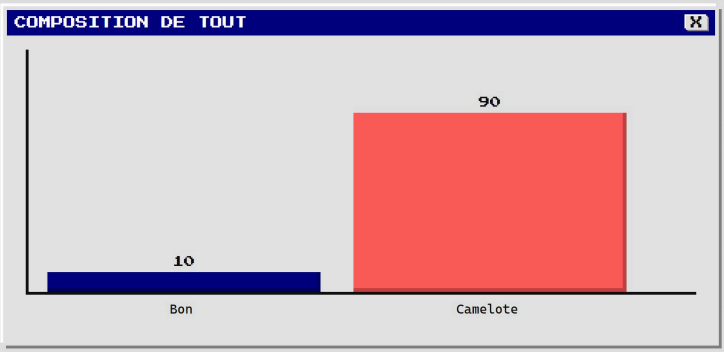
— Arthur C. Clarke, *« Profiles of the Future », édition révisée de 1973 (déjà dans une lettre à Science de 1968)*

Un écrivain de science-fiction observa qu'une technologie assez avancée, pour qui ne sait pas comment elle fonctionne, ressemble à un sortilège. Il pensait à des civilisations extraterrestres. Il n'avait pas vu ma tante devant le Wi-Fi.

Pour elle, le réseau est un fluide invisible qui habite la maison et se vexe facilement. On l'apaise en éteignant et en rallumant une boîte à petites lumières. Si ça ne suffit pas, on déplace la boîte. Si ça ne suffit toujours pas, on appelle le neveu, qui récite des formules incompréhensibles, et la boîte guérit. Ce n'est pas de la technologie. C'est une liturgie.

Le côté obscur de la loi, c'est qu'elle vaut aussi pour ceux qui vendent la technologie : plus ça ressemble à de la magie, moins les gens posent de questions. Moi, au contraire, j'explique toujours tout. Les clients m'écoutent avec respect, hochent la tête, puis redémarrent quand même. La magie gagne.

LOI DE STURGEON



Ça vaut pour la SF, les applis, les plugins. Probablement aussi pour ce graphique.

Quatre-vingt-dix pour cent de n'importe quoi, c'est de la camelote : ça vaut pour la science-fiction, mais aussi pour les livres, les films, les logiciels et les produits en tout genre.

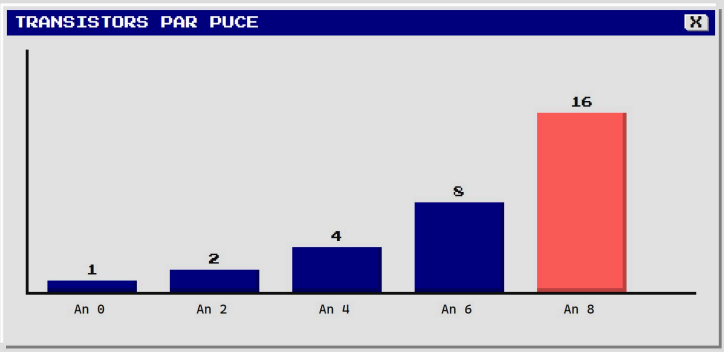
— Theodore Sturgeon, *Venture Science Fiction, mars 1958 (appelée aussi « révélation de Sturgeon »)*

Un écrivain de science-fiction s'entendait dire que quatre-vingt-dix pour cent de la science-fiction étaient des déchets. Il répondit : bien sûr, parce que quatre-vingt-dix pour cent de tout, ce sont des déchets. Fin de la discussion, applaudissements, et une loi qui vaut partout.

En informatique, c'est presque une unité de mesure. Quatre-vingt-dix pour cent des applis de la boutique sont inutiles. Quatre-vingt-dix pour cent des bibliothèques sur Internet ne sont pas maintenues. Quatre-vingt-dix pour cent des plugins Joomla que j'ai installés en 2011 sont toujours là, et personne ne se souvient à quoi ils servaient.

Le beau côté de la loi, c'est qu'elle n'est pas pessimiste : elle dit que les dix pour cent de bon existent. Le problème est de les trouver. J'y ai bâti une carrière. Mon travail, au fond, consiste à tamiser le sable. De temps en temps, je trouve une pépite. Plus souvent, un pilote.

LOI DE MOORE



En 1965, quatre points sur un graphique ont suffi. Ça a duré un demi-siècle.

Le nombre de transistors qu'on parvient à entasser sur une puce double à intervalles réguliers : d'abord chaque année, puis, après correction du tir, environ tous les deux ans.

— Gordon Moore, *Electronics, 1965 (prévision révisée en 1975)*

En 1965, un ingénieur regarda quatre points sur un graphique et dit : ça va continuer comme ça. Il a eu raison pendant un demi-siècle. Moi, je regarde quatre points sur le calendrier et je dis que je finirai le site d'ici vendredi. Nous avons des méthodes similaires, des résultats différents.

En pratique, cela signifie que tous les deux ans l'ordinateur devient deux fois plus puissant. Et tous les deux ans, le logiciel que vous utilisez devient deux fois plus lourd, si bien que le résultat net, c'est que Word s'ouvre toujours en six secondes. C'est un équilibre cosmique. Un match nul éternel entre ingénieurs du matériel et ingénieurs du logiciel, et l'arbitre, c'est le sablier de Windows.

Aujourd'hui, les transistors sont si petits qu'ils vont bientôt toucher les atomes. Mon cousin dit qu'à ce moment-là, il suffira de serrer plus fort. Il a même un tournevis exprès.

LE CONSEIL DU SYSADMIN

Murphy ne se bat pas, il se contourne. Avant toute intervention sérieuse, mise à jour, déménagement de serveur ou modification du site, je suis quatre règles écrites sur une vraie feuille, collée au moniteur.

- **Une copie d'abord.** Sauvegarde complète et vérifiée, faite maintenant et pas « l'autre jour ».
- **Un chemin pour revenir en arrière.** Si je ne sais pas annuler la modification, je ne la fais pas.
- **Une chose à la fois.** En changer trois d'un coup, c'est ne pas savoir laquelle a cassé la quatrième.
- **Une note par étape.** Date, ce que j'ai fait, ce qui s'est passé. Dans six mois, ce sera le seul document fiable.

Pas besoin d'un logiciel coûteux : un cahier suffit. Le cahier ne se met pas à jour tout seul et ne réclame pas de licence. Murphy frappera quand même, mais il trouvera la porte de service déjà ouverte pour sortir.

CHAPITRE 2

CHAP 02 — FERRAILLE

Ferraille : imprimantes, câbles et disques qui écoutent

Le logiciel fait peur parce qu'on ne le voit pas. Le matériel fait peur parce qu'on le voit très bien, et qu'il vous regarde. L'imprimante au voyant orange, le câble trop court, la vis qui roule sous la carte mère avec un bruit qui ressemble à un ricanement.

Ce chapitre parle des objets. Ceux qui sont sur le bureau, sous le bureau et, dans les pires cas, derrière le bureau, où la poussière a développé un microclimat. Ce sont les lois les plus démocratiques du livre : pas besoin de savoir programmer pour les avoir subies. Il suffit d'avoir dû imprimer un contrat à neuf heures moins cinq.

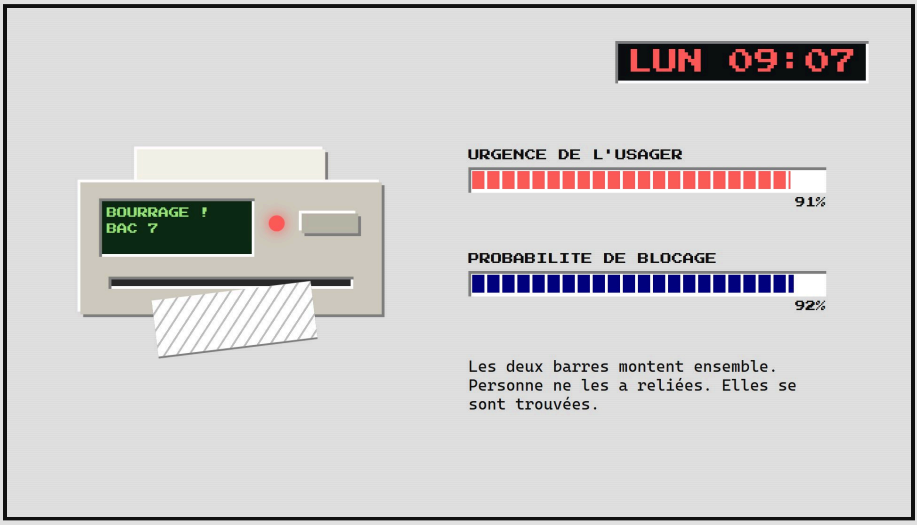
Dans la deuxième partie, on descend dans les disques, où vivent les données et leurs peurs. La sauvegarde qui n'existe pas, celle qui existe mais ne se restaure pas, le fichier qui vit toujours sur l'autre ordinateur. Ici l'ironie est plus fine, parce que la douleur est plus vraie. Une imprimante cassée, ça se remplace. Les photos de la communion, non.

Lisez-le avec un tournevis à portée de main. Il ne servira pas, mais ça rassure.

CONTENU.DIR

01 Loi de l'imprimante impatiente
02 Loi du toner voyant
03 Loi du bourrage sélectif
04 Loi de l'imprimante réseau fantôme
05 Loi du câble trop court
06 Loi du câble manquant
07 Loi de la clé USB quantique
08 Loi de la vis fugitive
09 Loi de la garantie expirée
10 Loi de la batterie optimiste
11 Loi du vidéoprojecteur hostile
12 Loi du pilote introuvable
13 Loi de la sauvegarde jamais faite
14 Loi du disque médium
15 Loi de la restauration jamais testée
16 Loi de l'enregistrement raté
17 Loi du fichier sur l'autre ordinateur
18 Loi de Segal
19 Loi de Kryder

LOI DE L'IMPRIMANTE IMPATIENTE



La probabilité que l'imprimante se bloque est directement proportionnelle à l'urgence de celui qui doit imprimer.

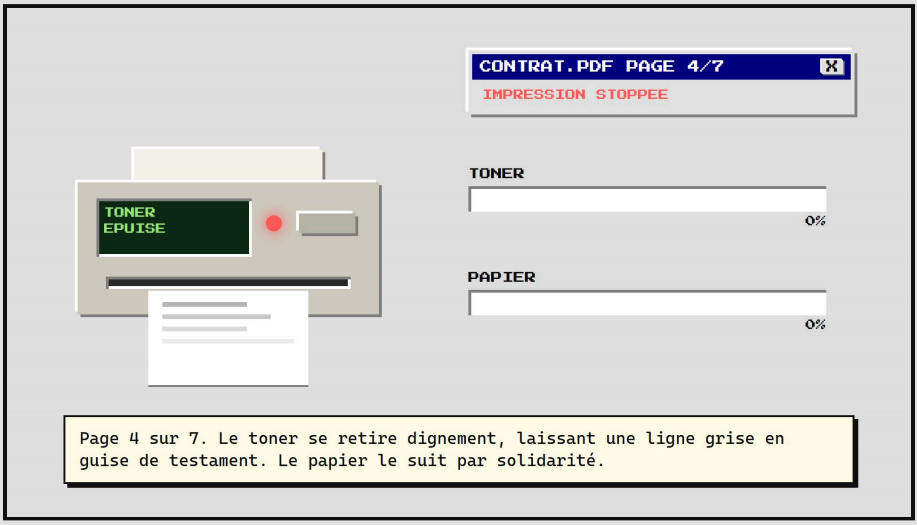
— Anonyme, *folklore informatique*

L'imprimante n'a pas d'yeux, pas d'oreilles et aucune connexion avec l'horloge de la maison. Et pourtant, elle sait. Le lundi à neuf heures, avec le train qui part à neuf heures dix, l'imprimante perçoit le rythme cardiaque de l'utilisateur comme un requin perçoit une goutte de sang.

J'ai connu un bureau qui, pendant des années, n'imprimait qu'avec calme, en chantonnant, pour ne pas se faire repérer. Ça marchait. Puis un jour, un employé a regardé sa montre. Quarante-deux minutes de « préchauffage en cours ».

Conseil de technicien : les documents importants s'impriment la veille au soir. Pas parce que l'imprimante dort, mais parce que le lendemain matin vous avez un plan B, et elle le sait.

LOI DU TONER VOYANT



Page 4 sur 7. Le toner se retire dignement, laissant une ligne grise en guise de testament. Le papier le suit par solidarité.

Le toner et le papier s'épuisent toujours au milieu du document le plus important de la semaine, jamais pendant les essais.

— Anonyme, *folklore informatique*

Le toner est un liquide prophétique. Pendant trois mois, il imprime les tracts de la fête du village, des recettes et des photos du chat sans broncher. Puis arrive le contrat à signer, et il se retire dignement à la page quatre sur sept, laissant une ligne grise en guise de testament.

Mon cousin secouait la cartouche chaque matin, par habitude, comme on fait avec un thermomètre. Il disait que ça faisait peur au toner et qu'il restait. Il avait un tiroir plein de cartouches secouées. Il n'en a jamais terminé une avant une signature chez le notaire.

Pendant ce temps, dans la salle de réunion, quelqu'un demande si on peut imprimer « seulement en noir ». La réponse est non : c'est justement le noir qui est vide.

Mini-conseil : gardez toujours une cartouche de rechange. Ça coûte moins cher que la honte.

LOI DU BOURRAGE SELECTIF



DERNIERE FEUILLE A EN-TETE

Logo en relief. Restants : 1. Valeur affective : élevée.

Le rouleau s'illumine d'une joie ancestrale et la transforme en accordéon.

FORME DE LA FEUILLE : INCONNUE DE LA GEOMETRIE EUCLIDIENNE

Le papier ne se coince que sur l'unique feuille que vous ne pouvez pas réimprimer : le papier à en-tête, le formulaire pré-imprimé, la dernière bonne feuille.

— Anonyme, *folklore informatique*

Les imprimantes ont un talent de sommelier. Face à une ramette à deux euros, elles sont dociles comme des agneaux. Mais dès que vous insérez la dernière feuille de papier à en-tête, celle avec le logo en relief, le rouleau s'illumine d'une joie ancestrale et la transforme en accordéon.

J'ai vu des collègues ouvrir des trappes dont j'ignorais l'existence. Trappe A, trappe B, trappe « ne pas ouvrir pendant le fonctionnement ». À l'intérieur, la feuille était pliée selon une forme que la géométrie euclidienne n'a pas prévue.

L'imprimante, pendant ce temps, affichait à l'écran une petite icône animée d'une main qui retire le papier. Tout en douceur. Comme quelqu'un qui explique à un enfant que le ballon est tombé dans le jardin du voisin.

Vrai conseil : avant d'utiliser la feuille précieuse, faites un essai sur du papier ordinaire. Toujours.

LOI DE L'IMPRIMANTE RESEAU FANTOME



Le document confidentiel que vous envoyez à l'impression finit toujours sur l'imprimante de l'autre étage, devant la mauvaise personne.

— Anonyme, *folklore informatique*

Dans chaque entreprise, il existe une imprimante que personne n'a installée. Elle apparaît dans la liste sous un nom du genre « HP_DEUXIEME_ETAGE_VIEILLE ». Personne ne sait où elle se trouve. Mais c'est toujours elle l'imprimante par défaut quand vous imprimez votre lettre de démission.

Un jour, j'ai suivi le parcours d'un document à travers tout le bâtiment. Parti de la comptabilité, il était passé par l'entrepôt et avait été ramassé, lu et commenté par le service expéditions. Trois personnes m'ont dit « jolie lettre ». Ce n'était pas la mienne.

L'imprimante fantôme ne se trompe pas. Elle choisit. Elle tient des archives secrètes de tout ce qu'elle n'aurait pas dû voir, et un jour elle écrira ses mémoires.

Conseil d'admin sys : vérifiez quelle imprimante est définie par défaut avant d'appuyer sur « Imprimer ». Deux secondes qui sauvent des carrières.

LOI DU CABLE TROP COURT



Tout câble est toujours dix centimètres plus court que la distance qu'il doit couvrir.

— Anonyme, *folklore informatique*

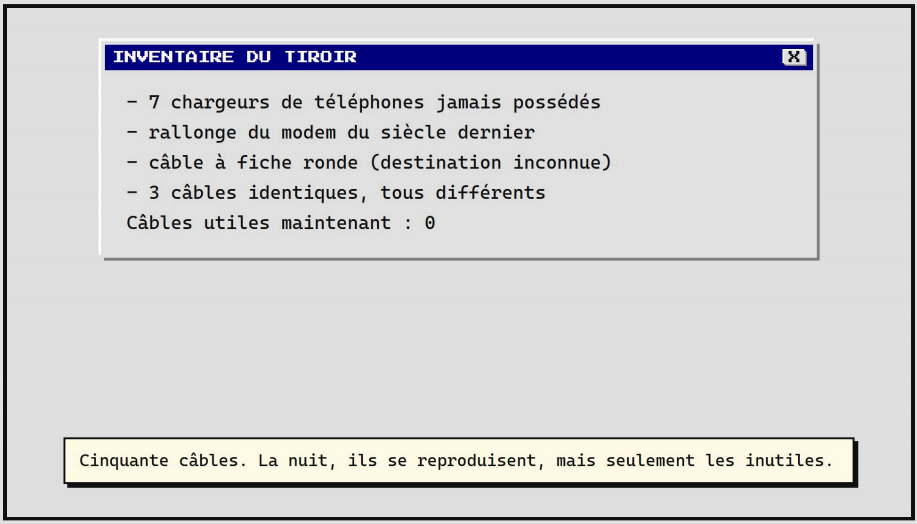
C'est une constante universelle, comme la vitesse de la lumière. Si la prise est à un mètre, le câble fait quatre-vingt-dix centimètres. Si vous achetez celui de deux mètres, le bureau se déplace. Personne ne l'a vu bouger, mais le mètre ne ment pas.

J'ai un ami qui a essayé de déjouer la loi en achetant un câble de dix mètres pour brancher un écran situé à cinquante centimètres. Le câble est arrivé. Mon ami a aussi déménagé, et la nouvelle prise était à dix mètres dix. Coïncidences, disent les sceptiques.

Le câble trop court est aussi un maître de vie : il enseigne la patience, l'humilité, et surtout à se glisser sous le bureau avec la lampe de poche dans la bouche.

Conseil de technicien : mesurez, ajoutez cinquante centimètres, et achetez celui-là. Un câble long s'enroule ; un câble court ne s'allonge pas.

LOI DU CABLE MANQUANT



Dans un tiroir contenant cinquante câbles différents, il n'y a jamais celui dont vous avez besoin maintenant.

— Anonyme, *folklore informatique*

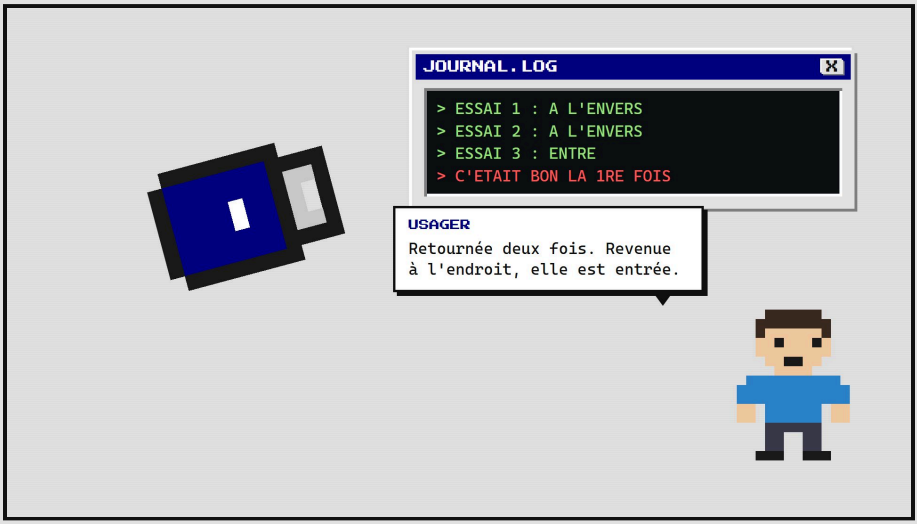
Chaque maison a son Tiroir. Un écosystème de câbles entremêlés qui se reproduisent la nuit : chargeurs de téléphones jamais possédés, rallonges de modems du siècle dernier, un câble mystérieux avec une fiche ronde qui ne rentre dans rien.

Cinquante câbles. Zéro avec le bon connecteur. Statistiquement impossible, et pourtant ça arrive à chaque fois. Le câble qu'il vous faut existe, bien sûr : il est dans le tiroir du voisin, qui de son côté cherche le vôtre.

J'ai proposé à la copropriété une bourse aux câbles, avec échanges le samedi matin. Ça a échoué parce que personne ne voulait céder son câble inutile. « On ne sait jamais », disaient-ils. La devise officielle de tout tiroir.

Vrai conseil : étiquetez les câbles avec un bout de ruban adhésif et jetez sans pitié ceux que vous ne reconnaissez pas depuis deux ans.

LOI DE LA CLE USB QUANTIQUE



La fiche USB entre toujours au troisième essai : la première fois elle est à l'envers, la deuxième aussi, la troisième elle était dans le bon sens depuis le début.

— Anonyme, *folklore informatique*

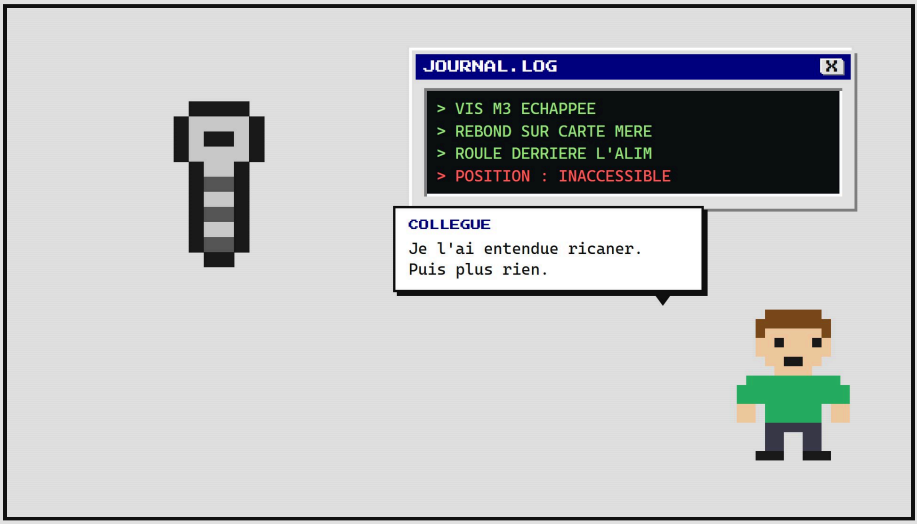
Le connecteur USB classique a deux côtés. Mathématiquement, au premier essai, vous avez cinquante pour cent de chances de tomber juste. En pratique, la probabilité est nulle. Le connecteur existe dans un état de superposition : il est faux dans les deux sens tant que vous ne l'avez pas retourné deux fois.

J'ai calculé qu'au cours de ma carrière, j'ai retourné des clés USB pendant environ trois jours consécutifs de ma vie. Je les considère comme des congés non pris.

Puis est arrivé le connecteur réversible, et l'humanité a fait la fête. Pendant un après-midi. Ensuite, nous nous sommes aperçus que les câbles réversibles ne chargent pas tous à la même vitesse, ne transmettent pas tous les données et n'allument pas tous l'écran. Le chaos s'est simplement déplacé à l'intérieur du câble.

Conseil : le logo USB sur le connecteur, en général, va vers le haut. En général.

LOI DE LA VIS FUGITIVE



La vis qui vous glisse des doigts pendant que vous ouvrez l'ordinateur tombe toujours à l'endroit le plus inaccessible du boîtier.

— Anonyme, *folklore informatique*

Dans un ordinateur, il y a des milliers d'endroits commodes où une vis pourrait atterrir. Elle choisit toujours le seul qui est sous la carte mère, derrière l'alimentation, dans une fente conçue par quelqu'un qui détestait les doigts humains.

Le technicien exécute alors l'antique danse du boîtier : il l'incline, le secoue, le retourne, écoute. La vis roule avec un bruit métallique, comme un ricanement. Puis se tait. Personne ne la reverra jamais, mais l'ordinateur fait désormais un bruit nouveau quand on le déplace.

Dans mon atelier, nous gardons un bocal de vis orphelines récupérées au fil des ans. Aucune n'a jamais servi à rien. Nous l'appelons « la caisse de retraite ».

Conseil de technicien : utilisez un tournevis aimanté et une petite coupelle pour les vis. Et ne travaillez jamais au-dessus de la moquette.

LOI DE LA GARANTIE EXPIREE



Un appareil tombe en panne le lendemain de l'expiration de la garantie, sauf dans les cas où il tombe en panne la veille et où vous avez perdu le ticket de caisse.

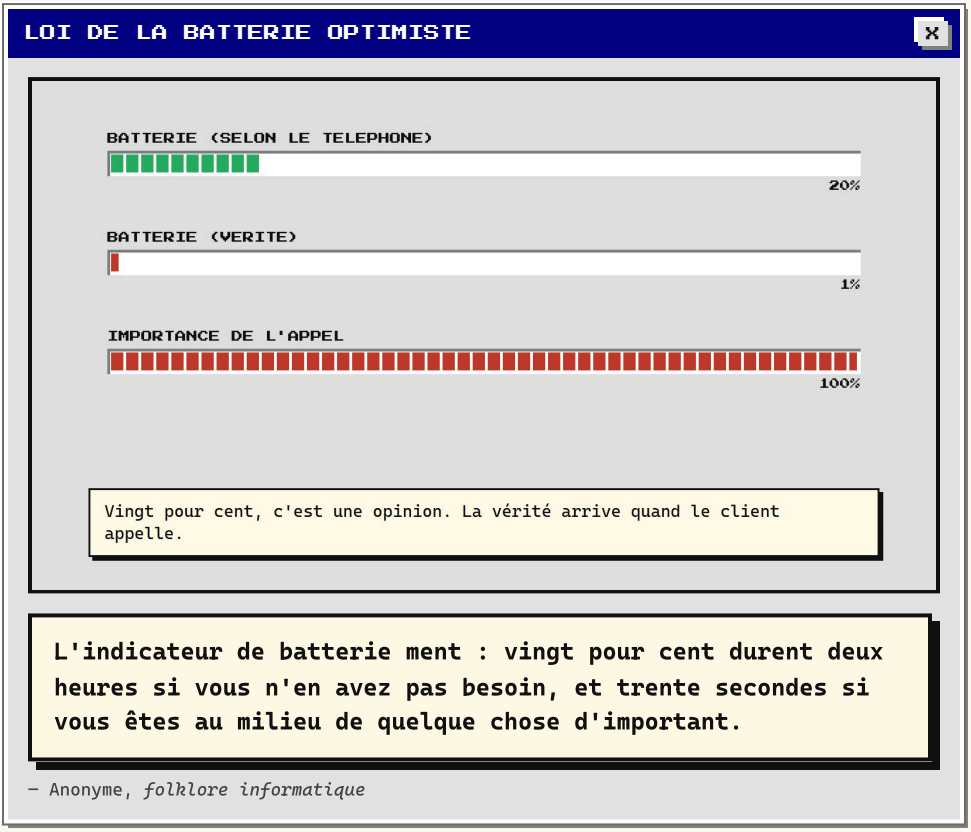
— Anonyme, *folklore informatique*

Dans chaque appareil moderne, il y a un calendrier. Ce n'est pas écrit dans le manuel, mais il y est. Il compte les jours avec la patience d'un moine et, à minuit du vingt-quatrième mois, il ferme les yeux en paix.

Une de mes clientes avait conservé tous ses tickets de caisse depuis 1998 dans une boîte à chaussures, classés par couleur. Le jour où l'imprimante est tombée en panne sous garantie, la boîte était à la cave. La cave était inondée. Le ticket était devenu un petit nuage blanc.

La garantie, en somme, existe : c'est la licorne du service client. Tout le monde en parle, certains jurent l'avoir vue.

Conseil de technicien : photographiez le ticket dès l'achat et enregistrez-le dans le cloud. Le papier thermique s'efface, les photos non.



L'indicateur de batterie ment : vingt pour cent durent deux heures si vous n'en avez pas besoin, et trente secondes si vous êtes au milieu de quelque chose d'important.

— Anonyme, *folklore informatique*

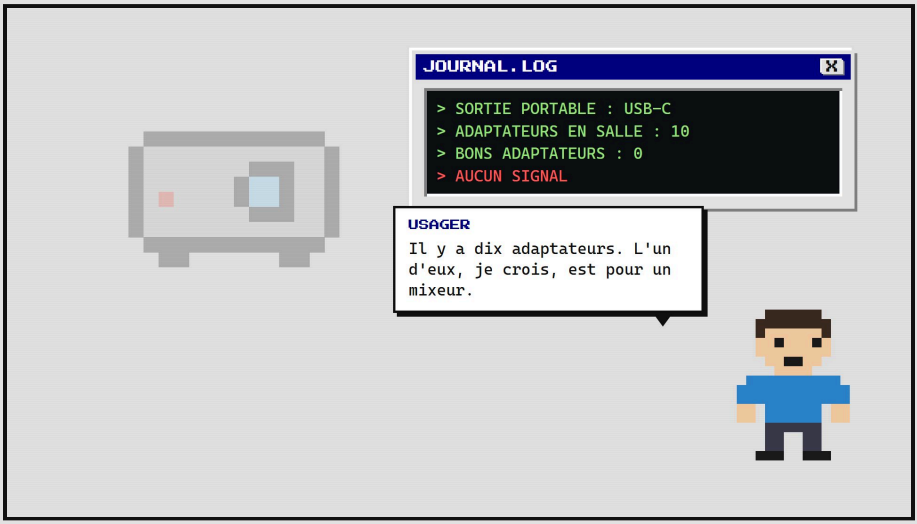
Le pourcentage de batterie n'est pas une donnée. C'est une opinion. Un sentiment. Le téléphone à vingt pour cent vous dit « t'inquiète, on va y arriver », sur le ton de quelqu'un qui roule sur la réserve sur l'autoroute.

Puis vous ouvrez la carte pour trouver le client, et le pourcentage descend comme un ascenseur sans câbles : dix-huit, onze, quatre, noir. Le chargeur est à la maison, en charge lui aussi, par solidarité.

Mon oncle soutenait que les batteries se rechargent toutes seules si on les laisse se reposer dans un tiroir. Il a raison pendant environ une minute. C'est la minute la plus optimiste de sa journée.

Vrai conseil : les batteries au lithium s'usent, et avec l'âge l'indicateur devient imprécis. Si l'appareil s'éteint alors qu'il affiche un pourcentage élevé, il est temps de la remplacer.

LOI DU VIDEOPROJECTEUR HOSTILE



Le vidéoprojecteur de la salle de réunion a une entrée différente de celle de votre portable, et le bon adaptateur est le seul que personne n'a apporté.

— Anonyme, *folklore informatique*

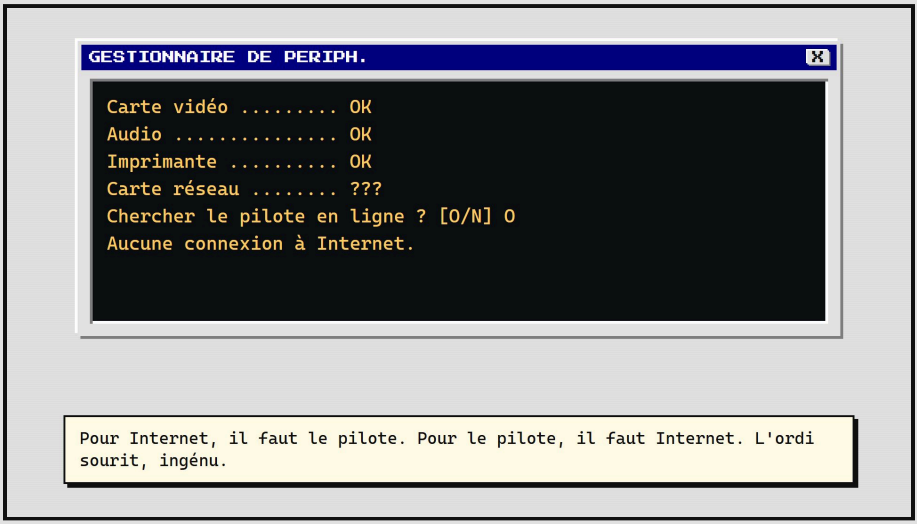
Chaque salle de réunion a un vidéoprojecteur suspendu au plafond comme une chauve-souris électronique. Il n'a qu'une seule entrée, d'un type que votre portable a abandonné il y a trois générations. Sur la table, une boîte d'adaptateurs : dix, tous pour d'autres ordinateurs, certains pour des objets qui ne sont pas des ordinateurs.

Pendant ce temps, le public attend. Quelqu'un propose « d'envoyer la présentation par mail et que chacun la regarde sur son téléphone ». C'est le moment où la civilisation repart du papier.

Quand l'image apparaît enfin, elle est violette, coupée à droite, et l'écran du portable s'éteint. Vous avez gagné, mais vous ne savez pas comment.

Conseil de technicien : apportez toujours votre adaptateur et une copie de la présentation en PDF sur clé USB. Et arrivez quinze minutes en avance.

LOI DU PILOTE INTROUVABLE



Après la réinstallation du système, le seul pilote qui manque est celui de la carte réseau, qui servirait à télécharger les autres pilotes.

— Anonyme, *folklore informatique*

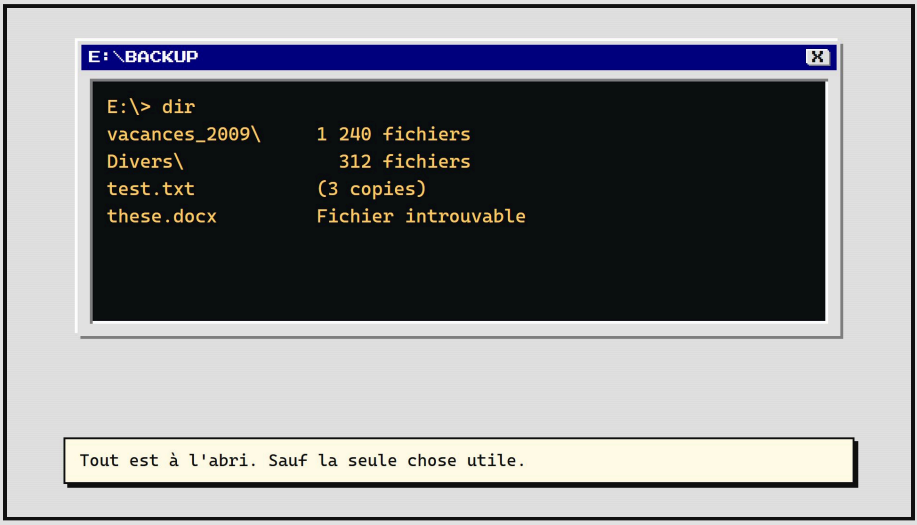
C'est le paradoxe parfait, digne d'une nouvelle de science-fiction. Pour avoir Internet, il vous faut le pilote. Pour avoir le pilote, il vous faut Internet. L'ordinateur fraîchement formaté vous regarde avec innocence, comme un nouveau-né qui ne sait pas encore parler à la box.

Le technicien part alors pour la Grande Expédition : un autre ordinateur, une clé USB, un site du fabricant qui a changé de structure six fois. Le pilote s'appelle à peu près « paquet_final_v2_nouveau_DEFINITIF.zip » et pèse plus lourd que le système d'exploitation.

Pendant ce temps, dans la pièce d'à côté, le client demande si, « tant que vous y êtes », vous pouvez aussi récupérer les photos du mariage.

Vrai conseil : avant de formater, téléchargez sur une clé USB les pilotes réseau de votre modèle. C'est la boîte à outils du lendemain.

LOI DE LA SAUVEGARDE JAMAIS FAITE



De toutes les sauvegardes possibles, celle dont vous avez besoin est toujours la seule que vous n'avez pas faite.

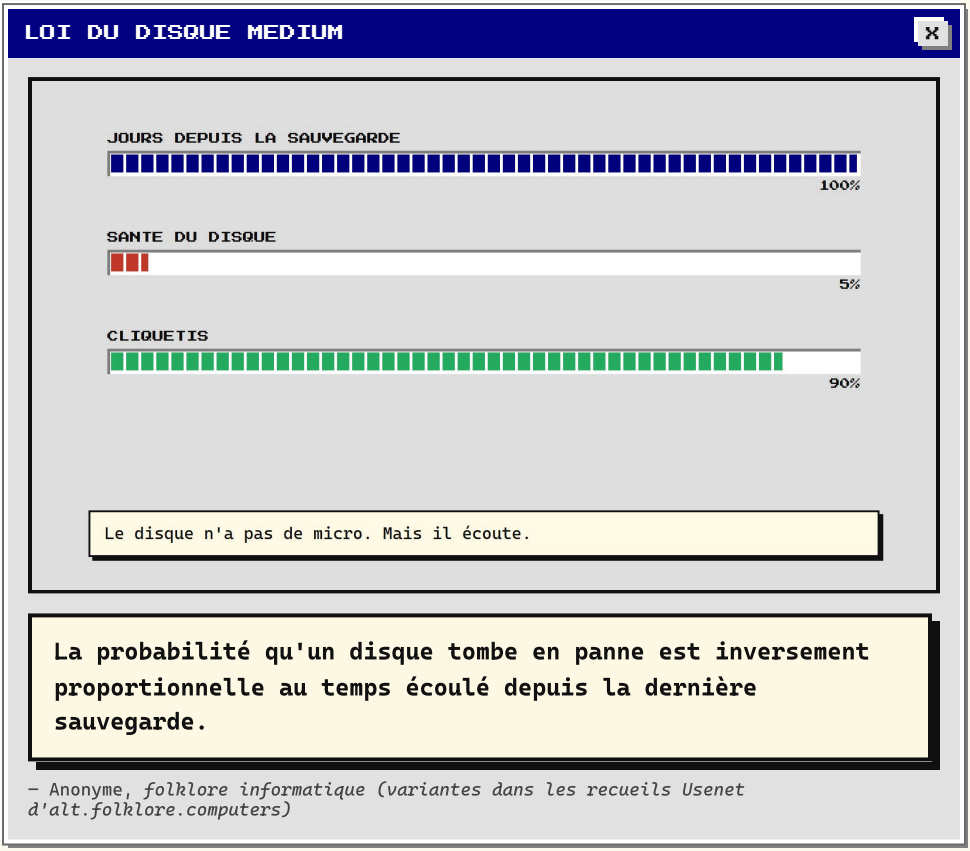
— Anonyme, *folklore informatique*

Vous avez la sauvegarde des photos des vacances 2009. Vous avez la sauvegarde du dossier « Divers ». Vous avez trois sauvegardes identiques d'un fichier nommé « essai.txt ». Puis le disque meurt, et le seul fichier dont vous avez vraiment besoin, c'est le mémoire, qui était sur le bureau « juste pour aujourd'hui ».

La sauvegarde est une forme de superstition à l'envers : si vous la faites, il ne se passe rien et vous pensez avoir perdu votre temps. Si vous ne la faites pas, l'univers organise une démonstration pratique.

J'ai un client qui faisait sa sauvegarde sur une clé USB, qu'il laissait branchée à l'ordinateur, qui était sous la fenêtre, sous laquelle est passé un orage. Maintenant, il a deux choses mouillées au lieu d'une.

Vrai conseil : règle du 3-2-1. Trois copies, sur deux supports différents, dont une hors de la maison ou dans le cloud.



La probabilité qu'un disque tombe en panne est inversement proportionnelle au temps écoulé depuis la dernière sauvegarde.

— Anonyme, *folklore informatique (variantes dans les recueils Usenet d'alt.folklore.computers)*

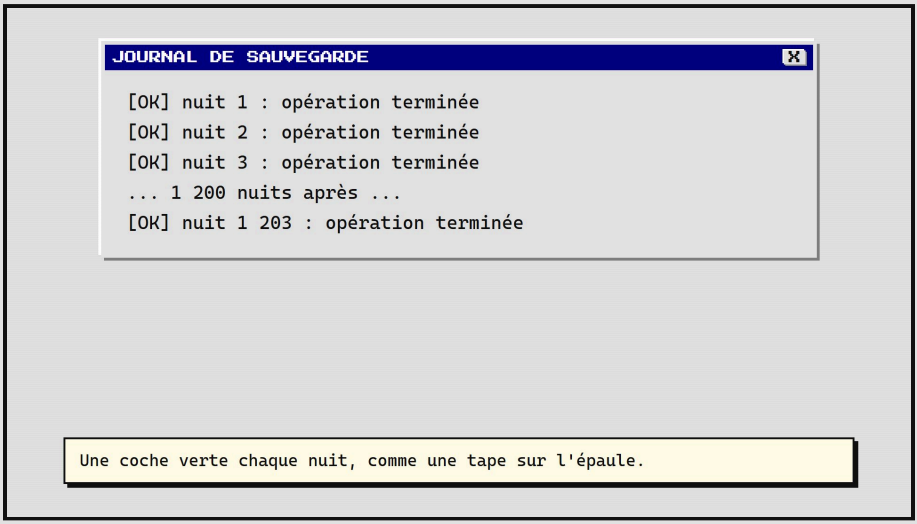
Le disque dur écoute. Il n'a pas de micro, mais il écoute. Le lendemain d'une sauvegarde complète, il tourne heureux, silencieux, éternel. Après six mois sans copie, il commence à faire un petit cliquetis, comme une horloge de film à suspense.

Les vieux techniciens disaient que le cliquetis, c'était « la tête de lecture qui dit bonjour ». En réalité, c'était la tête de lecture qui disait adieu.

Il n'existe qu'une façon de tromper le disque : faire la sauvegarde sans qu'il le sache. Discrètement, la nuit, automatiquement. Le disque ne découvre la copie qu'après coup, et à ce moment-là l'envie de tomber en panne lui passe. Ce n'est pas de la science. Mais les résultats sont excellents.

Conseil d'admin sys : programmez une sauvegarde automatique. Celle qui dépend de votre mémoire n'est pas une sauvegarde, c'est un espoir.

LOI DE LA RESTAURATION JAMAIS TESTEE



Une sauvegarde marche toujours, jusqu'au jour où vous essayez de la restaurer.

— Anonyme, *folklore informatique*

Pendant des années, le logiciel de sauvegarde a affiché une coche verte. Chaque nuit. « Opération terminée. » Une coche rassurante comme une tape sur l'épaule.

Puis arrive le jour où l'on en a besoin. Vous ouvrez l'archive et découvrez qu'elle contient un seul dossier vide, copié avec un soin religieux pendant mille deux cents nuits consécutives. Le logiciel n'a jamais menti : l'opération était bel et bien terminée. Laquelle, personne ne l'avait demandé.

Une sauvegarde non vérifiée, c'est comme un parachute jamais ouvert en magasin. Il est probablement en bon état. Probablement.

Vrai conseil de technicien : de temps en temps, au moins, essayez de restaurer un fichier au hasard depuis la sauvegarde. Si vous le récupérez, dormez tranquille. Sinon, vous venez d'éviter la pire des catastrophes.



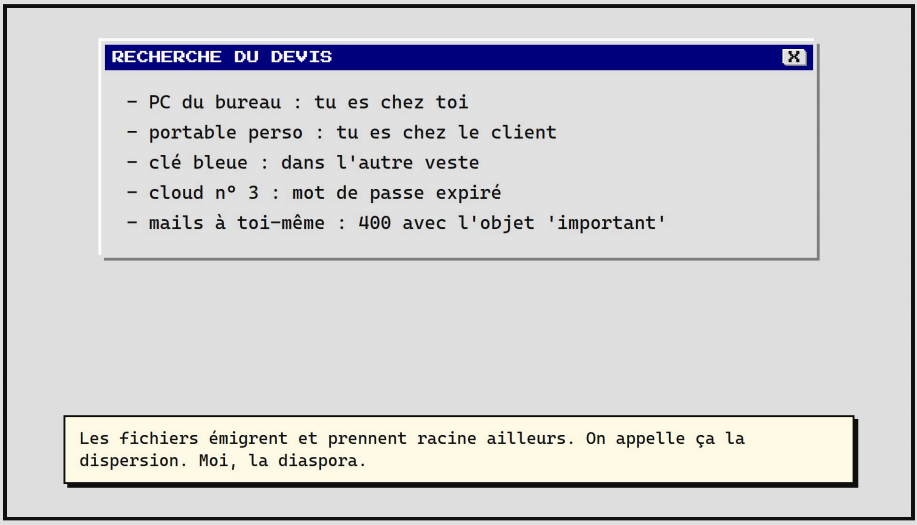
Il y a un moment, pendant le travail, où vous pensez : « Je devrais enregistrer. » C'est une pensée noble. Mais vous la repoussez d'une minute, juste le temps de finir la phrase. Le logiciel intercepte cette pensée, la trouve présomptueuse, et se ferme.

Il ne se ferme pas brutalement. Il se ferme avec élégance. Une petite fenêtre dit « Le programme a cessé de fonctionner », comme un serveur qui annonce que la cuisine est fermée.

Ma tante enregistrait chaque document après chaque mot. Elle mettait un après-midi à écrire une carte de vœux, mais elle n'a jamais rien perdu. On l'appelait « le Coffre-fort ».

Vrai conseil : apprenez le raccourci Ctrl+S et utilisez-le comme un tic. Et activez l'enregistrement automatique, si le logiciel le propose.

LOI DU FICHIER SUR L'AUTRE ORDINATEUR



Le fichier dont vous avez besoin en urgence se trouve toujours sur l'autre ordinateur, sur l'autre clé USB ou dans l'autre maison.

— Anonyme, *folklore informatique*

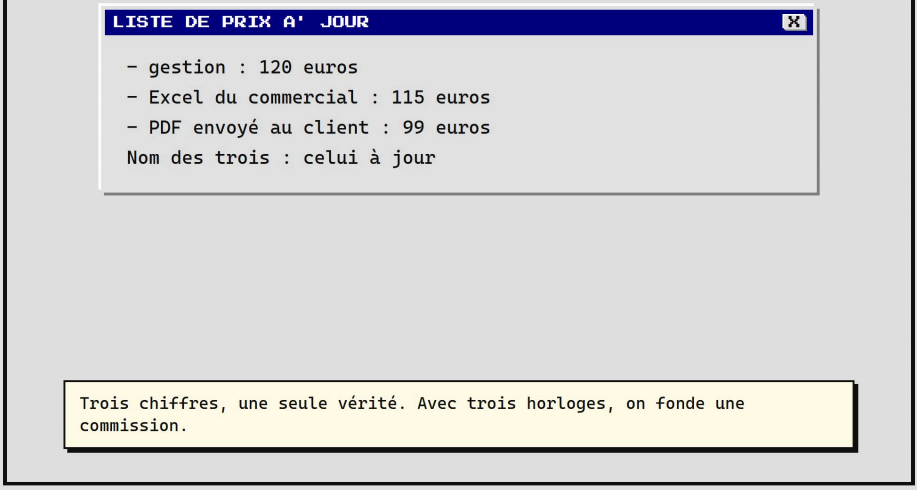
Vous avez trois ordinateurs, deux clés USB, un disque externe et quatre services de cloud. Le fichier du devis se trouve à un seul de ces endroits. Et c'est toujours celui auquel vous ne pouvez pas accéder : l'ordinateur du bureau quand vous êtes chez vous ; le portable de la maison quand vous êtes chez le client.

On appelle ça la dispersion des données, mais moi je l'appelle la diaspora. Les fichiers émigrent, prennent racine ailleurs, changent de nom en « devis_definitif_2_copie ».

Un jour, j'ai cherché un document pendant une heure. Il était joint à un e-mail que je m'étais envoyé à moi-même avec pour objet « important ». Il y en avait quatre cents avec le même objet.

Vrai conseil : choisissez un seul espace de travail principal, synchronisé, et donnez aux fichiers des noms clairs avec la date. Le vous du futur vous remerciera.

LOI DE SEGAL



Un homme avec une montre sait quelle heure il est. Un homme avec deux montres n'en est jamais sûr.

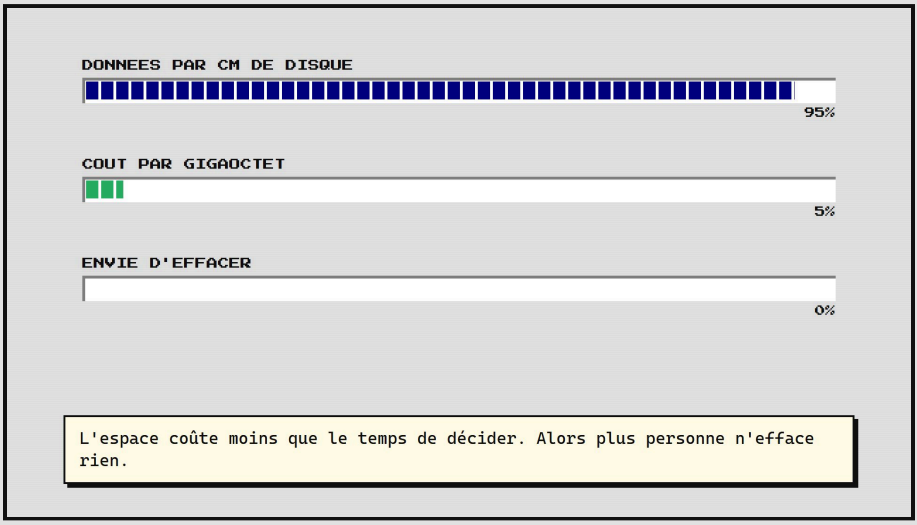
— Anonyme, *Dicton paru dans le San Diego Union en 1930 ; attribué ensuite par erreur à Lee Segall et rendu célèbre comme « loi de Segal » par les livres d'Arthur Bloch (attribution incertaine)*

Avec une seule montre, aucun doute : vous vous trompez en toute sérénité. Avec deux montres qui indiquent des heures différentes, la crise commence. Laquelle a raison ? Avec trois, on fonde une commission.

En informatique, c'est le drame des sources multiples. La grille tarifaire se trouve dans le logiciel de gestion, dans le fichier Excel du commercial et dans le PDF envoyé au client. Ce sont trois chiffres différents, et ils s'appellent tous « celui à jour ». C'est pour cela que ceux qui conçoivent des systèmes sérieux parlent de source unique de vérité : un seul endroit d'où tout le monde lit, et les autres copient.

Ça vaut aussi pour les serveurs : si les horloges des ordinateurs d'un réseau ne sont pas synchronisées, les logs racontent des histoires impossibles, avec des réponses qui arrivent avant les questions. Au bureau, j'ai résolu le problème. J'ai enlevé toutes les horloges sauf une. Elle est arrêtée depuis 2016, mais plus personne ne discute.

LOI DE KRYDER



La densité de données enregistrables sur un disque magnétique croît plus vite que la puissance des puces, et le coût par gigaoctet s'effondre en conséquence.

— Mark Kryder, *Article « Kryder's Law » de Chip Walter, Scientific American, août 2005*

Le premier disque dur commercial pesait autant qu'un réfrigérateur et contenait quelques mégaoctets. Aujourd'hui, une carte grande comme un ongle contient toutes les photos des vacances, y compris les quarante identiques du coucher de soleil. Voilà la loi de Kryder : l'espace pour les données devient plus dense et moins cher à un rythme vertigineux.

La conséquence psychologique est dévastatrice. Plus personne n'efface rien. Pourquoi choisir, si l'espace coûte moins cher que le temps de décider ? J'ai des clients avec douze copies du même PDF, nommées definitif, definitif2, definitif_vrai et definitif_vrai_cette_fois.

Ces dernières années, la croissance a ralenti, mais l'humanité a déjà trouvé la solution : accumuler quand même. Mon oncle garde ses disques cassés dans une boîte. Il dit que les données sont encore dedans, juste un peu timides.

LE CONSEIL DU SYSADMIN

Deux choses que presque personne ne fait et que tout le monde devrait faire. La première : **demander au disque comment il va**. Presque tous les disques, mécaniques et SSD, tiennent un registre de leur santé appelé SMART. Sous Windows, un programme gratuit comme CrystalDiskInfo le lit en un instant et le résume par une couleur : bleu, tout va bien ; jaune, c'est un avertissement ; rouge, c'est un problème. Le jaune n'est pas un avis, c'est un préavis : copiez vos données le jour même. Les SSD, contrairement aux vieux disques, lâchent souvent sans cliquetis d'adieu.

La seconde : **ne pas se fier aux disques en hibernation**. Un disque externe enfermé des années dans un tiroir peut ne pas redémarrer, et les clés USB oubliées perdent des bits. Rallumez vos archives de temps en temps, vérifiez que les fichiers s'ouvrent et, tous les quelques ans, copiez tout sur un support neuf. Les données ne vieillissent pas. Leurs contenants, si.

CHAPITRE 3

CHAP 03 — HELPDESK

Helpdesk : utilisateurs, mots de passe, mystères

Voici le chapitre de mon travail quotidien, et donc celui que j'ai relu avec le plus d'émotion. Il y a le téléphone qui sonne, l'utilisateur qui n'a touché à rien, le message d'erreur fermé à la vitesse de la lumière, l'ordinateur qui guérit dès que le technicien franchit le seuil.

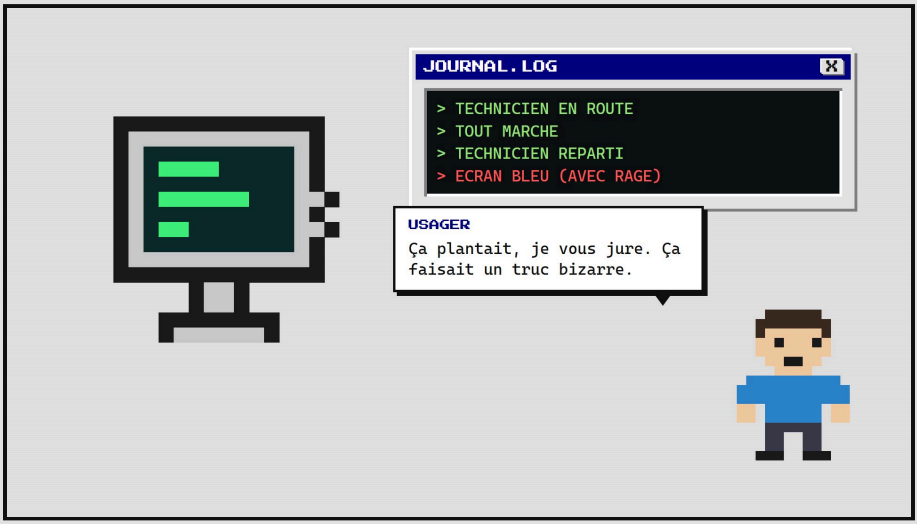
Mettons les choses au clair : ici, on ne rit pas des utilisateurs. On rit avec eux. Tout technicien est l'utilisateur de quelque chose d'autre, et moi, devant la chaudière de la maison, je suis exactement ce monsieur qui jure n'avoir touché à rien. Le rasoir d'Hanlon, au premier chapitre, le dit mieux que moi : derrière un désastre, il n'y a généralement pas de malveillance. Il y a un clic.

Puis viennent les mots de passe, qui méritent une section à eux seuls : trop simples, trop compliqués, oubliés le lendemain. Et le chapitre se termine par quelques lois sérieuses de la conception, celles qui expliquent pourquoi on rate un bouton petit et lointain, pourquoi un menu de quarante entrées paralyse et pourquoi une demi-seconde d'attente semble une éternité. Les lire aide à comprendre que beaucoup d'« erreurs de l'utilisateur » sont en réalité des erreurs de celui qui a dessiné l'écran.

CONTENU.DIR

01 Loi du technicien guérisseur
02 Loi du « j'ai touché à rien »
03 Loi de la prise débranchée
04 Loi du message d'erreur non lu
05 Loi de la panne du vendredi soir
06 Loi du parent qui s'y connaît
07 Loi du mot de passe oublié
08 Loi du mot de passe jamais assez bon
09 Principe de Kerckhoffs
10 Lois de Gilb sur le manque de fiabilité
11 Loi de Tesler
12 Loi de Jakob
13 Loi de Fitts
14 Loi de Hick
15 Seuil de Doherty

LOI DU TECHNICIEN GUERISSEUR



Le problème disparaît à l'instant exact où le technicien arrive, et réapparaît dès qu'il s'en va.

— Anonyme, *folklore informatique*

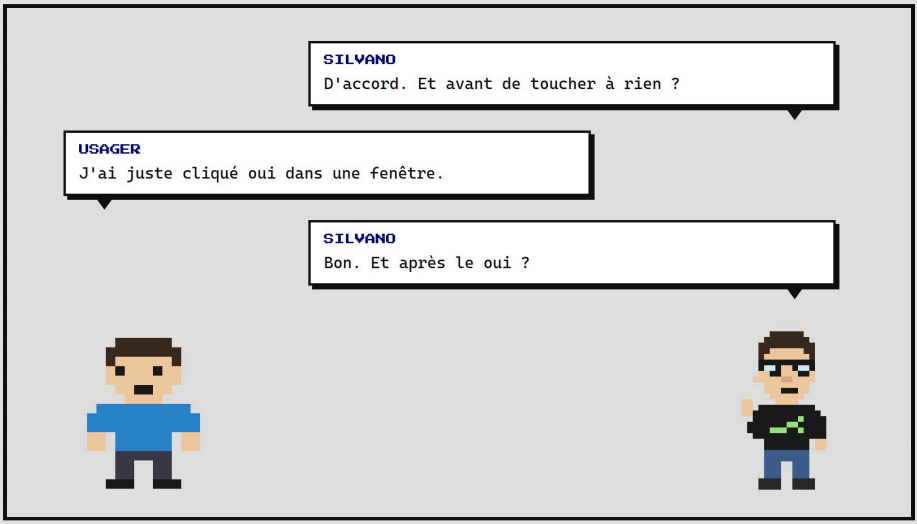
C'est mon unique superpouvoir. J'entre dans la pièce, et l'ordinateur qui affichait des écrans bleus depuis trois jours se met au garde-à-vous comme un soldat à la caserne. « Avant, ça ne marchait pas, je vous jure », dit l'utilisateur. Je le crois. J'ai vu cette scène mille fois.

Le problème reconnaît le technicien comme un chien reconnaît le vétérinaire : il est sage, il remue la queue, il nie tout. Puis, quand ma voiture quitte le parking, l'ordinateur recommence, avec un peu plus de rage pour le temps perdu.

J'ai pensé laisser au bureau une photo de moi grandeur nature. Un client l'a fait avec une silhouette en carton. Il dit que ça marche environ une semaine.

Conseil de technicien : quand l'erreur apparaît, prenez l'écran en photo avec votre téléphone. Une photo vaut mieux que mille « avant, il faisait un truc bizarre ».

LOI DU « J'AI TOUCHE A RIEN »



L'utilisateur qui dit « j'ai touché à rien » a touché à quelque chose ; la seule inconnue, c'est à quoi.

— Anonyme, *folklore informatique*

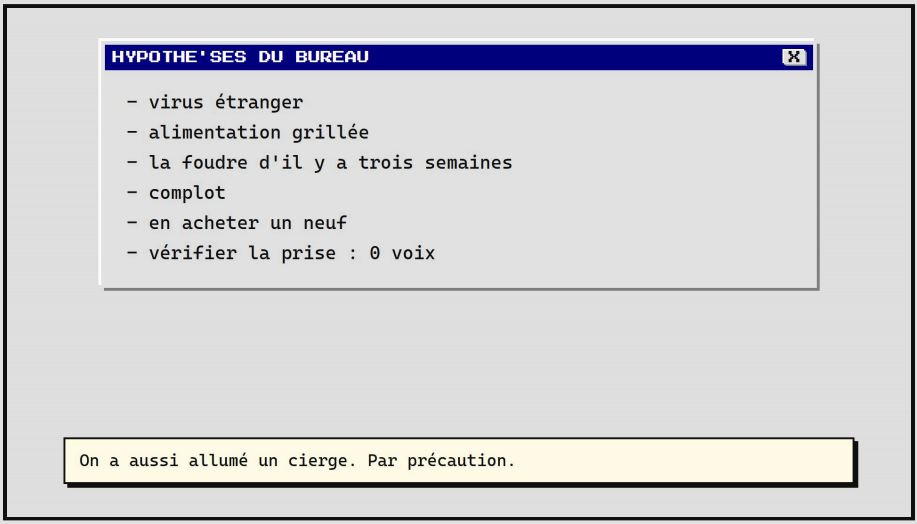
« J'ai touché à rien » est la plus ancienne phrase de l'assistance technique. On l'a trouvée gravée sur le premier ordinateur à lampes, à côté d'une empreinte digitale.

L'utilisateur ne ment pas. Dans sa mémoire, il n'a touché à rien. Il a juste cliqué sur « oui » dans une petite fenêtre, déplacé un dossier « qui gênait », installé un logiciel gratuit pour mieux voir les photos et changé un petit réglage qui « était moche ». Mais toucher, non.

Le technicien expérimenté ne conteste pas. Il hoche la tête avec la sérénité d'un confesseur et demande : « D'accord. Et avant de ne toucher à rien, vous faisiez quoi ? » C'est là qu'arrive la vérité.

Vrai conseil : raconter sincèrement les dernières manipulations fait gagner des heures. Le technicien ne juge pas, il doit seulement comprendre par où commencer.

LOI DE LA PRISE DEBRANCHEE



Avant de vérifier que la prise est branchée, on aura déjà envisagé un virus, une panne de carte mère et un complot.

— Anonyme, *folklore informatique*

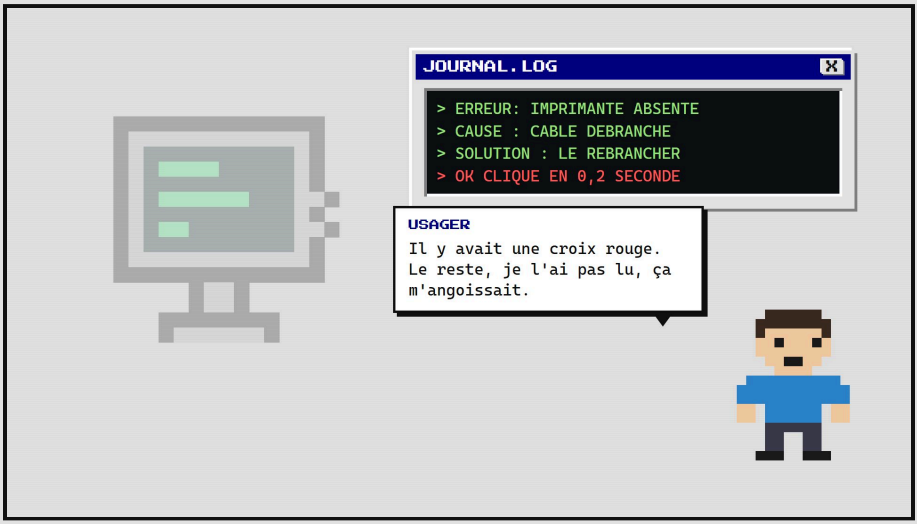
L'ordinateur ne s'allume pas. Aussitôt, le bureau se divise en écoles de pensée. Il y a ceux qui parlent de virus russes, ceux qui parlent d'alimentation grillée, ceux qui évoquent la foudre d'il y a trois semaines qui « a sûrement laissé quelque chose ». On envisage l'achat d'un nouvel ordinateur. On allume un cierge.

Puis le technicien arrive, se penche, et rebranche la prise que l'entreprise de ménage avait débranchée pour brancher l'aspirateur. Le silence qui suit est sacré.

La prise débranchée est la cause la plus humble de l'informatique. Elle n'a pas de code d'erreur, pas de log, pas de prestige. Mais elle gagne toujours.

Conseil de technicien : vérifiez d'abord les choses évidentes. Prise, multiprise allumée, câbles de l'écran bien enfoncés. Ce n'est pas banal : c'est de la méthode.

LOI DU MESSAGE D'ERREUR NON LU



L'utilisateur ferme le message d'erreur avant de le lire, puis appelle le technicien pour savoir ce qui était écrit.

— Anonyme, *folklore informatique*

Le message d'erreur est un petit poète incompris. Il apparaît avec toutes les informations nécessaires : le problème, la cause, et même la solution. L'utilisateur le regarde avec l'horreur de celui qui trouve une araignée dans la baignoire, et appuie sur « OK » à une vitesse qui défie la physique.

Puis il téléphone : « Il y a un message qui est sorti. » « Qu'est-ce qu'il disait ? » « Je ne sais pas, c'était en anglais. Ou en français. Il y avait une croix rouge. » C'est tout ce qui reste, comme le récit d'une observation d'OVNI.

Le technicien doit alors recréer l'erreur, en attendant que le poète revienne déclamer. Le poète, vexé, ne revient pas.

Vrai conseil : quand une erreur apparaît, ne la fermez pas tout de suite. Lisez calmement, prenez une photo ou une capture d'écran. La moitié du temps, la réponse est déjà là.

LOI DE LA PANNE DU VENDREDI SOIR



Les pannes graves ne se produisent que quand le technicien n'est pas là : le vendredi soir, le 15 août, ou le premier jour des vacances.

— Anonyme, *folklore informatique*

Le serveur a fonctionné sans une plainte pendant onze mois et trois semaines. Puis le technicien réserve l'hôtel à la mer. Le serveur le sait. Ne me demandez pas comment : il le sait.

Le vendredi à dix-huit heures une, pendant que les clés tournent dans la serrure du bureau, un petit voyant devient orange. À dix-huit heures deux, rouge. À dix-huit heures trois, le téléphone du technicien sonne, alors qu'il est déjà sur l'autoroute avec sa famille et un matelas pneumatique sur le siège.

J'ai connu un admin sys qui ne partait plus en vacances depuis six ans par peur de cette loi. Le serveur n'est jamais tombé en panne. Puis il est allé chez le dentiste un mardi matin.

Vrai conseil : une documentation claire, des accès notés dans un endroit sûr et un collègue de réserve. L'astreinte n'est pas de l'héroïsme, c'est de l'organisation.

LOI DU PARENT QUI S'Y CONNAIT



Celui qui, dans la famille, « s'y connaît en ordinateurs » passera chaque repas de fête à configurer l'imprimante d'un oncle.

— Anonyme, *folklore informatique*

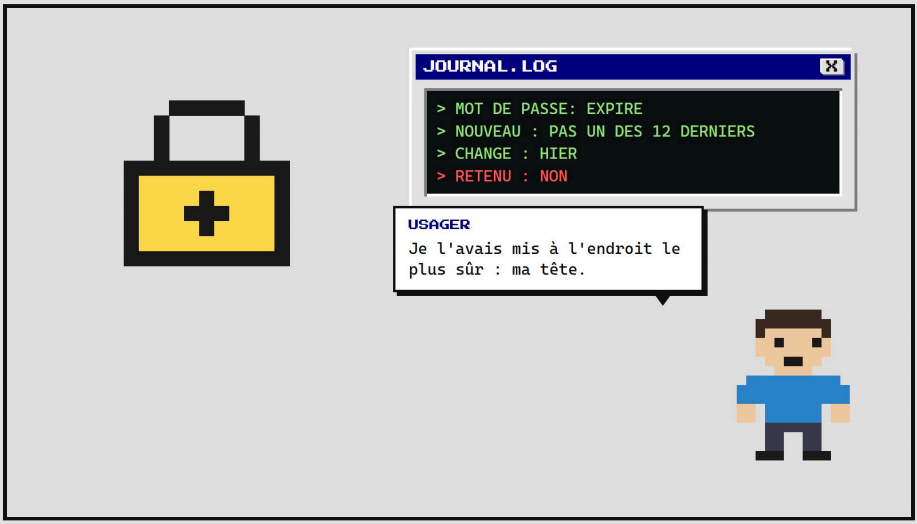
Il suffit d'une fois. Une seule fois où vous réparez le téléphone de mamie, et vous êtes nommé à vie Responsable Informatique de la Famille. Sans salaire, sans congés, avec astreinte permanente les jours fériés.

Vous arrivez pour le repas de Noël et vous n'avez pas le temps d'enlever votre manteau. « Tant que tu es là, tu pourrais regarder l'ordinateur ? » L'ordinateur est au grenier, il a douze ans et une couche de poussière qui fait office d'isolant thermique.

En fin de journée, vous avez mangé une tranche de dinde froide, mis à jour quatre téléphones et expliqué six fois que le Wi-Fi, ce n'est pas « Internet ». On vous paie avec un pot de confiture. Elle est excellente, avouons-le.

Vrai conseil : écrivez sur une feuille les mots de passe du Wi-Fi et des principaux comptes de vos proches, et laissez-la-leur dans un endroit sûr. Vous diviserez les appels par deux.

LOI DU MOT DE PASSE OUBLIE



Un mot de passe s'oublie toujours le lendemain du jour où on l'a changé, parce que le système l'imposait.

— Anonyme, *folklore informatique*

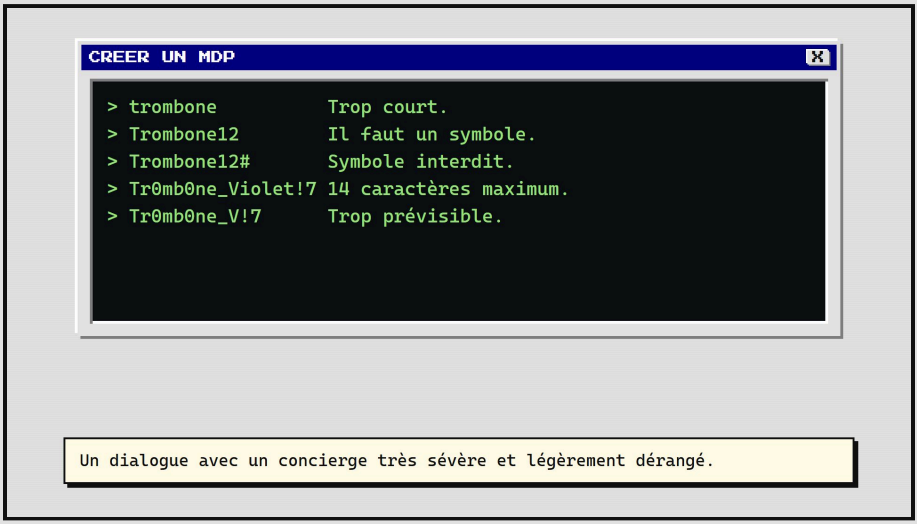
Pendant trois ans, vous avez utilisé le même mot de passe. Vous le connaissiez comme le prénom de votre enfant. Puis le système décide qu'il a expiré et en veut un nouveau, différent des douze derniers. Vous l'inventez avec soin. Le lendemain, il a disparu de votre mémoire comme un rêve au réveil.

Commence alors la chasse à la réinitialisation. « Saisissez la réponse à la question secrète. » Le nom de votre premier animal. Aviez-vous écrit « Minou », « minou » ou « MINOU » ? Le système ne le dit pas. Il vous bloque trente minutes pour réfléchir.

Le post-it jaune sous le clavier, pendant ce temps, rit. Lui, il a toujours été là.

Vrai conseil : utilisez un gestionnaire de mots de passe. Vous n'en retenez qu'un seul, robuste, et il garde les autres. Le post-it ne compte pas comme gestionnaire.

LOI DU MOT DE PASSE JAMAIS ASSEZ BON



Le site refuse tous les mots de passe jusqu'à ce qu'ils deviennent impossibles à retenir, puis refuse aussi celui-là parce qu'il contient un caractère non autorisé.

— Anonyme, *folklore informatique*

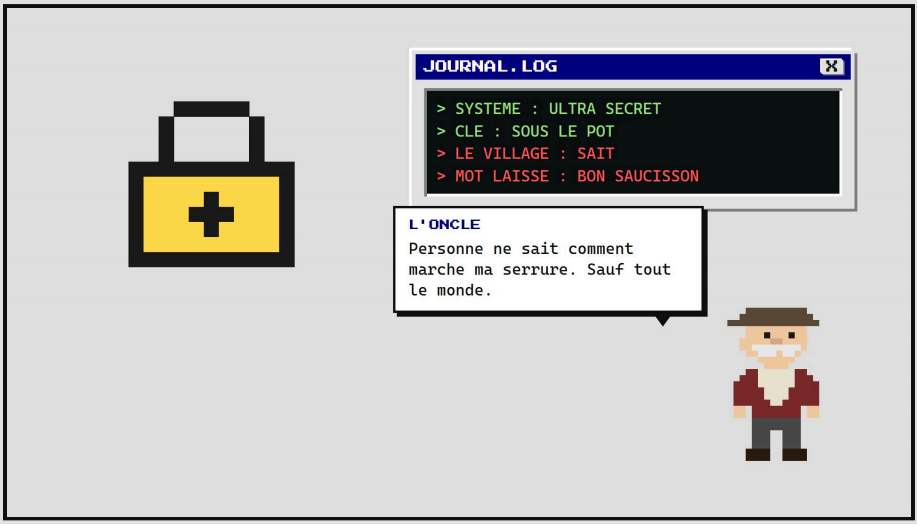
« Le mot de passe doit contenir au moins huit caractères. » Fait. « Une majuscule. » Fait. « Un chiffre. » Fait. « Un symbole. » Fait. « Il ne peut pas contenir le symbole que vous avez utilisé. » Ah.

C'est un dialogue avec un concierge très sévère et légèrement dérangé. Chaque tentative débloque une nouvelle règle, comme dans un jeu vidéo dont le niveau final aurait été écrit par un comité.

Au bout du compte, vous obtenez une séquence du genre « Tr0mb0ne_Violet!7 », fière comme un drapeau. Vous l'écrivez sur une feuille. La feuille va dans un tiroir. Le tiroir n'a pas de serrure. La sécurité informatique a fait un tour complet et elle est revenue au point de départ.

Vrai conseil : mieux vaut une longue phrase de mots pris au hasard qu'un mot court bourré de symboles. C'est plus sûr et plus facile à retenir.

PRINCIPE DE KERCKHOFFS



Un système cryptographique doit rester sûr même si l'ennemi connaît tout de son fonctionnement : le seul secret doit être la clé.

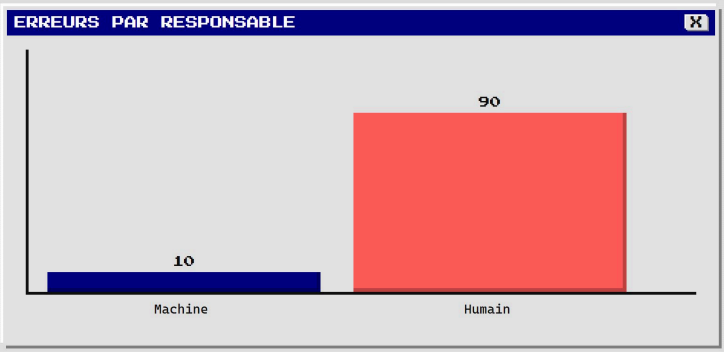
— Auguste Kerckhoffs, *La cryptographie militaire*, 1883

Le contraire s'appelle « la sécurité par l'obscurité » : le programme est sûr parce que personne ne sait comment il fonctionne. Ça dure exactement jusqu'au premier curieux qui a un après-midi de libre.

Kerckhoffs raisonnait en militaire : les appareils sont capturés, les manuels copiés, les espions corrompus. Si toute la sécurité repose sur le secret du mécanisme, un seul vol suffit pour tout perdre. Si elle repose sur la clé, on change la clé et on recommence. C'est pourquoi les algorithmes sérieux sont publics et étudiés par tous.

Mon oncle protège sa cave avec un système ultra-secret : la clé sous le pot de géraniums. Le système est obscur, mais tout le village le connaît. Dimanche, on lui a laissé un mot : « Excellent saucisson, bon millésime pour le vin. » La sécurité par les géraniums : déconseillée.

LOIS DE GILB SUR LE MANQUE DE FIABILITE



L'ordinateur se trompe toujours pareil. L'humain se trompe avec créativité.

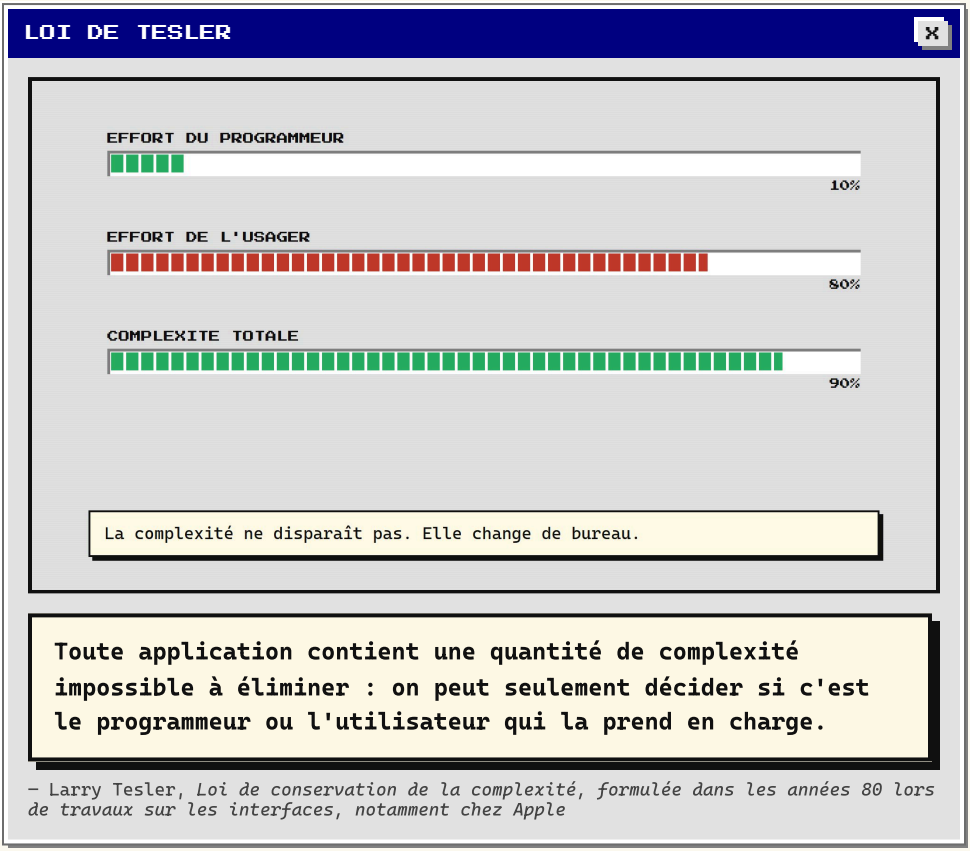
Les ordinateurs ne sont pas fiables, mais les êtres humains le sont encore moins ; et tout système qui dépend de la fiabilité humaine n'est pas fiable.

— Tom Gilb, *Lois diffusées dans les recueils des années 70 (par ex. Arthur Bloch) ; numérotation et formulations varient selon les sources (attribution en partie incertaine)*

L'ordinateur se trompe, oui. Mais il se trompe toujours de la même façon, et tôt ou tard on comprend comment. L'être humain se trompe avec créativité : aujourd'hui le mot de passe sur le post-it, demain le post-it sur l'écran, après-demain l'écran photographié et partagé dans le groupe de famille.

La leçon est une leçon d'ingénieur : ne concevez pas de systèmes qui ne marchent que si tout le monde est attentif. Les gens sont fatigués, distraits, pressés. Les contrôles automatiques, les confirmations et les sauvegardes existent pour eux.

Dans une entreprise, ils avaient une procédure de sauvegarde infaillible : chaque soir, un employé changeait le disque. Ça a marché pendant des années. Puis l'employé est parti à la retraite. Le disque est toujours là, inséré depuis 2015, fidèle comme un chien à la gare.



Toute application contient une quantité de complexité impossible à éliminer : on peut seulement décider si c'est le programmeur ou l'utilisateur qui la prend en charge.

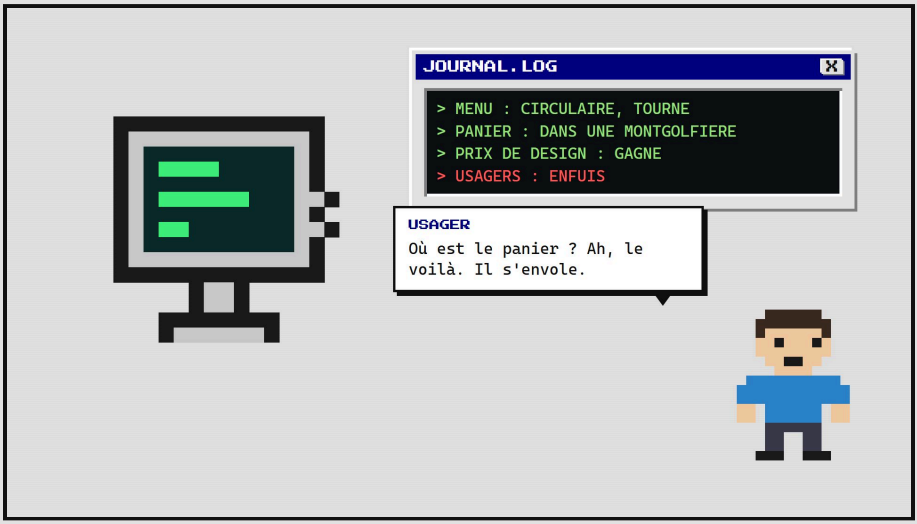
— Larry Tesler, *Loi de conservation de la complexité, formulée dans les années 80 lors de travaux sur les interfaces, notamment chez Apple*

La complexité est comme l'énergie : rien ne se crée, rien ne se perd, tout se déplace. Un formulaire avec un seul champ « simple » signifie que derrière, quelqu'un a écrit mille lignes pour comprendre ce que voulait dire l'utilisateur.

La loi est une invitation à la courtoisie : mieux vaut qu'un développeur souffre une fois que dix mille utilisateurs tous les jours. Malheureusement, la tentation inverse est forte, et naît le formulaire à quarante champs obligatoires.

Au bureau, nous avons simplifié l'imprimante : elle n'a plus qu'un seul bouton. Toute la complexité m'a été transférée, à moi qui reçois vingt-deux appels par jour pour expliquer ce que fait ce bouton. La conservation fonctionne. Moi, un peu moins.

LOI DE JAKOB



Les utilisateurs passent la plupart de leur temps sur d'autres sites, donc ils s'attendent à ce que le vôtre fonctionne comme ceux qu'ils connaissent déjà.

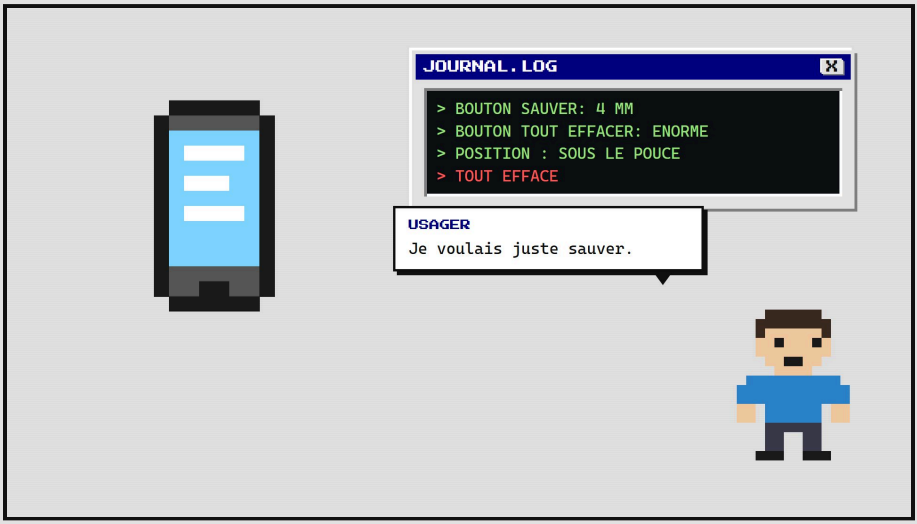
— Jakob Nielsen, *Alertbox, Nielsen Norman Group, 2000*

Le panier est en haut à droite. Le logo en haut à gauche ramène à l'accueil. La loupe sert à chercher. Aucune loi ne l'a décidé, mais des millions de personnes l'ont appris à force de l'utiliser partout. La loi de Jakob dit : ne combattez pas ces habitudes, surfez dessus.

De temps en temps arrive le designer visionnaire qui veut épater. Menu circulaire qui tourne avec le défilement. Bouton d'achat caché dans une montgolfière animée. Le site gagne un prix. Les utilisateurs s'enfuient vers un site ennuyeux où le panier est là où il doit être.

Être original, c'est très bien pour les couleurs, les photos, les textes. Dans la navigation, l'originalité est un coût que paie l'utilisateur. Un jour, j'ai mis le menu en bas à gauche pour des raisons artistiques. Le client l'a trouvé au bout d'une semaine. Il était venu me demander pourquoi son site n'avait qu'une seule page.

LOI DE FITTS



Le temps pour atteindre une cible d'un mouvement dépend de sa distance et de sa taille : plus elle est loin et petite, plus on met de temps et plus on se trompe.

— Paul Fitts, *Journal of Experimental Psychology*, 1954

Un psychologue des années cinquante étudiait les mouvements de la main, bien avant que les souris existent. Sa formule, pourtant, semble écrite pour ceux qui conçoivent des interfaces : un bouton grand et proche se clique vite, un bouton petit et lointain est une épreuve d'adresse de fête foraine.

C'est pour cela que les boutons importants devraient être grands, et les boutons dangereux, loin. Or c'est souvent l'inverse : la touche « Enregistrer » est un timbre-poste, et celle « Tout supprimer » est grande comme une porte, juste sous le pouce. Sur le téléphone, ça devient un sport extrême.

Les bords de l'écran sont des cibles infinies, parce que la souris s'y arrête toute seule. Mon oncle le prouve, lui qui ouvre toujours le menu Démarrer en lançant le curseur en bas à gauche d'un coup de poignet. Il dit que c'est une technique de pêche. Il ne rate jamais.

LOI DE HICK

CARTE DU CAFE'

- café
- cappuccino
- lait d'avoine, soja, riz
- lait d'amande, coco, épeautre
- 12 autres sortes de lait
- banc : zone de réflexion

Avant, c'était café ou cappuccino. Maintenant, devant la caisse, il y a un banc.

Le temps nécessaire pour décider croît avec le nombre d'options disponibles (de façon logarithmique, pas linéaire).

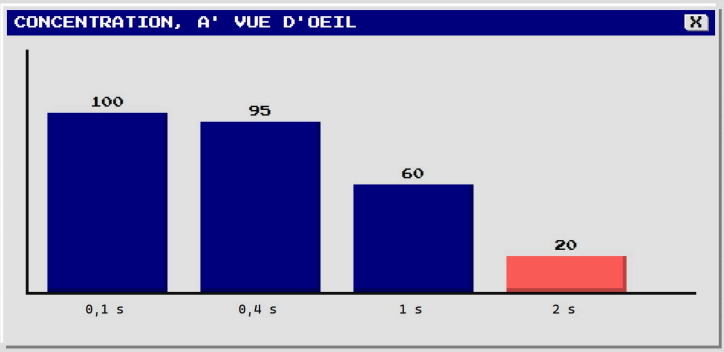
— William Edmund Hick (et Ray Hyman), *Quarterly Journal of Experimental Psychology, 1952 ; étude de Hyman, 1953 (appelée aussi loi de Hick-Hyman)*

Un menu à trois entrées se lit et se choisit en un instant. Un menu à quarante entrées, chacune avec son sous-menu, c'est un après-midi de votre vie. La loi de Hick dit que plus vous mettez d'options devant une personne, plus elle met de temps à décider. Ça croît moins que proportionnellement, mais ça croît.

C'est pourquoi les bons logiciels cachent les réglages avancés et ne montrent que l'essentiel. Et c'est pourquoi la fenêtre des options de certains logiciels, avec trente-deux onglets et deux cents cases à cocher, provoque chez l'utilisateur un regard fixe vers l'horizon.

Je le vois tous les jours au café. Avant, c'était « expresso ou crème ». Maintenant, il y a une carte avec dix-huit sortes de lait. Le barista a ajouté un banc devant la caisse. Il l'appelle la zone de traitement.

SEUIL DE DOHERTY



Au-delà du seuil, l'esprit vagabonde. Et quand il vagabonde, il va sur Internet.

La productivité augmente nettement quand l'ordinateur répond à l'utilisateur en moins de 400 millisecondes environ : sous ce seuil, aucun des deux n'attend l'autre.

— Walter J. Doherty et Arvind J. Thadhani, *« The Economic Value of Rapid Response Time »*, IBM, 1982 *(la valeur de 400 ms est surtout citée par des sources ultérieures)*

Au début des années quatre-vingt, on pensait que deux secondes d'attente étaient acceptables. Une étude d'IBM soutint le contraire : si l'ordinateur répond en moins d'une demi-seconde environ, l'utilisateur reste concentré et travaille beaucoup plus. Au-delà, l'esprit commence à vagabonder. Et quand il vagabonde, il va sur Internet.

C'est pour cela que les interfaces modernes affichent tout de suite quelque chose : une animation, une barre, un squelette gris de la page. Ce n'est pas plus rapide, mais ça a l'air vivant. L'utilisateur ne veut pas le résultat immédiat. Il veut savoir que quelqu'un l'écoute.

J'ai appliqué le seuil de Doherty à l'assistance téléphonique. Je réponds en moins de quatre cents millisecondes par « Oui, je suis là ». Ensuite, évidemment, je mets une heure à résoudre. Mais le client se sent suivi. Tout est dans la barre de progression émotionnelle.

LE CONSEIL DU SYSADMIN

Une grande partie du temps du support passe à comprendre ce qui s'est passé. Vous pouvez l'offrir à votre technicien, qui se souviendra de vous avec affection. Quand quelque chose ne va pas, préparez ces cinq informations avant d'appeler :

- **Ce que vous faisiez**, étape par étape, y compris le « j'ai touché à rien » en détail.
- **Le message exact**, photographié. Sous Windows 10 et 11, les touches Windows + Maj + S capturent une partie de l'écran.
- **Depuis quand** ça arrive, et si ça marchait avant.
- **Ce qui a changé** entre-temps : une mise à jour, un nouveau programme, un déménagement de bureau.
- **Si ça arrive toujours** ou seulement parfois, sur un seul ordinateur ou sur tous.

Avec ces réponses, beaucoup de pannes se règlent au téléphone. Les autres se règlent plus vite, parce que le technicien arrive avec le bon outil déjà en main.

CHAPITRE 4

CHAP 04 — RESEAU

Réseaux et Internet : le fil invisible

Le réseau est la chose la plus utilisée et la moins comprise de la maison moderne. Tout le monde le veut, personne ne sait où il est. Dans la box ? Dans le mur ? Dans l'air ? Ma voisine affirme qu'il passe par la prise du téléphone et qu'il se repose la nuit. Je ne l'ai pas contredite : dans certains quartiers, elle a raison.

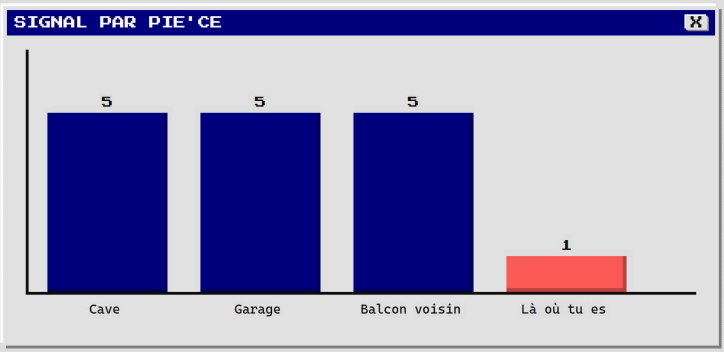
Le chapitre s'ouvre sur les lois de la connexion à la maison et au bureau : le Wi-Fi qui capte partout sauf là où vous êtes, le téléchargement qui meurt à 99 %, la visio qui se fige quand c'est votre tour. Puis il passe aux lois du grand Internet, nées entre forums, journaux en ligne et discussions de groupe. Ici la technologie compte peu et les gens comptent énormément : celui qui corrige en se trompant, celui qui titre avec un point d'interrogation, celui qui tente de cacher une chose et la rend célèbre.

Ce sont des lois qui valent aussi hors de l'écran. Celle de Brandolini, par exemple, je l'ai vue à l'œuvre à table, à une fête de village et une fois dans une salle d'attente : réfuter une ânerie demande toujours plus d'effort que de la dire. Ce chapitre ne la réfute pas. Il la documente.

CONTENU.DIR

01 Loi du Wi-Fi capricieux
02 Loi de la box à redémarrer
03 Loi de la visio décisive
04 Loi du téléchargement à 99 %
05 Loi du site qui marche partout
06 Loi de Metcalfe
07 Loi de Nielsen sur la bande passante
08 Principe de robustesse
09 Loi de Cunningham
10 Loi de Skitt
11 Loi de Betteridge sur les titres
12 Loi de Poe
13 Loi de Godwin
14 Effet Streisand
15 Loi de Brandolini

LOI DU WI-FI CAPRICIEUX



Barres de signal. Le Wi-Fi est un chat : il va où il veut.

Le Wi-Fi fonctionne parfaitement dans toutes les pièces de la maison sauf celle où vous vous trouvez.

— Anonyme, *folklore informatique*

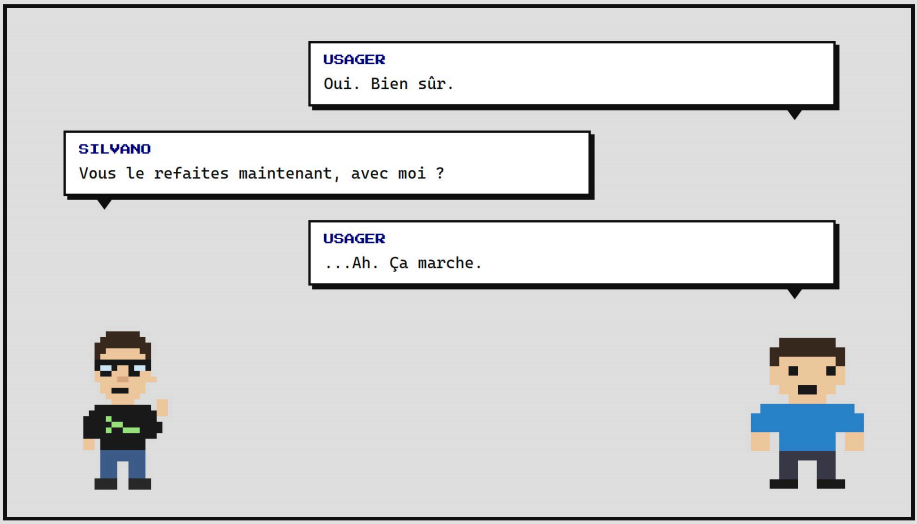
Le signal Wi-Fi est un chat. Il va où il veut. À la cave, il capte très bien, au garage aussi, sur le balcon du voisin on le voit en haute définition. Mais vous êtes au salon, et au salon il y a deux barres qui clignotent comme un appel au secours.

Alors commence le rituel. Vous vous déplacez de cinquante centimètres. Vous levez le téléphone vers le ciel comme une offrande. Vous vous asseyez sur l'accoudoir du canapé dans une position que le kiné déconseille. Ça marche. Vous bougez d'un centimètre. Ça ne marche plus.

J'ai un client qui a mis sa box dans une armoire en bois massif « pour ne pas la voir ». Maintenant, même le Wi-Fi ne la voit plus.

Conseil de technicien : la box doit être placée en position centrale, en hauteur et dégagée. Les murs épais, les miroirs et les aquariums sont ses ennemis naturels.

LOI DE LA BOX A REDEMARRER



Avant d'appeler l'assistance, personne ne redémarre la box ; après avoir appelé, le problème se résout en redémarrant la box.

— Anonyme, *folklore informatique*

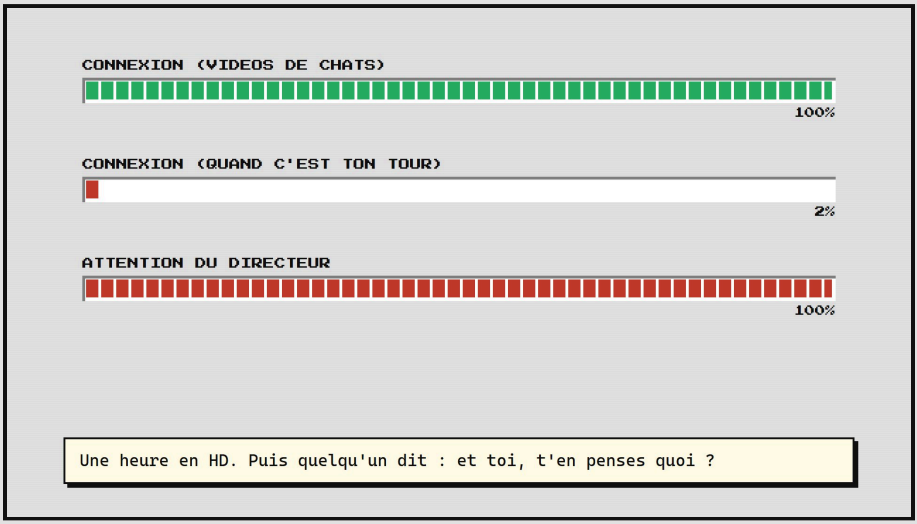
Le redémarrage de la box est le sacrement laïque de l'ère numérique. Tout le monde en connaît le pouvoir, personne n'y croit vraiment tant que ce n'est pas dit par quelqu'un avec un casque et un numéro vert.

« Vous avez essayé d'éteindre et de rallumer ? » « Bien sûr. » « Vous pouvez le refaire maintenant avec moi ? » Silence. Bruit de meubles déplacés. Petites lumières qui s'éteignent. Petites lumières qui s'allument. « Ah. Maintenant, ça marche. »

Frères et sœurs de la connexion ! La box n'est pas un appareil. C'est un état d'âme qu'il faut réinitialiser de temps en temps, comme nous le lundi matin.

Conseil de technicien : éteignez, attendez au moins trente secondes, puis rallumez et laissez à la box deux bonnes minutes pour redémarrer. Si le problème revient souvent, il faut en chercher la cause : le redémarrage est un pansement, pas le remède.

LOI DE LA VISIO DECISIVE



La connexion tient des heures de vidéos de chats, mais elle coupe exactement quand c'est à vous de parler dans la visioconférence importante.

— Anonyme, *folklore informatique*

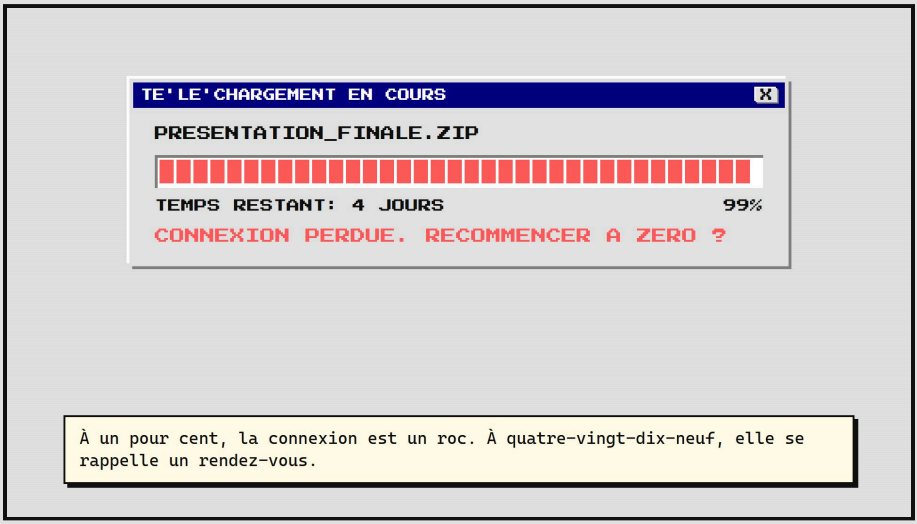
Pendant toute la réunion, la connexion a été impeccable. Vous avez vu les collègues en haute définition, entendu la respiration du directeur, compté les plantes derrière son bureau. Puis quelqu'un dit : « Et toi, tu en penses quoi ? »

À cet instant précis, votre visage se fige dans une expression indéchiffrable, entre la stupeur et l'éternuement. Votre voix devient celle d'un robot sous l'eau. Les autres vous voient ainsi pendant quarante secondes, et décident de passer à la suite.

Quand vous revenez en ligne, la réunion a déjà décidé. De votre sort. En votre absence.

Vrai conseil : pour les appels importants, si vous le pouvez, reliez l'ordinateur à la box avec un câble réseau. Le Wi-Fi est pratique, le câble est fiable.

LOI DU TELECHARGEMENT A 99 %



À un pour cent, la connexion est un roc. À quatre-vingt-dix-neuf, elle se rappelle un rendez-vous.

Un téléchargement ou un envoi s'interrompt toujours à quatre-vingt-dix-neuf pour cent, jamais à un.

— Anonyme, *folklore informatique*

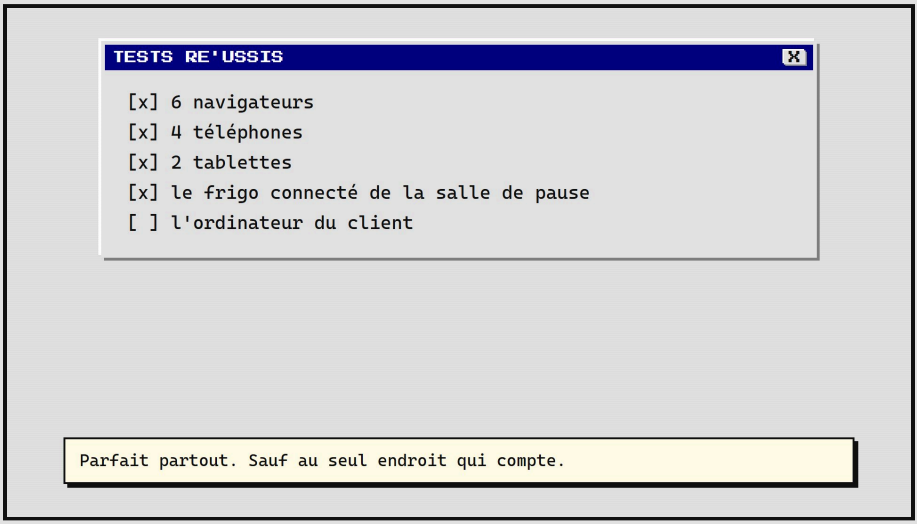
Aucun téléchargement n'a jamais échoué à un pour cent. Ce serait un échec honnête, sans douleur. Non : la barre avance lentement pendant une heure, atteint quatre-vingt-dix-neuf, vous laisse entrevoir la terre promise, puis apparaît le message « erreur réseau ».

Les scientifiques appellent ça de la cruauté. Les techniciens l'appellent « le dernier pour cent », qui dure notoirement autant que les quatre-vingt-dix-neuf autres. C'est là que l'ordinateur fait les calculs sérieux, dit-on. En réalité, il est juste en train de décider s'il vous aime.

J'ai vu des hommes adultes fixer une barre bloquée à quatre-vingt-dix-neuf pendant vingt minutes, sans respirer, comme on regarde un penalty décisif.

Conseil de technicien : pour les gros fichiers, utilisez des logiciels ou des services qui permettent de reprendre le téléchargement. Ainsi, si ça coupe, ça repart d'où ça s'était arrêté.

LOI DU SITE QUI MARCHE PARTOUT



Le site web fonctionne sur tous les ordinateurs du bureau sauf sur celui du client qui doit le valider.

— Anonyme, *folklore informatique*

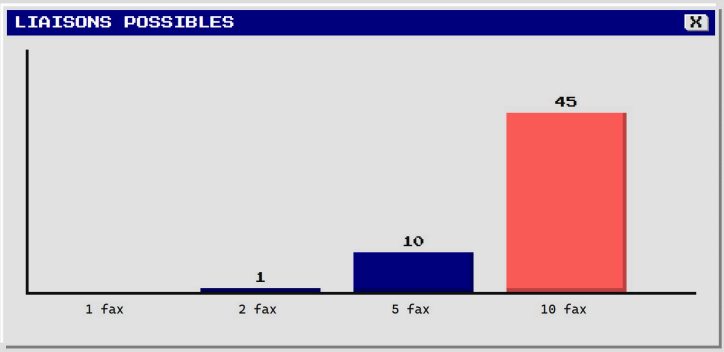
Le site a été testé sur six navigateurs, quatre téléphones, deux tablettes et le frigo connecté de la salle de pause. Tout est parfait. Puis le client l'ouvre depuis son ordinateur et voit un menu décalé à gauche et un logo grand comme un camion.

L'ordinateur du client est un lieu mystérieux. Il a un navigateur qui ne se met à jour que les jours pairs, six barres d'outils installées par un neveu en 2016 et le zoom réglé à cent cinquante pour cent « parce qu'on lit mieux ».

Et le client n'appelle jamais pour dire « sur mon ordinateur, ça s'affiche mal ». Il dit : « Le site ne marche pas. » Sur le ton de quelqu'un qui annonce une éclipse.

Vrai conseil : avant d'accuser le site, essayez de vider le cache et de remettre le zoom à cent pour cent. Ça résout plus de problèmes qu'on ne le croit.

LOI DE METCALFE



Un fax seul est un bibelot qui fait du bruit. Dix fax forment une société, avec ragots.

La valeur d'un réseau croît avec le carré du nombre d'appareils (ou de personnes) connectés : chaque nouveau nœud augmente les connexions possibles de tous les autres.

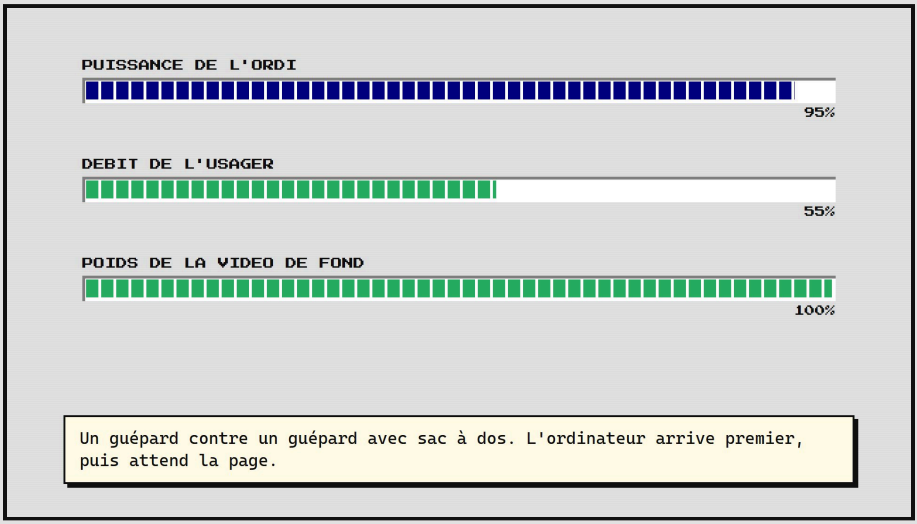
— Robert Metcalfe, *Formulée vers 1980 pour promouvoir Ethernet ; le nom « loi de Metcalfe » est de George Gilder, 1993*

Un fax tout seul ne sert à rien. C'est un bibelot qui fait du bruit. Deux fax peuvent se parler. Dix fax forment déjà une société, avec ses commérages et son papier thermique. Voilà la loi de Metcalfe : chaque nouveau connecté fait croître la valeur du réseau pour tous, et pas qu'un peu, au carré.

C'est la raison pour laquelle personne ne quitte un réseau social ou un groupe de discussion : si vous partez, vous perdez tous les autres. La valeur n'est pas dans le logiciel, elle est dans les gens piégés à l'intérieur. Avec bonheur, bien entendu.

Le revers de la médaille, je le connais bien. Le groupe de discussion de la copropriété compte quarante membres. Selon la formule, il vaut énormément. En pratique, il produit mille six cents disputes possibles sur le stationnement des vélos. Ça aussi, mathématiquement, c'est de la valeur.

LOI DE NIELSEN SUR LA BANDE PASSANTE



La vitesse de connexion d'un utilisateur averti croît d'environ 50 % par an : plus lentement que la puissance des ordinateurs.

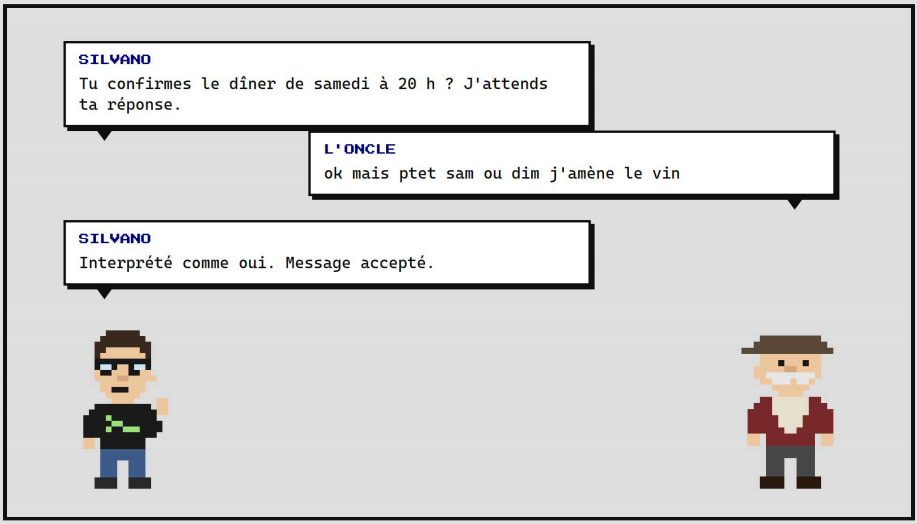
— Jakob Nielsen, *Alertbox, Nielsen Norman Group, 1998 (mise à jour en 2008 et 2019)*

Les processeurs doublent tous les deux ans, la bande passante croît de cinquante pour cent par an. Ça paraît beaucoup, mais c'est comme une course entre un guépard et un guépard avec un sac à dos. L'ordinateur arrive toujours le premier. Puis il s'assoit et attend que la page se charge.

La leçon pour ceux qui font des sites est simple : ne partez pas du principe que tout le monde a la fibre. Quelque part, il y a toujours un utilisateur dans une ferme, avec une clé 4G posée sur le rebord de la fenêtre, au seul endroit où ça capte, qui essaie d'ouvrir votre vidéo d'arrière-plan de quatre-vingt-dix mégas.

Ce client, je l'ai. Pour améliorer le signal, il a mis la clé au sommet d'une rame à haricots. Ça marche. Mais quand il y a du vent, Internet oscille.

PRINCIPE DE ROBUSTESSE



Soyez conservateur dans ce que vous envoyez et libéral dans ce que vous acceptez des autres.

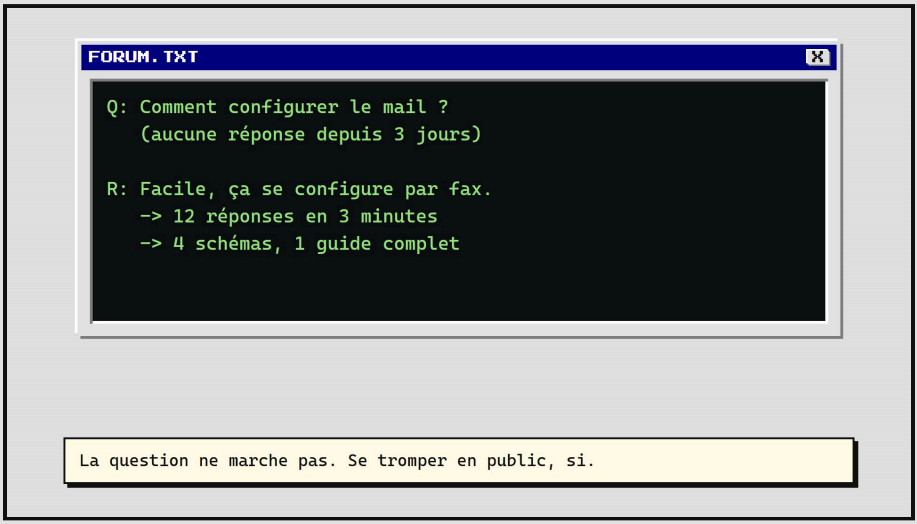
— Jon Postel, *Spécifications des protocoles Internet, RFC 760 (1980) et RFC 793 (1981)*

C'est le savoir-vivre d'Internet : quand vous parlez, appliquez le protocole à la lettre ; quand vous écoutez, soyez indulgent avec qui se trompe d'une virgule. Grâce à cette politesse, des machines écrites par des personnes différentes parviennent à se comprendre.

Il y a un revers : si tout le monde accepte les erreurs, personne ne les corrige jamais. Après trente ans de tolérance, les navigateurs lisent des pages HTML qui semblent écrites par un chat sur le clavier, et doivent continuer à le faire pour toujours.

J'ai appliqué le principe à ma vie sociale. J'envoie des messages impeccables, avec une ponctuation de notaire. J'accepte n'importe quelle réponse, même un simple pouce. Ma tante m'a répondu avec une aubergine. Je l'ai interprétée comme un « oui ». Maintenant, je suis témoin de mariage.

LOI DE CUNNINGHAM



La meilleure façon d'obtenir la bonne réponse sur Internet n'est pas de poser une question, mais d'écrire la mauvaise réponse.

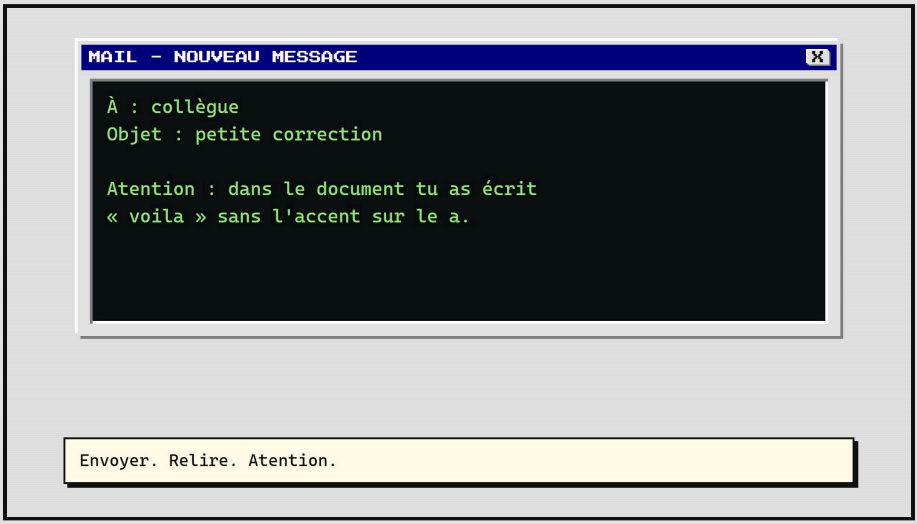
— Steven McGeady, *Baptisée par Steven McGeady en 2010 en l'honneur de Ward Cunningham, qui a pourtant nié l'avoir jamais dite (attribution contestée)*

Vous demandez sur un forum comment configurer le serveur de messagerie : silence, une botte de foin qui roule. Vous écrivez avec assurance une procédure fausse : en trois minutes arrivent douze experts, quatre schémas et une correction signée par un ingénieur à la retraite.

Ça marche parce que corriger quelqu'un procure plus de satisfaction que l'aider. C'est un moteur de connaissance alimenté par l'envie d'avoir raison. Paradoxe final : l'homme à qui elle est dédiée dit ne l'avoir jamais prononcée. La loi s'est démontrée toute seule.

Moi, je l'utilise pour tout. J'ai écrit dans un groupe que Joomla s'installe par fax. J'ai maintenant un guide complet, à jour, avec captures d'écran. Et aussi un fax, qu'un utilisateur m'a envoyé pour prouver que ça ne marche pas.

LOI DE SKITT



Tout message qui corrige l'erreur d'autrui contiendra, presque à coup sûr, au moins une erreur à lui.

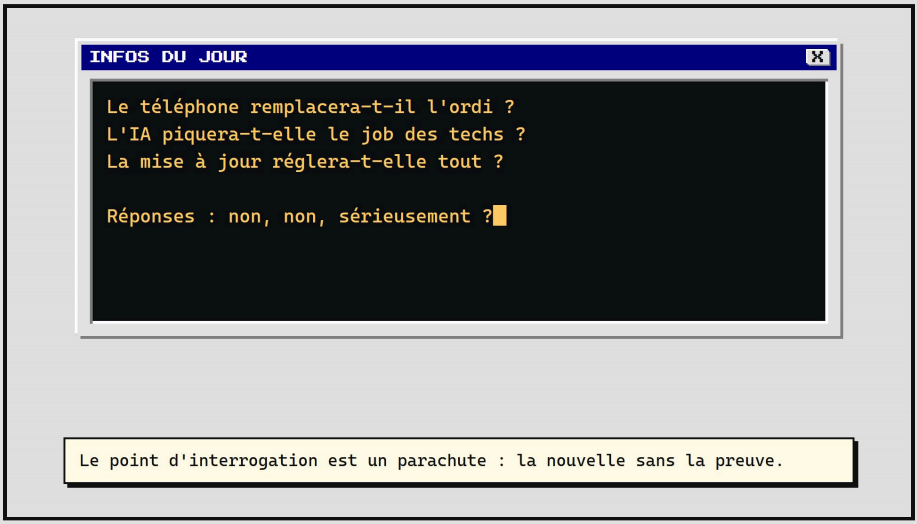
— Anonyme (pseudonyme « Skitt »), *Usenet, groupe alt.usage.english, vers 1999 (attribution incertaine ; variantes antérieures connues comme « loi de Muphry »)*

Vous écrivez un e-mail au collègue : « Attention, dans le document tu as écrit “language” avec un u en trop. » Vous envoyez. Vous relisez. Vous avez écrit « Attantion ». Le collègue répond par un seul mot, et ce mot est parfait.

La loi de Skitt est une loi de la nature de la correction. L'univers ne tolère pas l'orgueil orthographique : chaque fois que vous jouez les donneurs de leçons, une lettre rebelle se détache du clavier et va se cacher à l'endroit le plus visible de votre message. Ça vaut aussi pour le code : le correctif qui répare le bug du collègue introduit un bug à vous.

Par prudence, je ne corrige plus personne. J'ai délégué à un correcteur automatique. Hier, il a changé « serveur » en « sueur ». Le client a lu « la sueur est tombée » et m'a demandé si j'allais bien.

LOI DE BETTERIDGE SUR LES TITRES



À tout titre de journal qui se termine par un point d'interrogation, on peut répondre « non ».

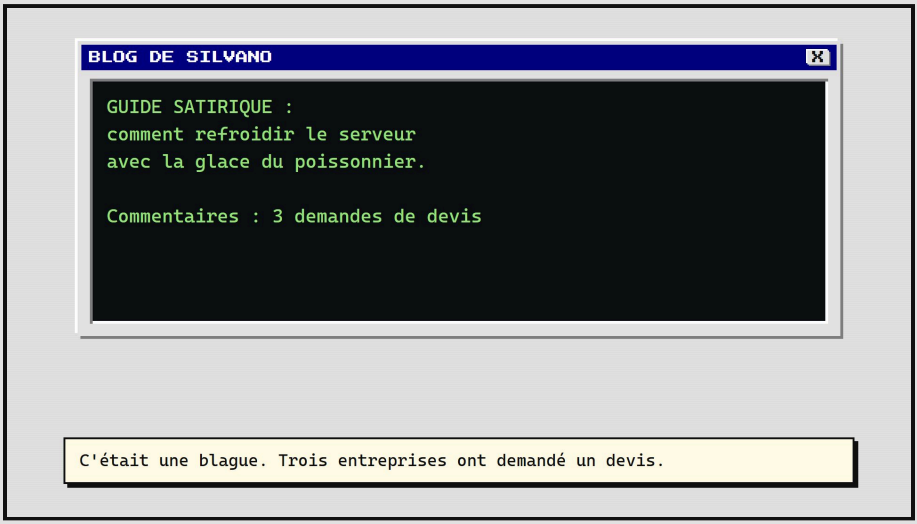
— Ian Betteridge, *Article sur son blog, février 2009 (idée qui circulait déjà sous d'autres noms dans les années 80)*

« Ce nouveau téléphone remplacera-t-il l'ordinateur ? » Non. « L'intelligence artificielle va-t-elle voler le travail des techniciens ? » Non, au contraire, c'est à nous de la configurer. « La nouvelle mise à jour résoudra-t-elle tous les problèmes ? » Je vous en prie.

La loi de Betteridge dévoile une ruse ancienne : si le journaliste avait des preuves pour un oui, il écrirait une affirmation. Le point d'interrogation est un parachute. Il permet de lancer la nouvelle fracassante sans devoir la prouver, et d'encaisser les clics de ceux qui espèrent.

J'ai fait une expérience sur mon blog. J'ai publié « La disquette va-t-elle revenir à la mode ? ». Trois mille visites. Puis j'ai publié « La disquette ne reviendra pas à la mode ». Deux visites, dont une de ma mère. Le point d'interrogation, ai-je compris, est le vrai moteur de l'édition numérique.

LOI DE POE



Sans signal clair de l'intention de l'auteur, la parodie d'une position extrême est indiscernable d'une opinion sincère du même type.

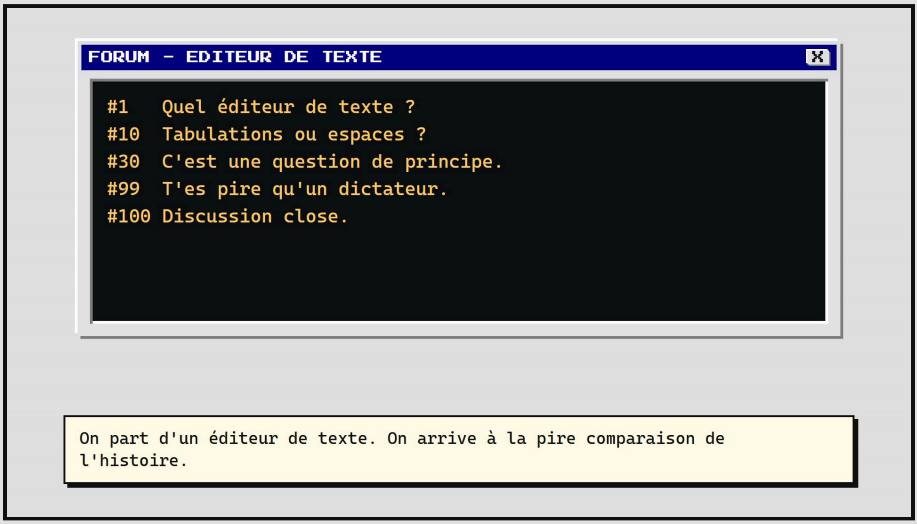
— Nathan Poe, *Forum christianforums.com, 2005 (un avertissement similaire remonte à Usenet, 1983)*

Vous écrivez sur Internet que la meilleure façon de protéger son mot de passe est de l'imprimer et de le coller sur l'écran, comme ça les voleurs ne le cherchent pas là où c'est évident. C'est une blague. Trois personnes vous remercient. Une le fait vraiment. Une autre lance une formation payante.

Voilà la loi de Poe : en ligne, l'ironie n'a pas de ton de voix, et certaines opinions réelles sont si extrêmes que la parodie n'arrive plus à les dépasser. C'est pour ça que les smileys sont nés. Ce sont nos sourcils levés numériques.

Un jour, j'ai publié un guide satirique sur la façon de refroidir le serveur avec la glace de la poissonnerie. Trois entreprises m'ont écrit pour demander un devis. Une a déjà commandé les daurades. Elle dit que les logs sentent la mer, maintenant.

LOI DE GODWIN



Plus une discussion en ligne s'allonge, plus la probabilité que quelqu'un fasse une comparaison avec Hitler ou les nazis s'approche de la certitude.

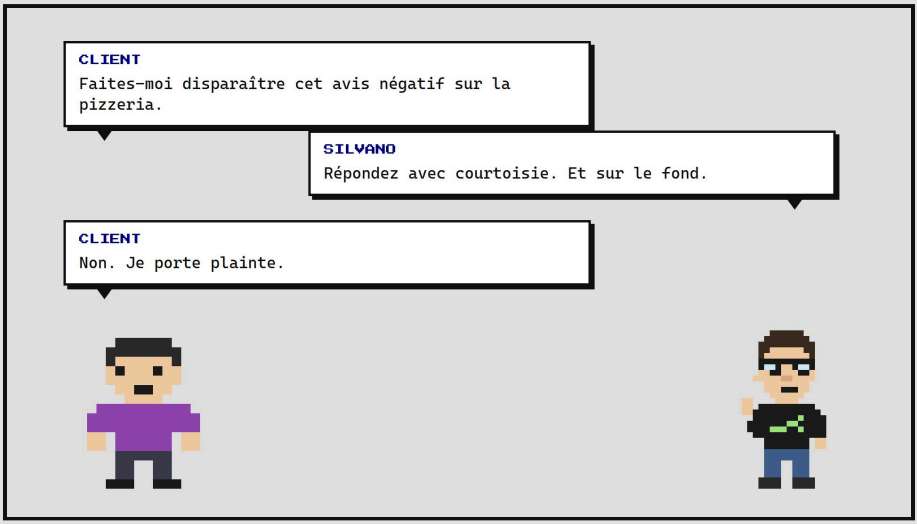
— Mike Godwin, *Usenet*, *1990*

Toute discussion en ligne commence poliment. Quelqu'un demande quel éditeur de texte utiliser. Au dixième message, on parle de tabulations contre espaces. Au trentième, de principes moraux. Au centième, quelqu'un compare son adversaire au pire dictateur de l'histoire. Pour un éditeur de texte.

La loi de Godwin ne dit pas que ça arrive, elle dit que c'est inévitable, comme la gravité ou comme le câble HDMI toujours du mauvais côté. Sur les vieux forums valait aussi une règle non écrite : le premier qui en arrive à cette comparaison a perdu la discussion.

Je modère un forum de passionnés d'imprimantes. Nous avons tenu trois cent vingt messages. Puis quelqu'un a qualifié le pilote d'une certaine marque de régime totalitaire. Pour être honnête, il n'avait pas tout à fait tort. Mais le fil a été fermé quand même.

EFFET STREISAND



Chercher à cacher, supprimer ou censurer une information en ligne finit par la rendre bien plus visible et répandue.

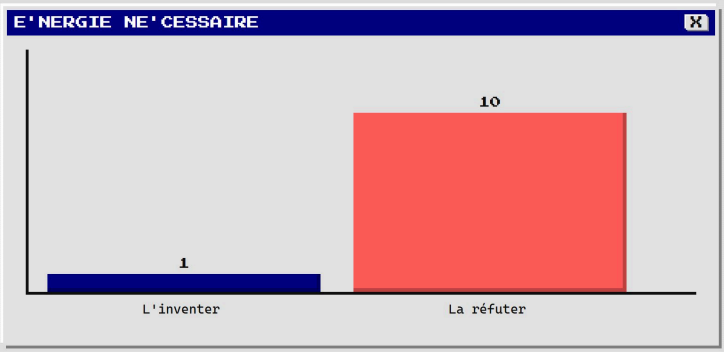
— Mike Masnick, *Techdirt, janvier 2005 (inspiré d'une affaire de 2003 sur une photo aérienne)*

Le nom vient d'une affaire de 2003 : un procès pour faire retirer d'une archive en ligne la photo aérienne d'une villa, prise pour documenter l'érosion de la côte. Avant le procès, très peu de gens avaient vu la photo. Après, la moitié du monde l'a vue. Un blogueur en tira le nom d'un phénomène.

Internet fonctionne comme un chat : si vous lui dites de ne pas regarder une boîte, il ne regardera qu'elle. Chaque demande de suppression est une enseigne lumineuse qui dit « ici, il y a quelque chose d'intéressant ». Les copies, les archives et les captures d'écran se multiplient plus vite que n'importe quel avocat.

Un client m'a demandé de faire disparaître un avis négatif sur sa pizzeria. Je lui ai conseillé de répondre aimablement. Il a préféré menacer de procès. Maintenant, l'avis est le plus lu du département. La pizza, entre-temps, s'est améliorée. Mais personne ne le sait.

LOI DE BRANDOLINI



Trois secondes pour l'inventer. Deux heures, quatre liens et un graphique pour la réfuter.

L'énergie nécessaire pour réfuter une ânerie est d'un ordre de grandeur supérieure à celle qui a servi à la produire.

— Alberto Brandolini, *Message sur Twitter, janvier 2013 (connue aussi comme principe d'asymétrie du baratin)*

Écrire « la 5G fait faner les géraniums » demande trois secondes. Expliquer pourquoi c'est faux demande deux heures, quatre liens, un graphique et la patience d'un moine. Pendant ce temps, la phrase sur les géraniums a fait le tour de trois groupes de discussion familiaux.

La loi, formulée par un programmeur italien, décrit une asymétrie cruelle : inventer coûte peu, vérifier coûte cher. C'est la raison pour laquelle les intox gagnent les courses de vitesse, et pour laquelle tout admin sys a passé au moins un dîner à expliquer que non, l'ordinateur n'a pas attrapé le virus par la prise électrique.

J'ai trouvé une parade. Quand un client me transfère une intox, je réponds par une intox opposée, tout aussi courte. Elles s'annulent. Je me suis déjà fait une réputation de personne bien informée. J'ai aussi quelques plaintes pour géraniums, mais c'est en cours de vérification.

LE CONSEIL DU SYSADMIN

Quand « Internet rame », avant de vous disputer avec votre fournisseur, faites le test du câble. Branchez un ordinateur directement sur la box avec un câble réseau et mesurez le débit avec un test en ligne. Puis refaites le test en Wi-Fi, dans la pièce où vous travaillez d'habitude.

- Si **c'est lent même avec le câble**, le problème vient de la ligne ou de la box : appelez le fournisseur, chiffres en main.
- Si **c'est rapide avec le câble et pas en Wi-Fi**, la ligne est innocente. Le problème est le signal dans la maison : position de la box, murs, interférences.

Deux détails utiles. Les box modernes émettent sur deux bandes : le 5 GHz est plus rapide mais traverse moins bien les murs, le 2,4 GHz porte plus loin mais il est plus encombré. Et si les pièces sont grandes, un système mesh ou un point d'accès relié par câble fonctionne bien mieux que les répéteurs bon marché. Parfois, déplacer la box de deux mètres vaut un changement d'abonnement.

CHAPITRE 5

CHAP 05 — DEBUG

La chasse au bug

Ici, on entre au laboratoire. Le bug, pour qui ne le fréquente pas, est une erreur dans un programme. Le mot est plus vieux que les ordinateurs, mais il existe un journal de bord de 1947 avec une vraie mite collée dessus, trouvée à l'intérieur d'une machine. Depuis, la chasse ne s'est jamais arrêtée, et la mite a eu énormément de petits-enfants.

Les lois de ce chapitre racontent le bug dans son habitat naturel. Il disparaît quand le collègue arrive, se cache dans la dernière ligne que vous vérifiez, attend le vendredi après-midi, se présente dix minutes après la mise en ligne. Parfois, s'il survit assez longtemps, les utilisateurs s'attachent à lui et il devient une fonctionnalité.

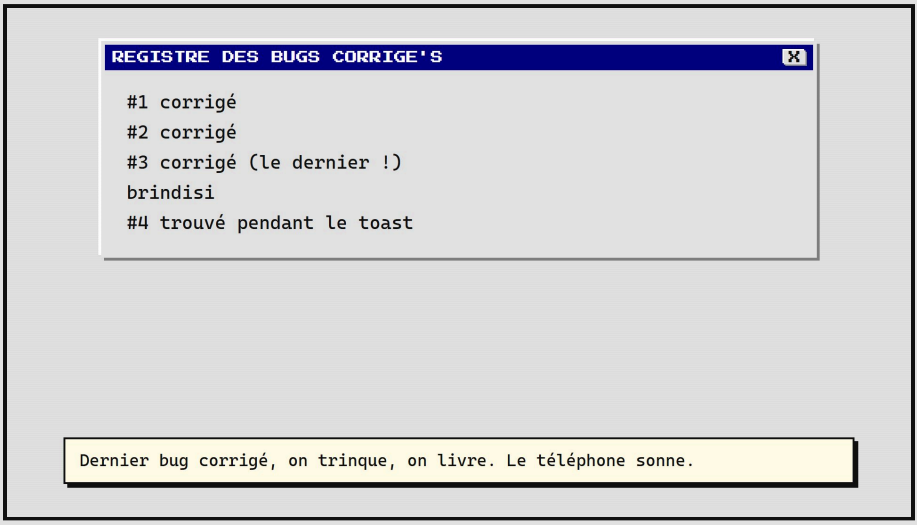
Qui ne programme pas trouvera ici la réponse à la question qu'il se pose depuis des années : pourquoi le programme que j'utilise tous les jours fait-il parfois des choses bizarres ? Réponse courte : parce qu'il y a toujours un autre bug. Réponse longue : ce chapitre.

Qui programme, en revanche, se reconnaîtra à chaque page. Je m'excuse d'avance pour les nuits que je vais vous rappeler.

CONTENU.DIR

01 Loi de Lubarsky sur l'entomologie cybernétique
02 Loi du « ça marche sur mon PC »
03 Loi du collègue thaumaturge
04 Loi de la ligne innocente
05 Loi de Kernighan
06 Loi de la correction prolifique
07 Loi du bug évolué
08 Loi du bug ponctuel
09 Loi du vendredi après-midi
10 Loi de la partie non testée
11 Loi du « ça compile, donc ça marche »
12 Loi de la donnée de test
13 Loi du message oraculaire
14 Loi du calendrier traître
15 Loi de Linus
16 Loi de Hyrum

LOI DE LUBARSKY SUR L'ENTOMOLOGIE CYBERNETIQUE



Il y a toujours un bug de plus.

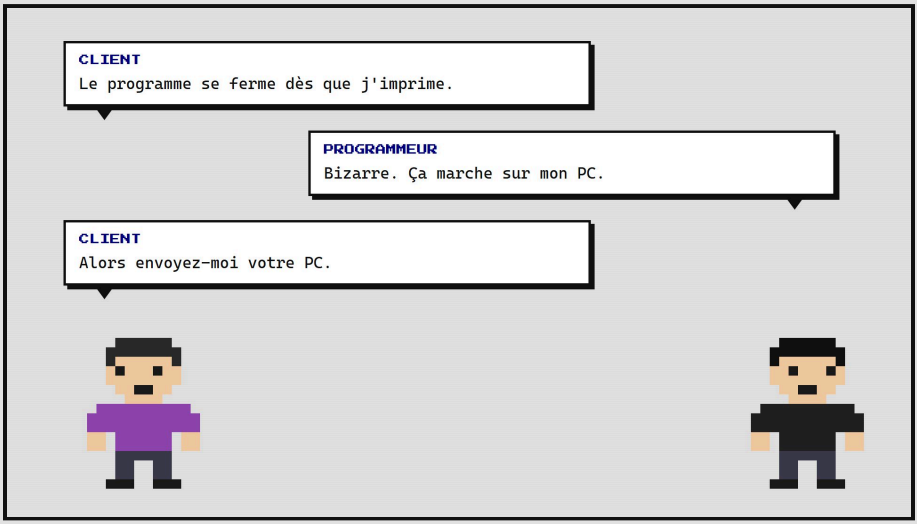
— Anonyme, *Dicton ayant circulé dans les recueils d'humour informatique (par ex. listes Multics et Usenet) ; identité de Lubarsky non vérifiée (attribution incertaine)*

Sept mots, et la carrière de tout programmeur est résumée. Vous corrigez le dernier bug, vous trinquez, vous livrez. Le téléphone sonne. Il y avait un dernier-dernier bug.

Le sens est une sage humilité : aucun programme non trivial n'est exempt d'erreurs. On peut les rendre rares, on peut les rendre inoffensives, mais croire les avoir toutes attrapées est le bug le plus dangereux de tous.

J'ai appelé un désinsectiseur pour le serveur. Il a pulvérisé quelque chose sur le boîtier, a dit « garantie six mois » et il est parti. Le lendemain, le logiciel de gestion imprimait les factures en cyrillique. Je l'ai rappelé. Il m'a dit que cette espèce n'était pas couverte. Entomologie cybernétique : la science exacte du toujours.

LOI DU « ÇA MARCHE SUR MON PC »



Toute erreur signalée par le client est impossible à reproduire sur l'ordinateur du programmeur, où tout marche parfaitement.

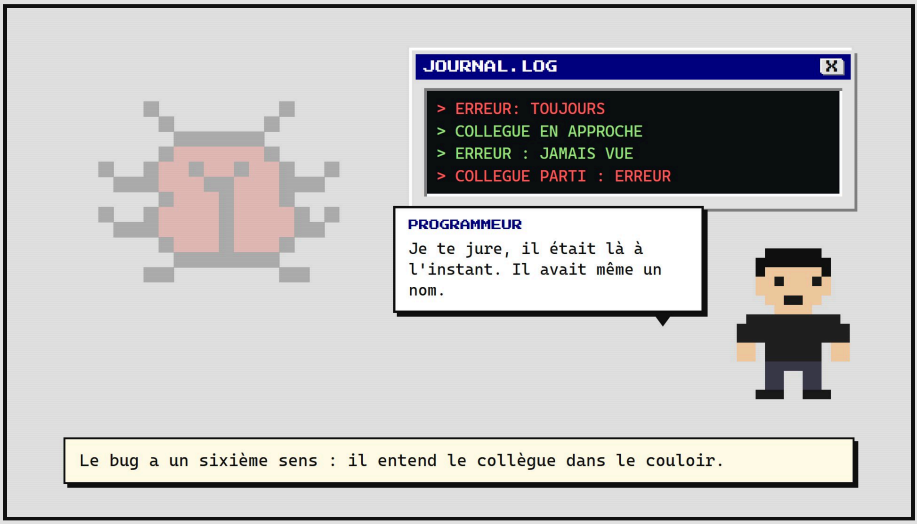
— Anonyme, *folklore informatique*

C'est la phrase la plus prononcée de l'histoire de l'informatique, plus que « vous avez essayé de redémarrer ? ». Le programmeur la dit avec une sincérité touchante : sur son ordinateur, le programme marche vraiment. Le problème, c'est que le client n'habite pas dans son ordinateur.

L'ordinateur du développeur est un écosystème protégé, une serre tropicale avec toutes les bonnes bibliothèques, les bons réglages, la bonne langue. L'ordinateur du client est une jungle avec un Windows de 2016, trois antivirus en guerre les uns contre les autres et l'heure réglée sur le fuseau d'Honolulu.

Il fut une période où j'ai sérieusement envisagé d'expédier mon PC aux clients avec le logiciel. J'aurais résolu quatre-vingt-dix pour cent des problèmes. Puis j'ai compris que le client aurait appelé quand même, pour dire que mon PC sur son bureau ne marchait pas.

LOI DU COLLEGUE THAUMATURGE



Le bug qui vous tourmente depuis des heures disparaît à l'instant où vous appelez un collègue pour le lui montrer, et réapparaît dès qu'il retourne à son bureau.

— Anonyme, *folklore informatique*

Dans chaque bureau, il y a un collègue doté de pouvoirs. Il ne programme pas mieux que vous, il n'a pas fait plus d'études. Simplement, quand il s'approche de votre écran, les bugs se mettent au garde-à-vous et font semblant de ne pas exister.

« Montre-moi », dit-il. Vous refaites exactement les mêmes manipulations, avec la voix tremblante de quelqu'un qui signale un fantôme. Le programme marche. Il vous tapote l'épaule avec l'air du médecin de famille. Il s'en va. Le bug réapparaît, et vous sourit.

Les programmeurs ont même un nom pour ces erreurs timides : des bugs qui changent de comportement quand on les observe. Moi, j'ai réglé le problème à la racine. J'ai imprimé une photo du collègue grandeur nature et je l'ai accrochée derrière ma chaise. Ça marche dans soixante pour cent des cas.

LOI DE LA LIGNE INNOCENTE

SUSPECTS, DANS L'ORDRE

1. le code du collègue
2. la lib externe
3. l'OS
4. rayons cosmiques
5. ma ligne (impossible)

Six heures d'enquête. Le coupable était le président de la commission.

Le bug se trouve toujours dans la dernière ligne que vous vérifiez, et c'est celle dont vous étiez absolument sûr qu'elle était correcte.

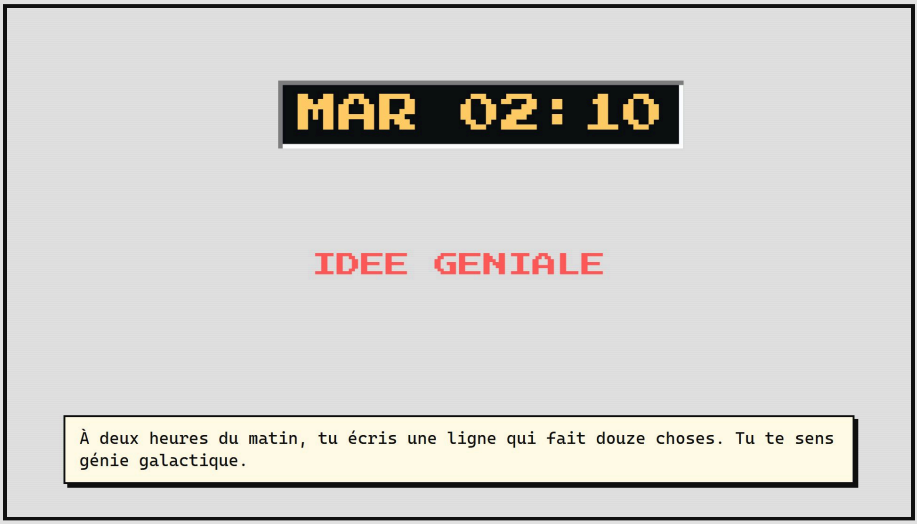
— Anonyme, *folklore informatique*

La chasse au bug suit un rituel précis. On soupçonne d'abord le code des collègues, puis la bibliothèque externe, puis le système d'exploitation, puis le compilateur, enfin les rayons cosmiques. Ce n'est qu'en dernier, à contrecœur, qu'on regarde sa propre ligne. Celle qui est parfaite. Celle écrite avec fierté.

Et le voilà, l'erreur. Un point-virgule mal placé, un égal à la place d'un double égal. Six heures d'enquête pour un caractère. Une commission d'enquête parlementaire tout entière pour découvrir que le coupable était le président de la commission.

Il existe aussi une explication logique, qui gâche la poésie : on arrête de chercher quand on l'a trouvé, donc il est toujours au dernier endroit où l'on regarde. Mais inutile de le dire à un programmeur à deux heures du matin. Lui sait que le code se moque de lui. Et, sincèrement, il a raison.

LOI DE KERNIGHAN



Déboguer est deux fois plus difficile qu'écrire le code. Donc, si vous écrivez le code de la façon la plus astucieuse possible, vous n'êtes par définition pas assez astucieux pour le déboguer.

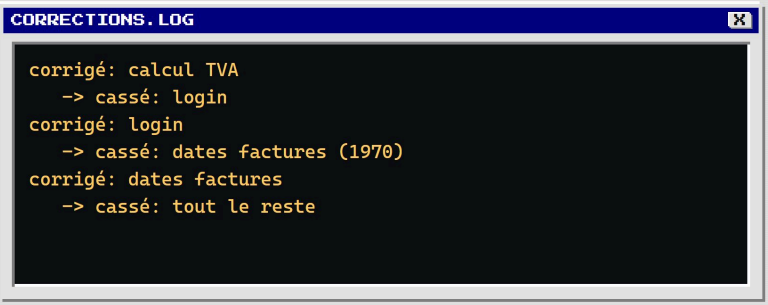
— Brian Kernighan, *The Elements of Programming Style (avec P. J. Plauger), 2e éd., 1978*

C'est un syllogisme impitoyable. À deux heures du matin, inspiré, vous écrivez une ligne qui fait douze choses à la fois. Vous vous sentez un génie galactique. Le lendemain matin, la ligne ne marche pas et le génie de la nuit ne répond pas au téléphone.

Le vrai message est : écrivez simple. Le code astucieux est une dette envers vous-même du futur, qui sera plus fatigué, plus vieux et avec moins de café.

Moi, j'ai réglé le problème avec méthode : j'écris le code à 50 % de mon intelligence, comme ça le débogage me réussit à 100 %. Les collègues disent que ça se voit. Je le prends comme un compliment à l'ingénierie, pas à moi.

LOI DE LA CORRECTION PROLIFIQUE



Le logiciel, ce sont des vases communicants : on pousse ici, ça sort là. Par terre.

Tout bug corrigé à la hâte en génère au moins deux nouveaux, de préférence dans une partie du programme qui n'avait rien à voir.

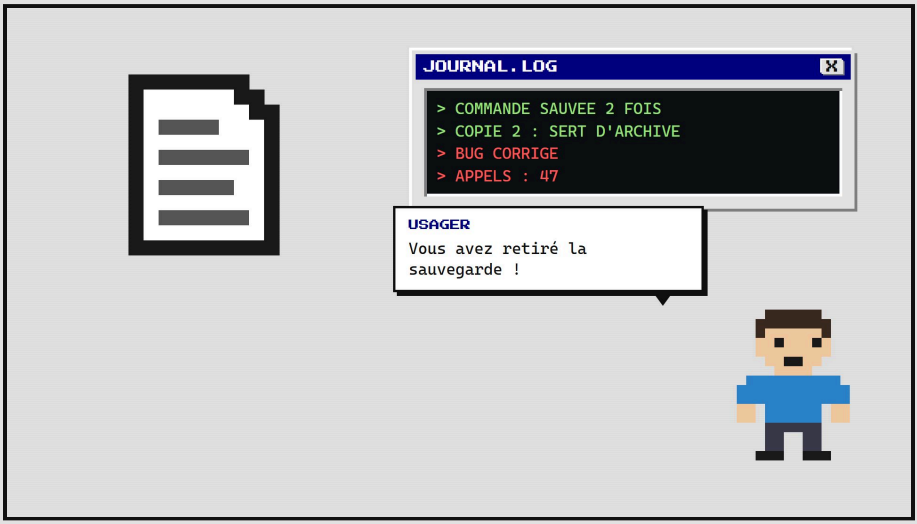
— Anonyme, *folklore informatique*

Les bugs sont comme les têtes de l'hydre des légendes : on en coupe une, il en repousse deux. À une différence près : les têtes de l'hydre, au moins, repoussaient au même endroit. Les nouveaux bugs préfèrent apparaître à l'autre bout du programme, là où personne ne les cherche.

Vous corrigez le calcul de la TVA, et la connexion cesse de marcher. Vous réparez la connexion, et les factures sortent datées de 1970. Le logiciel est un système de vases communicants : on pousse d'un côté, ça sort de l'autre, en général sur le carrelage.

Le gourou de la qualité, dans ces cas-là, prononce le mot magique : « tests automatisés ». Il a raison. Mais le bug urgent arrive toujours quand les tests automatisés sont « presque prêts ». J'aborde la correction express dans le même esprit qu'on change une ampoule dans une vieille maison : avec confiance, et l'extincteur à portée de main.

LOI DU BUG EVOLUE



Tout bug suffisamment avancé est indiscernable d'une fonctionnalité.

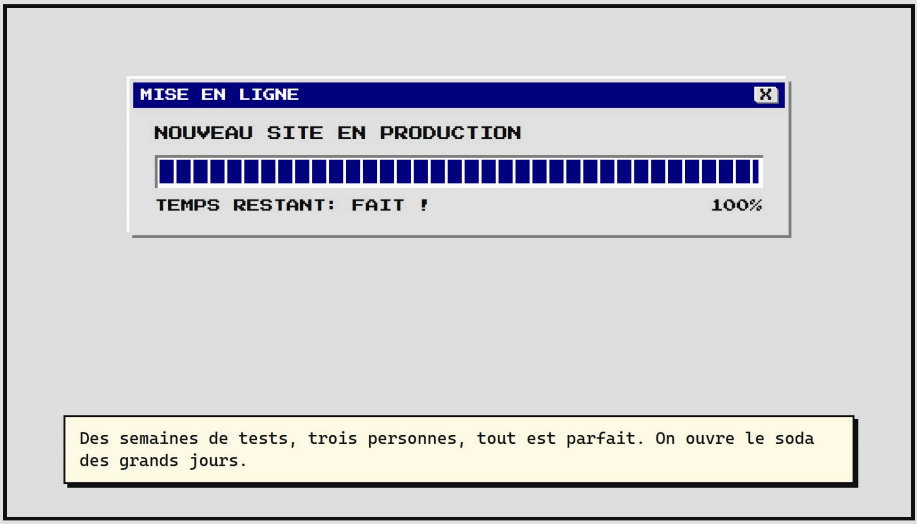
— Anonyme, *folklore informatique*

De temps en temps, une erreur survit si longtemps qu'elle prend racine. Les utilisateurs la découvrent, s'y habituent, la trouvent pratique. À ce stade, la corriger devient un acte de violence.

Exemple réel, raconté avec le respect qui s'impose : un logiciel de gestion qui, par erreur, enregistrait deux fois le brouillon de la commande. Les employés avaient appris à utiliser la deuxième copie comme archive. Quand nous l'avons corrigé, ils ont appelé en masse : « Vous avez supprimé la fonction de sauvegarde. »

De là vient la célèbre phrase que tout programmeur prononce au moins une fois dans sa vie, menton levé et regard vers l'horizon : « Ce n'est pas un bug, c'est une fonctionnalité. » C'est une formule magique, comme abracadabra. Elle ne résout rien, mais elle transforme la honte en marketing. Et le marketing, chacun le sait, ne se corrige pas.

LOI DU BUG PONCTUEL



Le bug critique est découvert dix minutes après la mise en ligne, jamais dix minutes avant.

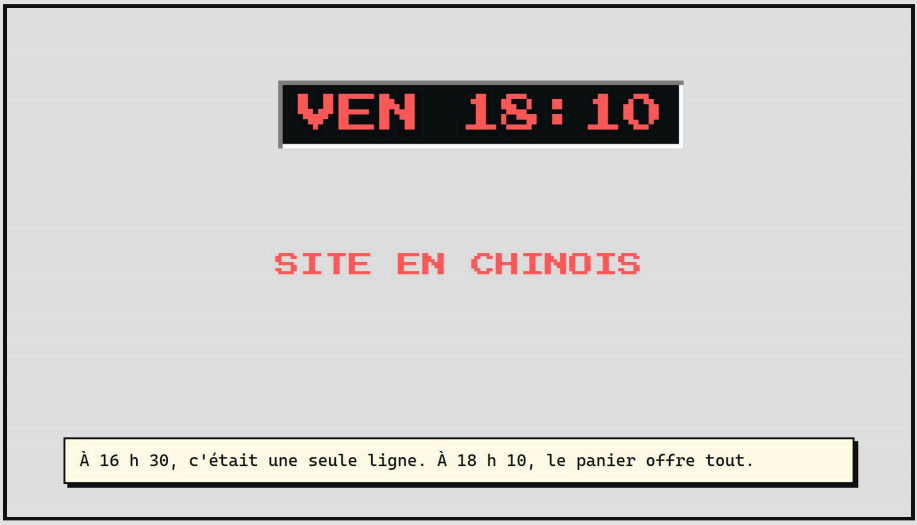
— Anonyme, *folklore informatique*

Pendant des semaines, le programme est testé, retesté, contrôlé par trois personnes différentes. Tout va bien. Puis on appuie sur le bouton « publier », on débouche le soda des grands moments, et à cet instant précis quelqu'un dit : « Euh, pardon… »

Ce n'est pas de la malchance, c'est un phénomène presque physique. Tant que le code est en test, l'esprit est en mode défensif et ne voit rien. Dès que le code est public et irréversible, l'esprit se détend et devient soudain génial. Il voit tout. Trop tard, mais il voit tout.

J'ai essayé de tromper le destin : je fais semblant de publier, j'attends dix minutes, puis je publie pour de vrai. Ça a marché deux fois. La troisième fois, j'ai trouvé le bug à la dixième minute de la fausse publication et je l'ai fêté comme un but. Puis est arrivé le deuxième bug, le vrai.

LOI DU VENDREDI APRES-MIDI



Une modification mise en production le vendredi après-midi cassera après six heures du soir, quand tous les collègues sont déjà sur l'autoroute.

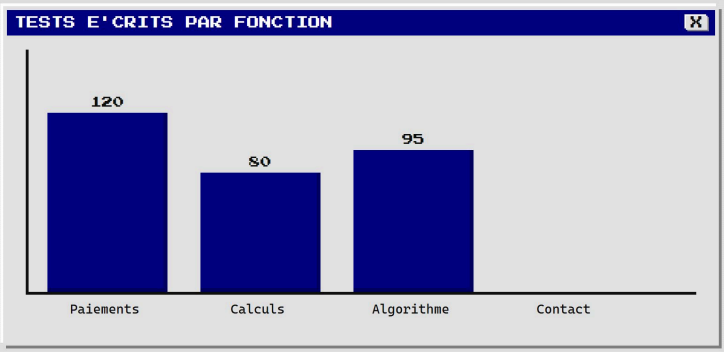
— Anonyme, *folklore informatique*

Dans le calendrier de l'informaticien, le vendredi après-midi est une zone sacrée et maudite, comme les terres frontalières des sagas de fantasy. Celui qui publie une mise à jour à ces heures-là défie les anciens dieux de la production, et les anciens dieux ne pardonnent pas.

Le scénario est toujours le même. À seize heures trente : « C'est une toute petite modif, une seule ligne. » À dix-sept heures : tout est vert, tout est parfait. À dix-huit heures dix : le site affiche la page d'accueil en chinois et le panier offre la marchandise.

Pendant ce temps, sur l'autoroute, le collègue qui connaît le système a mis son téléphone en mode avion pour raisons spirituelles. Chez nous, à l'agence, règne un décret non écrit : le vendredi, on ne fait que des choses réversibles. Dépoussiérer le clavier, lire la documentation, réfléchir. Surtout réfléchir, ça ne part jamais en production.

LOI DE LA PARTIE NON TESTEE



Forteresse côté paiements. Porte de derrière fermée à la ficelle.

La partie du programme qui casse en production est toujours celle que personne n'avait pensé à tester, parce qu'elle était trop simple pour se tromper.

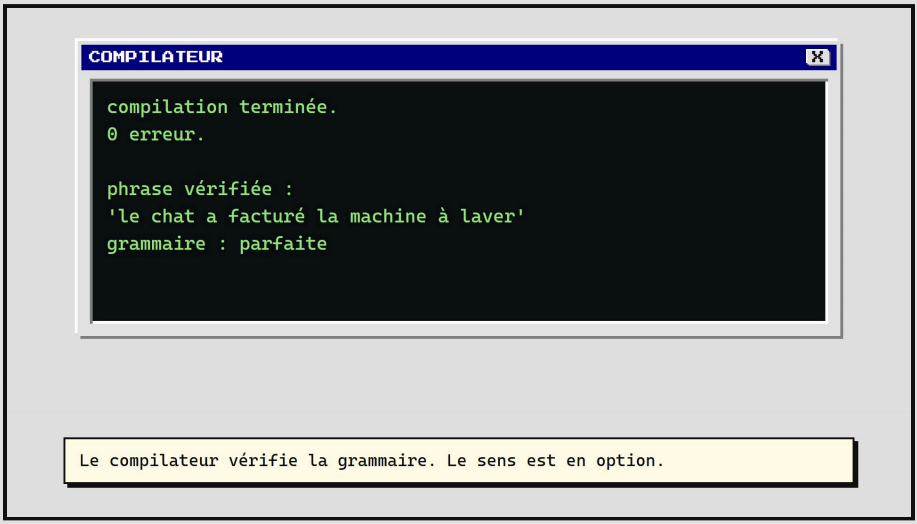
— Anonyme, *folklore informatique*

Les tests se concentrent toujours sur les parties difficiles : calculs complexes, algorithmes, paiements. Sur celles-là, on bâtit des forteresses. Puis le site s'écroule à cause du bouton « Contactez-nous », que personne n'avait jamais cliqué parce que « ce n'est qu'un formulaire ».

C'est la version informatique du château fort aux murailles épaisses de trois mètres, avec la porte de derrière fermée par une ficelle. L'ennemi n'escalade pas les murs. L'ennemi, tout simplement, frappe derrière.

Le prophète de la qualité répète depuis des années le même sermon : « Ce qui n'est pas testé est cassé, simplement vous ne le savez pas encore. » C'est une phrase qui fait peur, parce qu'elle est vraie. Maintenant, je teste aussi les choses banales, une par une, avec le zèle d'un notaire. Le client me demande pourquoi ça prend si longtemps. Je lui réponds que je vérifie la porte de derrière.

LOI DU « ÇA COMPILE, DONC ÇA MARCHE »



Si le code compile sans erreur, le programmeur fatigué le considérera prêt à être livré.

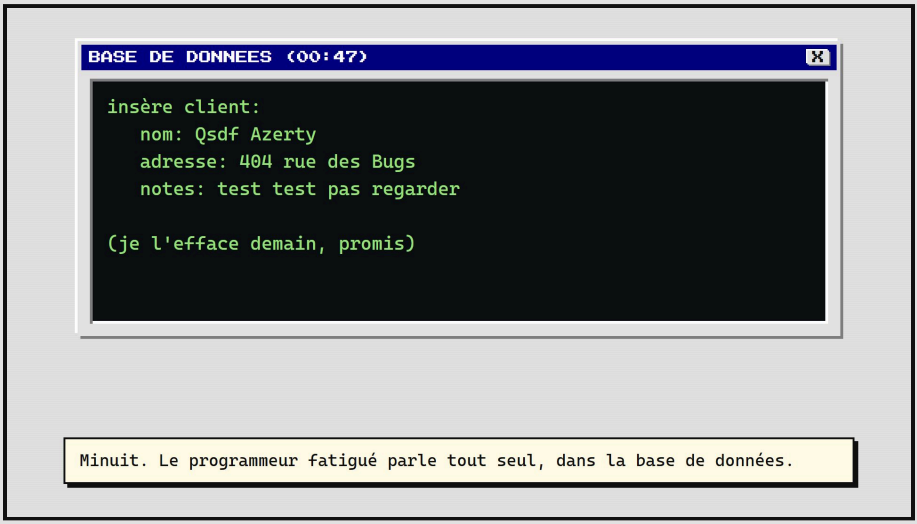
— Anonyme, *folklore informatique*

Compiler, c'est traduire le code en programme exécutable. Le compilateur vérifie que la grammaire est correcte, pas que le programme a un sens. C'est comme un correcteur d'épreuves qui approuve avec enthousiasme la phrase « le chat a facturé la machine à laver ».

Et pourtant, à sept heures du soir, devant le message « compilation terminée », quelque chose se déclenche dans le cerveau humain. Une voix intérieure dit : c'est bon. Le code part chez le client comme un navire qui quitte le port avec les voiles montées à l'envers.

Dans beaucoup de bureaux circule la version courte de la devise : « Ça compile ? On expédie. » On la prononce en riant. Puis on l'applique. Je l'ai fait une seule fois, dans un passé lointain. La machine à laver, au sens figuré, tourne encore. Et j'ai appris à essayer le programme avant, au moins une fois.

LOI DE LA DONNEE DE TEST



La donnée de test la plus embarrassante saisie pendant le développement apparaîtra, tôt ou tard, devant le client le plus important.

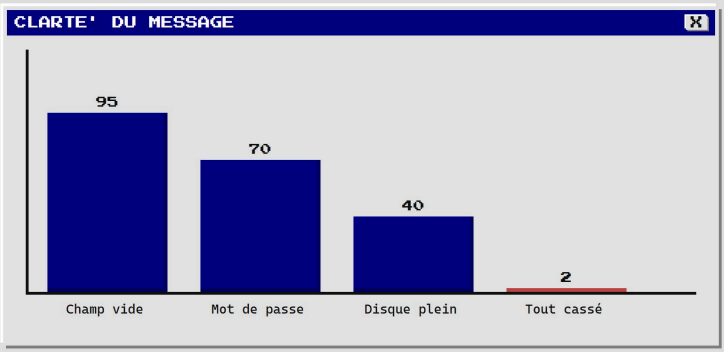
— Anonyme, *folklore informatique*

Pendant le développement, personne ne saisit de données réalistes. On écrit « azer », « test test », « Jean Toto », « produit qui n'existe pas ». Ce sont les bouche-trous du programmeur fatigué, sa façon de parler tout seul à minuit.

Puis vient la présentation officielle. Salle de réunion, vidéoprojecteur, directeur général au premier rang. On ouvre la liste des clients et en tête, en gras, il y a M. Azer Qsdf, domicilié « 404 rue des Bugs ». Silence. Quelqu'un tousse.

La règle que j'ai apprise à mes dépens est simple : écrivez chaque donnée de test comme si le client devait la lire. Parce qu'il la lira. Aujourd'hui, dans mes environnements de test, il n'y a que des noms sobres et respectables, du genre « Client Exemple SARL ». Un peu plus triste, mais beaucoup moins de sueurs froides.

LOI DU MESSAGE ORACULAIRE



Plus l'erreur est grave, moins elle parle. Un oracle à court de forfait.

Plus une erreur est grave, plus le message qui l'annonce est vague, générique et incompréhensible.

— Anonyme, *folklore informatique*

Les petites erreurs parlent clairement : « Le champ E-mail n'est pas valide. » Les erreurs catastrophiques, elles, s'expriment comme les oracles antiques : « Une erreur inattendue s'est produite. » Code 0x8000FFFF. Fin de la communication.

Inattendue par qui, d'ailleurs ? C'est une phrase écrite par un programmeur qui, des années plus tôt, avait parfaitement prévu que quelque chose tournerait mal, mais n'avait pas envie de dire quoi. C'est un mot laissé sur le frigo : « Il s'est passé un truc. Salut. »

Les clients me les lisent au téléphone avec la voix de quelqu'un qui récite une prophétie. J'écoute en silence, j'acquiesce, puis je prononce la formule sacrée que tout admin sys connaît : « Vous avez essayé de redémarrer ? » Ça ne marche pas toujours. Mais c'est la seule réponse à la hauteur de la question.

LOI DU CALENDRIER TRAITRE

DATES QUE LE PROGRAMME DE'TESTE

- 29 février
- la nuit du changement d'heure
- 1er janvier
- 30 février (de son invention)

Trois cent soixante-deux jours parfaits. Les trois autres, roman gothique.

Tout programme qui gère des dates marchera toute l'année, sauf le 29 février, la nuit du changement d'heure et le premier jour de janvier.

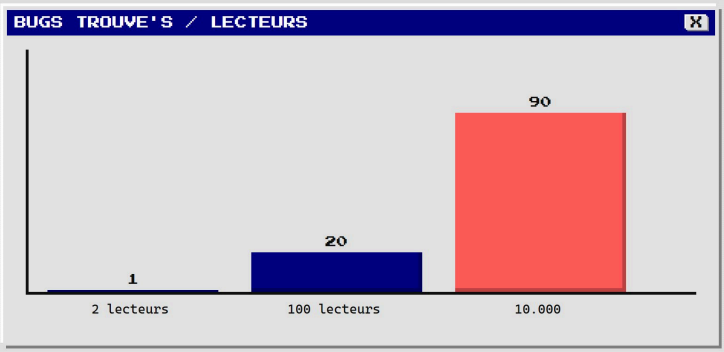
— Anonyme, *folklore informatique*

Le temps a l'air simple. Soixante secondes, soixante minutes, vingt-quatre heures. Puis un programmeur essaie de le gérer dans le code et découvre l'horreur : années bissextiles, fuseaux horaires, heure d'été, mois de vingt-huit jours, pays qui changent d'avis.

Le résultat, c'est un logiciel qui marche à merveille trois cent soixante-deux jours par an. Les trois autres, il fait des choses de roman gothique : réservations dans le passé, fiches de paie avec une heure fantôme, échéances fixées au 30 février.

Une nuit de fin octobre, un de mes logiciels de gestion a enregistré deux fois le même pointage, à deux heures et demie, parce que cette heure avait vraiment existé deux fois. L'employé était présent à deux endroits en même temps. Le directeur voulait le promouvoir. J'ai corrigé le bug, avec le regret de celui qui a détruit un miracle.

LOI DE LINUS



Code ouvert et beaucoup de lecteurs : tôt ou tard, le défaut sort.

Avec suffisamment d'yeux, tous les bugs remontent à la surface.

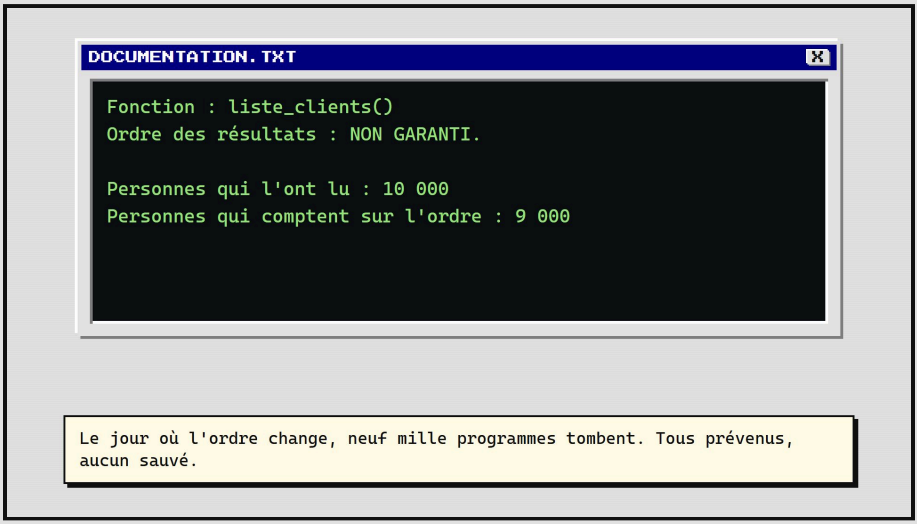
— Eric S. Raymond, *The Cathedral and the Bazaar, 1997 (essai), 1999 (livre) ; baptisée en l'honneur de Linus Torvalds*

L'idée est magnifique : si le code est ouvert et que des milliers de personnes le lisent, quelqu'un trouvera le défaut. C'est la raison pour laquelle le logiciel libre peut rivaliser avec les grandes entreprises, et souvent gagner.

Le détail gênant, c'est le mot « suffisamment ». Il faut aussi que les yeux regardent vraiment. Il existe des bibliothèques utilisées par la moitié de la planète et lues, dans les faits, par deux bénévoles fatigués et un chat.

J'ai essayé de l'appliquer au bureau : j'ai projeté le code au mur pendant la pause café. Vingt yeux. Aucun bug trouvé, mais trois collègues ont signalé que la police était moche et un a demandé si le vidéoprojecteur était neuf. Le bug est toujours là, bien éclairé.

LOI DE HYRUM



Avec suffisamment d'utilisateurs, tout comportement observable de votre système sera utilisé par quelqu'un, quoi que vous ayez promis dans la documentation.

— Hyrum Wright, *Observation née dans l'ingénierie logicielle de Google, années 2010 (hyrumslaw.com)*

Vous écrivez dans la documentation : « L'ordre des résultats n'est pas garanti. » Dix mille personnes lisent. Neuf mille construisent leur programme en supposant que l'ordre est toujours le même. Le jour où vous le changez, la moitié d'Internet tombe.

La loi nous dit que le vrai contrat d'un logiciel n'est pas celui qui est écrit, mais tout ce qu'il fait, défauts compris. Corriger un bug peut casser celui qui avait bâti toute une activité sur ce bug.

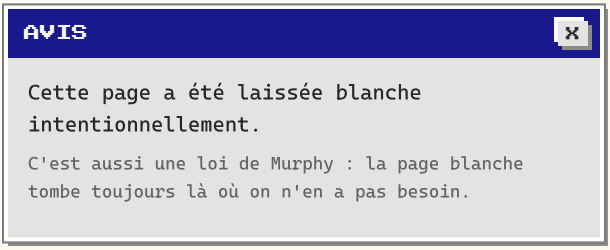
Un jour, j'ai corrigé une faute de frappe dans un message d'erreur. Le lendemain, un client furieux m'a appelé : son script cherchait justement ce mot mal orthographié pour savoir que quelque chose avait mal tourné. J'ai rétabli la faute. Elle est maintenant classée monument historique.

LE CONSEIL DU SYSADMIN

La chasse au bug n'est pas un art divinatoire, c'est une méthode. La mienne, éprouvée sur pas mal de nuits, est celle-ci.

- **D'abord le reproduire.** Notez les étapes exactes qui font apparaître l'erreur. Un bug que vous ne savez pas faire revenir, vous ne saurez même pas dire si vous l'avez corrigé.
- **Lire les logs.** Souvent, le programme a déjà écrit quelque part ce qui a mal tourné. Ça paraît évident : c'est l'étape qu'on saute le plus souvent.
- **Couper en deux.** Si ça marchait hier et plus aujourd'hui, cherchez à mi-chemin ce qui a changé. Git a même une commande pour ça, `git bisect`, qui trouve la modification coupable en divisant chaque fois le champ par deux.
- **L'expliquer à voix haute.** À un collègue, ou à un objet sur le bureau. En racontant le problème ligne par ligne, la ligne fautive se dénonce souvent toute seule.

Et quand vous l'avez trouvé, écrivez un test qui le rattrapera s'il revient. Les bugs rentrent toujours à la maison, comme les chats.


AVIS
Cette page a été laissée blanche intentionnellement.
C'est aussi une loi de Murphy : la page blanche tombe toujours là où on n'en a pas besoin.

CHAPITRE 6

CHAP 06 — CODE

L'art d'écrire du code (et de le regretter)

Si le chapitre précédent était la rubrique des faits divers, celui-ci est le traité de philosophie. Les lois qui suivent ne parlent pas d'erreurs isolées, mais de la façon d'écrire du logiciel : comment on nomme les choses, combien on commente, quand on optimise, pourquoi le code malin d'aujourd'hui est la migraine de demain.

Beaucoup portent la signature de grands noms de la programmation, des gens qui ont construit des langages, des systèmes et des livres sur lesquels des générations ont étudié. Il est rassurant de découvrir qu'eux aussi, un jour, se sont retrouvés devant un programme écrit de leur main quelques mois plus tôt sans y comprendre goutte. Le génie ne vaccine pas.

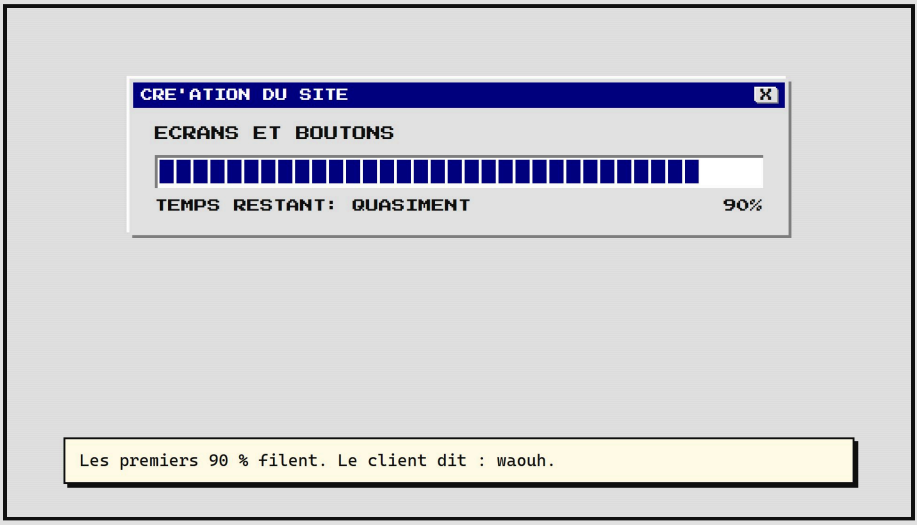
Le fil conducteur est unique, et je le dis tout de suite pour que les pressés puissent sauter à la fin : la simplicité est difficile, mais c'est la seule chose qui tient dans le temps. Le reste n'est que commentaire. Menteur, probablement.

Pour ceux qui ne programment pas : pas de panique. Chaque loi est expliquée avec des exemples de tous les jours, et vous comprendrez pourquoi votre logiciel de gestion met deux ans à déplacer un bouton. Ce n'est pas de la paresse. C'est de l'archéologie.

CONTENU.DIR

01 Règle des quatre-vingt-dix/quatre-vingt-dix
02 Loi des deux choses difficiles
03 Loi des deux problèmes
04 Loi d'Eagleson
05 Loi du commentaire menteur
06 Loi de la documentation posthume
07 Loi du code adopté
08 Loi d'Osborn
09 Axiome de Flon
10 Principe de Hoare
11 Optimisation prématurée
12 Règles de Pike
13 Loi de Déméter
14 Loi des abstractions qui fuient
15 Dixième règle de Greenspun
16 Loi de Wadler
17 Lois de Lehman

REGLE DES QUATRE-VINGT-DIX/QUATRE-VINGT-DIX



Les premiers 90 % du code demandent 90 % du temps de développement. Les 10 % restants demandent les autres 90 %.

— Tom Cargill, *Bell Labs ; popularisée par Jon Bentley dans la rubrique Programming Pearls, Communications of the ACM, 1985*

Les mathématiques disent 180 %. Le programmeur dit : exactement. Le projet est toujours « presque fini » pendant une période qui dure autant que le projet entier.

Les premiers 90 %, c'est la belle partie : les écrans, les boutons, le client qui dit « waouh ». Les derniers 10 %, c'est quand on découvre qu'un nom de famille peut contenir une apostrophe, que quelqu'un est né un 29 février et que la TVA change justement la semaine de la livraison.

Dans mon bureau, il y a un site à 98 % depuis 2017. Chaque année, nous fêtons l'anniversaire du « il ne manque presque rien ». Nous avons un gâteau. Les bougies sont à 98 % : à moitié allumées, mais avec beaucoup d'enthousiasme.

LOI DES DEUX CHOSES DIFFICILES

TARIFS.PHP

```
prix = cache.lire('café')
> 1,00 euro
(le nouveau tarif dit 1,20)
> vide cache
> 1,00 euro
(le cache y est attaché)
```

Le cache se souvient de tout. Surtout de ce qu'il devrait oublier.

En informatique, il n'existe que deux problèmes vraiment difficiles : invalider le cache et nommer les choses.

— Phil Karlton, *maxime attribuée à Karlton à l'époque de Netscape, rapportée par Tim Bray et par Martin Fowler (« TwoHardThings »)*

Ceux qui ne programment pas pensent que les parties difficiles sont les mathématiques, la sécurité, l'intelligence artificielle. Erreur. Les deux vraies montagnes sont celles-ci : faire oublier une vieille chose à l'ordinateur, et décider comment appeler une nouvelle chose.

Le cache, c'est la mémoire rapide qui met des données de côté pour ne pas aller les rechercher à chaque fois. Ça marche très bien, jusqu'au jour où il vous affiche fièrement le prix d'hier. Le nom, lui, est une torture philosophique : « donnees », « donnees2 », « donneesNouvelles », « donneesNouvellesVraies ». J'ai vu des équipes entières bloquées un après-midi sur un nom de variable, comme un concile médiéval.

Il existe aussi une variante folklorique, qui ajoute un troisième problème : les erreurs d'un en plus ou en moins dans les comptages. Les choses difficiles sont deux, mais personne n'a jamais réussi à les compter correctement.

LOI DES DEUX PROBLEMES

CONTROLE_EMAIL

```
regle = /^[a-z.]+@+[a-z]*\.fr$/

test('toto@@.fr')          -> VALIDE
test('cabinet@comptable.com')
                           -> REFUSEE
```

Une formule pour reconnaître les e-mails. Accepte toto, refuse le comptable.

Face à un problème, certaines personnes pensent : « Je vais le résoudre avec des expressions régulières. » Elles ont alors deux problèmes.

— Jamie Zawinski, *message sur Usenet de 1997, retrouvé par Jeffrey Friedl sur le blog regex.info (2006)*

Les expressions régulières sont des formules pour chercher du texte, surpuissantes et écrites dans une langue qui ressemble à un chat ayant marché sur le clavier. Elles servent, par exemple, à vérifier qu'une adresse e-mail est valide. Le résultat est une ligne de symboles qu'aucun être humain ne sait plus lire au bout de dix minutes, y compris celui qui l'a écrite.

Le premier problème était de trouver les e-mails. Le deuxième, c'est de comprendre pourquoi la formule accepte « toto@@.fr » et refuse l'expert-comptable. Le troisième, car il s'en ajoute toujours un, c'est que le seul collègue qui la comprenait a changé de boulot.

Je les utilise avec la prudence qu'on réserve à un sortilège trouvé au grenier. Elles marchent, mais chaque fois que je les modifie, il me semble entendre un coup de tonnerre au loin. Et en général, effectivement, le serveur tombe.

LOI D'EAGLESON

VIEUX_PROJET.PHP

```
function f($toto2) {
  $x = $toto2 * 3.14 + 7;    // ?
  // à corriger plus tard
  return f2($x, true, true, false);
}
```

Aucun commentaire, variables mystérieuses. Qui a fait ça ?

Le code que vous avez écrit vous-même et que vous n'avez pas regardé depuis six mois ou plus pourrait tout aussi bien avoir été écrit par quelqu'un d'autre.

— Anonyme, *Dicton répandu chez les programmeurs ; origine et identité d'Eagleson non vérifiées (attribution incertaine)*

Vous ouvrez un vieux fichier. Qui a écrit ce truc ? Des variables appelées « toto2 », aucun commentaire, une fonction de trois cents lignes. Vous vérifiez l'auteur. C'est vous. Vous-même d'il y a six mois, un parfait inconnu avec vos identifiants.

La loi enseigne que le premier destinataire de la documentation n'est pas le collègue : c'est vous dans le futur, qui aurez tout oublié. Écrire du code clair est un geste de gentillesse envers un inconnu qui vous ressemble beaucoup.

Désormais, je traite mon moi du passé comme un consultant externe. Je lui écris des e-mails de réclamation. Il ne répond jamais. Hier, j'ai trouvé un de ses commentaires : « À arranger après ». Après quand ? Nous sommes dans l'après. L'après, c'est moi.

LOI DU COMMENTAIRE MENTEUR

REMISES.PHP

```
// calcule la remise de 10 %
fonction remise(commande):
   envoie_mail(directeur)
   remise = 30%
   // pas touche
```

Le commentaire raconte une intention ancienne. Le code, un accident moderne.

Quand le commentaire et le code ne sont pas d'accord, ils ont probablement tort tous les deux.

— Anonyme, *folklore informatique*

Les commentaires sont les phrases que les programmeurs laissent dans le code pour expliquer ce qu'il fait. En théorie, c'est une gentillesse envers ceux qui viendront après. En pratique, ce sont comme les panneaux d'une route qui a été déplacée entre-temps.

Le code change, le commentaire reste. On trouve ainsi écrit « calcule la remise de dix pour cent » au-dessus d'une fonction qui envoie un e-mail au directeur. Qui a raison ? Personne. Le commentaire raconte une intention ancienne, le code un accident moderne.

Il existe aussi l'école de pensée opposée, celle des programmeurs fiers qui ne commentent jamais : si c'était difficile à écrire, ça doit être difficile à comprendre. Moi, je suis entre les deux. Je commente toujours, mais avec des phrases du genre « ne pas toucher ». Ça n'explique rien, mais ça sauve beaucoup de vies.

LOI DE LA DOCUMENTATION POSTHUME

MANUEL UTILISATEUR V2 – PAGE 1

1. Cliquer sur le bouton vert à gauche
2. Ouvrir le menu Fichier
3. Imprimer sur disquette

Version installée : 4.0
Bouton vert : supprimé il y a deux ans

Le code court. La doc marche, et s'arrête souvent pour prendre des photos.

La documentation, quand elle existe, décrit avec une grande précision la version précédente du programme.

— Anonyme, *folklore informatique*

La documentation technique est un genre littéraire à part entière, entre le roman historique et le testament. Elle raconte avec force détails un monde qui n'existe plus : des boutons déplacés, des menus renommés, des fonctions supprimées entre-temps lors d'une réunion de quatre minutes.

La raison est simple. Le code court, la documentation marche. Quand quelqu'un écrit enfin le manuel de la version deux, les programmeurs se disputent déjà sur la quatre.

J'ai un client qui conserve le manuel de son logiciel de gestion dans une vitrine, comme une relique. De temps en temps, il l'ouvre et lit les instructions à voix haute, d'un ton solennel. Aucune ne correspond plus à l'écran. Mais il dit que ça l'aide à se souvenir du monde tel qu'il était quand les boutons étaient à gauche. Et moi, franchement, je le comprends.

LOI DU CODE ADOPTE

SOLUTION_DU_FORUM

```
// trouvé sur un forum, année inconnue
// aucune idée, mais ça sert
x = x + 0
attends(0 seconde)
répète 1 fois:
   marche()
```

Lignes offertes par un inconnu. On les accepte avec gratitude et sans questions.

Le code copié sur Internet marche à la perfection jusqu'à ce que vous essayiez de comprendre pourquoi il marche.

— Anonyme, *folklore informatique*

Tout programmeur possède une archive secrète de lignes empruntées aux forums : des solutions écrites par des inconnus, à une autre époque, pour un autre problème. On les colle, elles marchent, et on les regarde avec la gratitude de quelqu'un qui reçoit un rein d'un donneur anonyme.

Les ennuis commencent quand, par scrupule, on essaie de les comprendre. On lit, on réfléchit, on enlève une ligne qui semble inutile. Tout s'effondre. On remet la ligne. Ça ne marche plus, même avec la ligne. Le code adopté est vexé et a décidé de ne pas vous pardonner.

Depuis, je traite ces lignes comme les anciens traitaient les amulettes : respect, distance, aucune question. Dans mon code, il y a encore des morceaux qui portent le commentaire « je ne sais pas pourquoi, mais c'est nécessaire ». Ce sont les lignes les plus stables de tout le projet. La connaissance, parfois, est surestimée.

LOI D'OSBORN

CONFIG.PHP

```
const ANNEE_EN_COURS = 2019;
const TAUX_TVA = 19.6;     // puis 20, puis ?
var temperature = 0;       // jamais changée

echo ANNEE_EN_COURS;
> 2019
```

La variable n'a pas bougé depuis le début. La constante a changé d'avis plusieurs fois.

Les variables ne varient pas ; les constantes ne restent pas constantes.

— Anonyme, *Dicton répandu dans l'humour informatique ; origine et identité d'Osborn non vérifiées (attribution incertaine)*

En théorie, la variable change et la constante reste immobile. En pratique, la variable reste clouée à la mauvaise valeur depuis sa création, tandis que la constante « taux de TVA » a changé quatre fois en dix ans.

Derrière la boutade, il y a une leçon : les choses que vous croyez immuables, comme les taxes, les indicatifs téléphoniques, les formats de date, sont celles qui changeront. Et celles qui devraient se mettre à jour souvent restent bloquées à cause d'une erreur d'initialisation.

Dans mon code, il y a une constante appelée ANNEE_COURANTE. Elle vaut 2019. Personne n'a le courage d'y toucher. Nous avons décidé que, pour le logiciel, on est toujours en 2019. On y est bien. Les prix sont plus bas.

AXIOME DE FLON



Il n'existe pas, il n'a jamais existé et il n'existera jamais de langage de programmation dans lequel il soit difficile d'écrire de mauvais programmes.

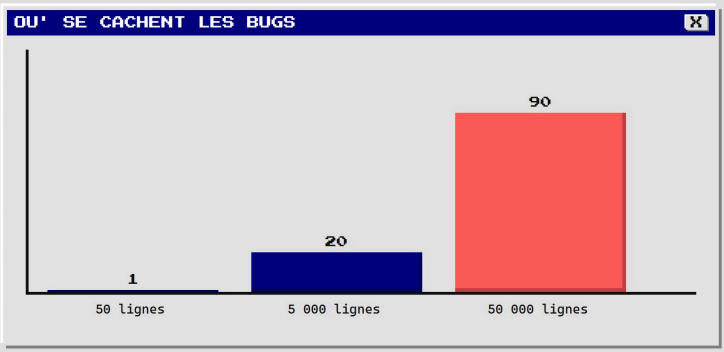
— Lawrence Flon, *On Research in Structured Programming*, *SIGPLAN Notices*, octobre 1975

Chaque nouveau langage arrive avec une promesse messianique : « Avec moi, le code sera enfin propre ! » Trois mois plus tard, quelqu'un a déjà écrit une fonction de neuf cents lignes qui s'appelle « gerer_tout ».

Flon ne disait pas que tous les langages se valent. Il disait que la qualité dépend de celui qui écrit, pas de l'outil. Un bon langage aide, mais il ne vous empêche pas de construire un labyrinthe.

J'ai suivi le cours d'un gourou des langages modernes. Il nous a montré un code parfait, élégant, sûr. Puis il nous a laissés seuls dix minutes. À son retour, nous avions réinventé les variables globales, le copier-coller et un dossier nommé « definitif_vrai ». Le langage était innocent. Nous, non.

PRINCIPE DE HOARE



En 50 lignes, le bug est une mouche sur la nappe. En 50 000, il a maison et potager.

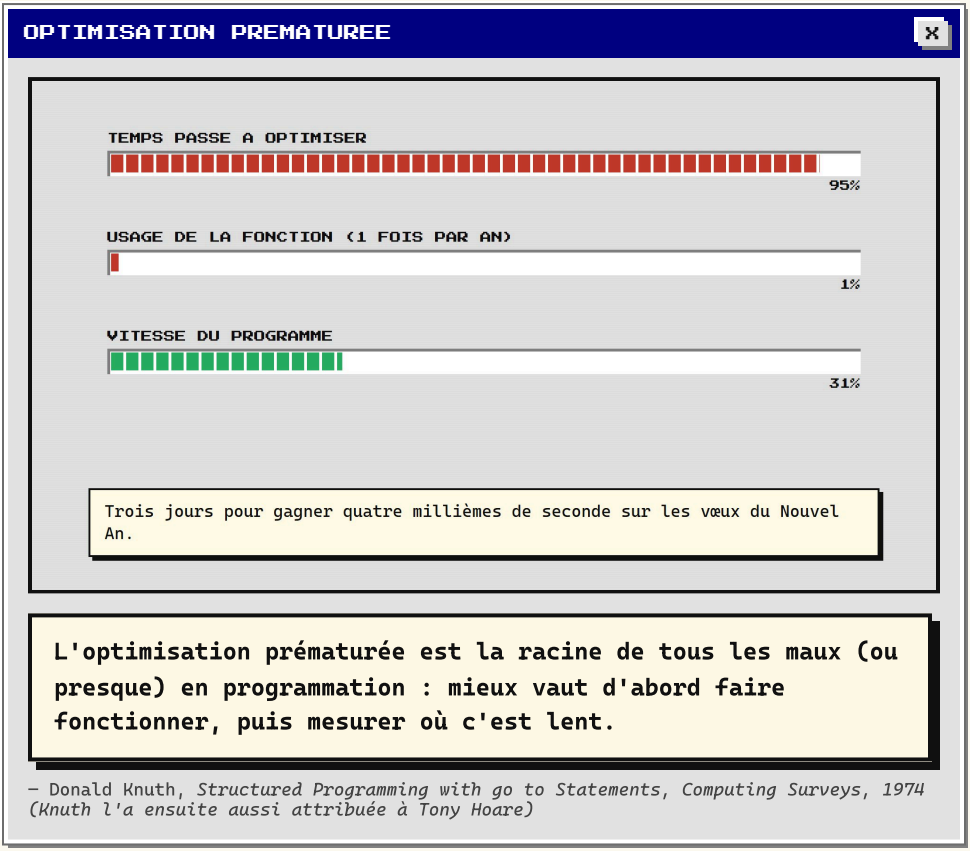
Il y a deux façons de concevoir un logiciel : le rendre si simple qu'il n'a évidemment pas de défauts, ou si compliqué qu'il n'a pas de défauts évidents. La première est bien plus difficile.

— C. A. R. Hoare, *The Emperor's Old Clothes, discours pour le prix Turing 1980, publié dans Communications of the ACM, 1981*

La complexité est une excellente cachette. Dans un programme de cinquante lignes, un bug se voit comme une mouche sur une nappe blanche. Dans un programme de cinquante mille, il a une maison, une famille et un potager.

Hoare nous rappelle que la simplicité n'est pas de la paresse : c'est la chose la plus pénible à obtenir. Il faut enlever, repenser, jeter du code auquel on s'était attaché. Ajouter est facile ; renoncer demande du courage.

J'ai demandé à un collègue si son module avait des défauts. Il m'a répondu : « Aucun qui se voie. » C'était vrai. On ne voyait rien. Pas même la logique. C'était un chef-d'œuvre de brouillard lombard sous forme de code, et dessous, dans le brouillard, les bugs jouaient aux boules.



— Donald Knuth, *Structured Programming with go to Statements*, *Computing Surveys*, 1974 *(Knuth l'a ensuite aussi attribuée à Tony Hoare)*

Il existe un type de programmeur qui passe trois jours à rendre ultra-rapide une fonction appelée une fois par an, le premier janvier, pour imprimer les vœux. Il gagne quatre millisecondes. Il fête ça comme un alunissage.

Le sens de la phrase relève du bon sens : on ne sait pas où se trouve la lenteur tant qu'on n'a pas mesuré. Optimiser à l'aveugle rend le code illisible et laisse intact le vrai goulet d'étranglement, qui est en général une requête lancée à l'intérieur d'une boucle.

Mon oncle a optimisé sa cave : étagères par ordre alphabétique, index, cache des bouteilles préférées. Puis il s'est aperçu que la porte était fermée et que la clé était dans la cave. Le temps d'accès était infini, mais bien organisé.

REGLES DE PIKE

PROFIL.TXT

```
Hypothèse : lent au calcul
Mesure réelle :
  calcul         2%
  lecture fich. 97%   (même fichier x 1000)
  autre          1%
```

Mesurer d'abord, optimiser ensuite. L'intuition des performances est un horoscope.

On ne peut pas savoir à l'avance où un programme perdra du temps : mesurez d'abord. Les algorithmes sophistiqués sont lents quand les données sont peu nombreuses, et ce sont les données, plus que le code, qui déterminent la bonne structure.

— Rob Pike, *Notes on Programming in C, 1989*

Les règles sont cinq, et sonnent comme les commandements d'un moine du terminal : mesurez, ne devinez pas ; n'utilisez pas d'algorithmes compliqués si n est petit, et n est en général petit ; la simplicité bat l'élégance ; ce sont les données qui commandent.

Le message, c'est que l'intuition d'un programmeur sur les performances est aussi fiable qu'un horoscope. On croit savoir où se trouve la lenteur, on passe deux jours à optimiser la mauvaise chose, et le vrai problème était une boucle qui lisait mille fois le même fichier.

J'ai implémenté un arbre équilibré auto-optimisant pour gérer l'annuaire du bureau. Il contient sept numéros. Le vieux papier collé sur le frigo les trouve plus vite. Le papier a même un élégant support magnétique.

LOI DE DEMETER

COMMANDE.JS

```
nom = client.commande.adresse
         .ville.departement.region.nom
// six maillons: si un change, tout tombe

mieux: nom = client.departement()
```

Une chaîne de commérages : chaque pièce en sait trop sur les autres.

Un objet ne devrait parler qu'à ses voisins directs et pas aux inconnus : pas de chaînes d'appels qui traversent la moitié du programme.

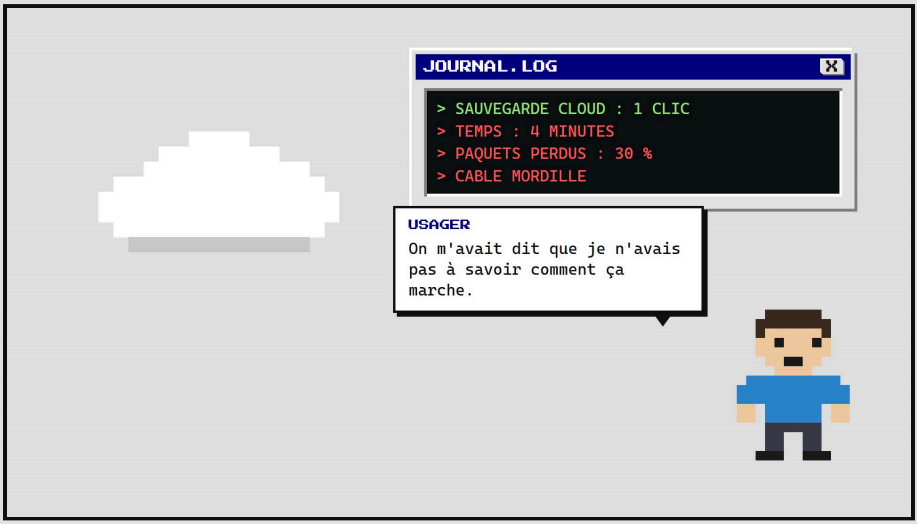
— Ian Holland, *Projet Demeter, Northeastern University, 1987 (groupe de Karl Lieberherr)*

On l'appelle aussi principe de connaissance minimale. En pratique : si vous voulez payer le marchand de journaux, vous lui donnez l'argent. Vous ne lui tendez pas votre portefeuille pour qu'il fouille dedans, trouve la carte, appelle la banque et demande le solde à votre mère.

Dans le code, une ligne comme client.commande.adresse.ville.departement.nom est un commérage en chaîne : chaque maillon en sait trop sur les autres, et si l'un change, tous s'écroulent. Réduire les connaissances rend les pièces remplaçables.

J'ai appliqué la loi au bureau. Je ne demande plus au collègue de demander au technicien de demander au fournisseur. Je demande au collègue, point. Il ne sait rien, mais le système est beaucoup plus découplé. Maintenant, plus rien ne marche, mais de façon élégante.

LOI DES ABSTRACTIONS QUI FUIENT



Toutes les abstractions non triviales finissent, tôt ou tard, par laisser fuir les détails qu'elles devaient cacher.

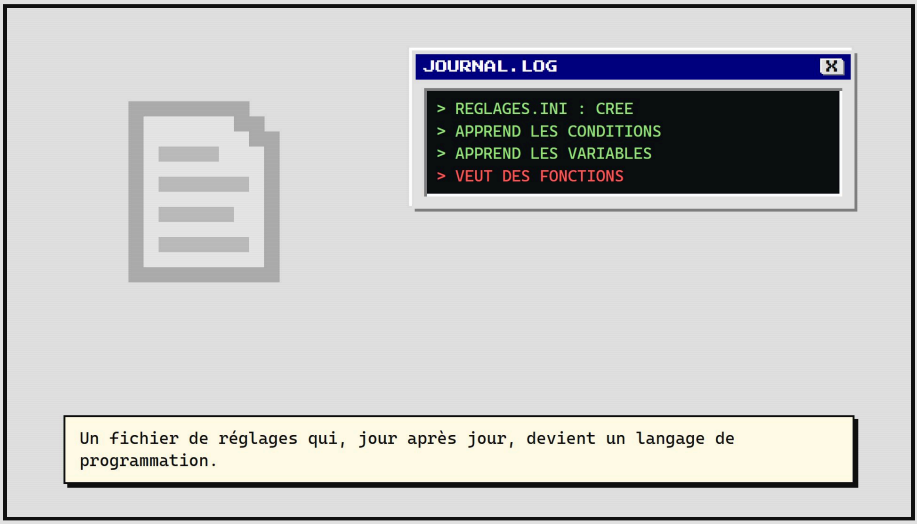
— Joel Spolsky, *The Law of Leaky Abstractions*, *Joel on Software*, 2002

Une abstraction est une belle façade : « Ne vous souciez pas du fonctionnement du réseau, enregistrez le fichier, c'est tout. » Ça marche très bien, jusqu'au jour où le fichier met quatre minutes et où vous devez découvrir ce qu'est une latence, un paquet perdu et un câble grignoté par un rongeur.

Le fait est que les outils pratiques font gagner du temps dans le travail ordinaire, mais ne vous dispensent pas de comprendre ce qu'il y a dessous. Quand ils fuient, et ils fuient toujours, il faut quelqu'un qui sache descendre à la cave.

Le cloud est l'abstraction suprême : « Vos données sont dans un nuage. » Puis le nuage tombe en panne dans un datacenter qui a un nom, une adresse et une climatisation cassée. Le nuage a un code postal. Je le savais. J'y suis allé.

DIXIEME REGLE DE GREENSPUN



Tout programme suffisamment compliqué, écrit en C ou en Fortran, contient une version improvisée, lente et pleine de bugs de la moitié du langage Lisp.

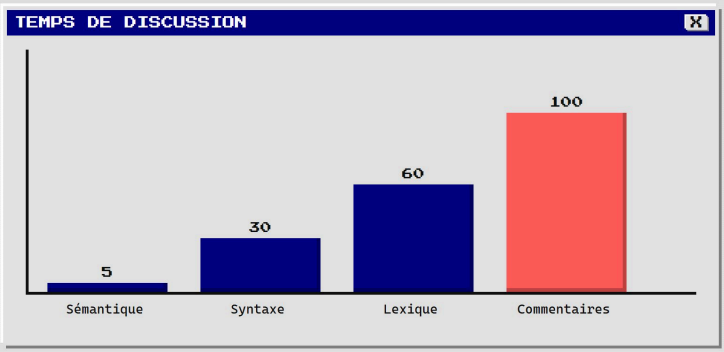
— Philip Greenspun, *Vers 1993 ; les neuf premières règles n'ont jamais existé*

Elle s'appelle « dixième », mais c'est la seule. Les neuf autres n'ont jamais été écrites. Rien que ça en fait la règle la plus honnête de l'informatique : elle promet neuf choses et n'en livre aucune, comme certains devis.

Le sens, c'est que les gros programmes finissent par réinventer, mal, des outils puissants qui existaient déjà : un mini-langage de configuration, un interpréteur fait maison, un système de règles écrit dans un fichier texte qui, à un moment donné, a même des boucles.

Dans mon vieux logiciel de gestion, il y avait un fichier de paramètres. Puis il a appris les conditions. Puis les variables. Un jour, il a demandé à pouvoir déclarer des fonctions. Nous l'avons éteint à temps. Maintenant, nous le gardons dans un tiroir, mais la nuit je l'entends compiler.

LOI DE WADLER



Le cœur du langage : cinq minutes. Le symbole des commentaires : trois semaines.

Dans la conception d'un langage, le temps de discussion augmente à mesure qu'on descend de la sémantique à la syntaxe, puis à la syntaxe lexicale, jusqu'aux commentaires, qui l'emportent sur tout.

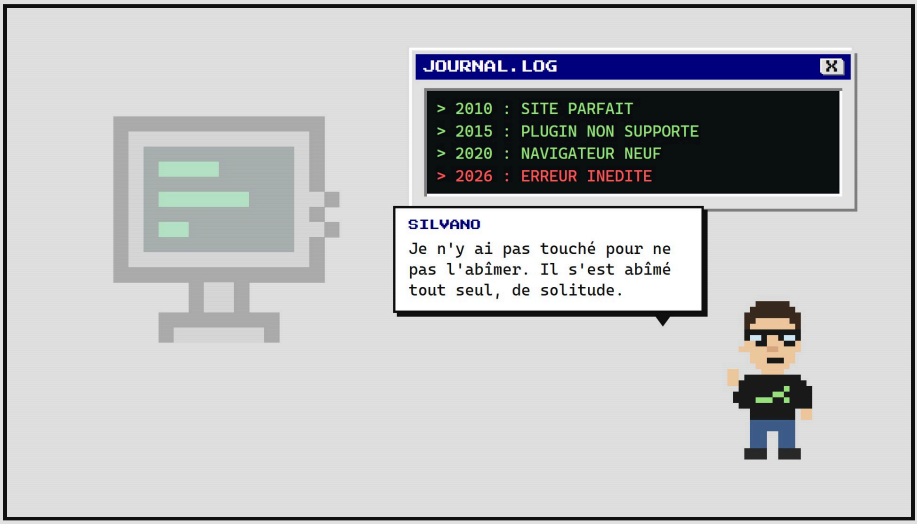
— Philip Wadler, *Messages sur la liste de diffusion de Haskell, 1992 et 1996*

Le sens profond d'un langage ? Cinq minutes, tout le monde d'accord. Le symbole des commentaires ? Trois semaines, deux schismes et un fork. Plus la question est petite, plus chacun se sent expert.

C'est la version programmeur d'une règle humaine universelle : sur les choses difficiles, on se tait ; sur les choses faciles, on se bat. Quiconque a participé à une revue de code a vu cinquante commentaires sur les espaces et zéro sur la logique.

Dans notre groupe, nous avons longuement débattu : tabulations ou espaces ? Nous avons créé un comité. Le comité a produit un document. Le document était indenté de façon mixte. Maintenant, le comité a un sous-comité. Le projet avance, dans les commentaires.

LOIS DE LEHMAN



Un logiciel utilisé dans le monde réel doit changer en permanence, sinon il devient de moins en moins utile ; et chaque changement augmente sa complexité, à moins que quelqu'un ne travaille exprès à la réduire.

— Meir M. Lehman, *Lois de l'évolution du logiciel, formulées avec László Bélády à partir des années 70*

Le logiciel est comme un jardin, pas comme une statue. Si on le laisse tranquille, il ne reste pas parfait : les mauvaises herbes poussent toutes seules. Le monde autour change, les lois changent, les navigateurs changent, et le programme impeccable de 2010 demande aujourd'hui d'installer un plugin disparu.

La partie la plus cruelle est la seconde : chaque modification ajoute du désordre. Une rustine ici, un cas particulier là, et le code devient une copropriété avec douze interphones différents.

J'ai un site Joomla que je ne touche plus depuis des années pour ne pas l'abîmer. Ce n'est pas moi qui l'ai abîmé. Il s'est abîmé tout seul, par pure solitude. Maintenant, je vais lui rendre visite le dimanche, je mets à jour une extension et il me remercie avec une nouvelle erreur.

LE CONSEIL DU SYSADMIN

Si vous écrivez du code, même un peu, même rien que pour vous, utilisez un **système de gestion de versions**. Le plus répandu est Git, et il est gratuit. Il fonctionne comme une machine à remonter le temps avec des étiquettes : chaque fois que le travail est dans un bon état, vous enregistrez une photo avec une phrase qui explique ce que vous avez changé.

Les avantages sont énormes. Vous revenez à hier en une commande. Vous voyez exactement quelle ligne a changé, et quand. Vous testez une idée folle sur une branche séparée et vous la jetez sans dégâts. Adieu les dossiers en double avec la date dans le nom et le mot « vieux » répété trois fois.

Trois règles d'or : enregistrez souvent et par petits pas ; écrivez des messages que comprendrait même un inconnu, c'est-à-dire vous dans six mois ; ne mettez jamais de mots de passe ni de clés d'accès dans le dépôt, même pas « juste une seconde ». Cette seconde, sur Internet, dure pour toujours.

CHAPITRE 7

CHAP 07 — REUNIONS

Projets, réunions et clients

Tôt ou tard, tout informaticien découvre une vérité gênante : les problèmes les plus difficiles ne sont pas dans l'ordinateur. Ils sont dans la salle de réunion, dans le devis, dans le mail avec douze personnes en copie. Le code, on peut le corriger. Une estimation lâchée dans un couloir, non.

Ce chapitre rassemble les lois du travail en équipe. On part des délais, toujours faux par excès d'optimisme, et on passe aux clients, qui savent ce qu'ils voulaient seulement après l'avoir vu. Puis viennent les réunions, qui se multiplient par récursion et discutent toujours du point le moins important, et les démos, qui échouent en proportion du grade de celui qui regarde.

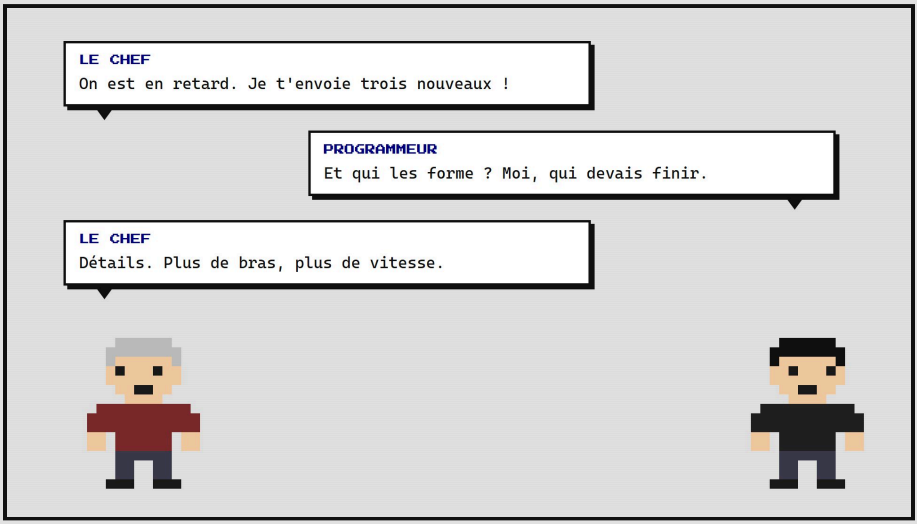
À la fin arrivent les lois les plus amères, celles sur les organisations : les chiffres qui deviennent des objectifs et cessent de dire la vérité, les solutions provisoires qui portent l'entreprise, les systèmes qui finissent par ressembler à l'organigramme, couloirs fermés compris.

Qui travaille dans un bureau reconnaîtra ses collègues, ses chefs et aussi lui-même. C'est normal. L'astuce, c'est de se reconnaître avant la réunion, pas pendant.

CONTENU.DIR

01 Loi de Brooks
02 Loi du multiplicateur pi
03 Loi de l'estimation assermentée
04 Loi du client voyant
05 Loi de l'exigence fantôme
06 Loi de la réunion récursive
07 Loi de futilité de Parkinson
08 Loi de Sayre
09 Lois de Wiio
10 Loi de la démo devant le client
11 Corollaire du public important
12 Loi du provisoire éternel
13 Loi de Conway
14 Loi de Gall
15 Loi de Weinberg
16 Loi de Goodhart
17 Loi de Campbell
18 Principe de Dilbert

LOI DE BROOKS



Ajouter des personnes à un projet logiciel en retard le retarde encore davantage.

— Fred Brooks, *The Mythical Man-Month*, 1975

Le manager visionnaire regarde le diagramme de Gantt, voit rouge et proclame : « Doublons l'équipe ! » C'est le même raisonnement selon lequel neuf femmes feraient un enfant en un mois.

Le problème, c'est que les nouveaux venus doivent être formés, et que ceux qui les forment sont justement ceux qui étaient en train de finir le travail. Ensuite, chaque personne en plus multiplie les conversations : à quatre, on se parle ; à douze, on fait des réunions pour décider quand faire les réunions.

Un jour, pour rattraper un retard, on m'a affecté trois stagiaires. Le premier jour, j'ai expliqué où sont les toilettes. Le deuxième, le mot de passe du Wi-Fi. Le troisième, le projet avait deux semaines de retard de plus, mais le Wi-Fi marchait à merveille.

LOI DU MULTIPLICATEUR PI



Pour obtenir la durée réelle d'un projet logiciel, prenez l'estimation du programmeur et multipliez-la par pi. Si le projet est important, prenez aussi les décimales.

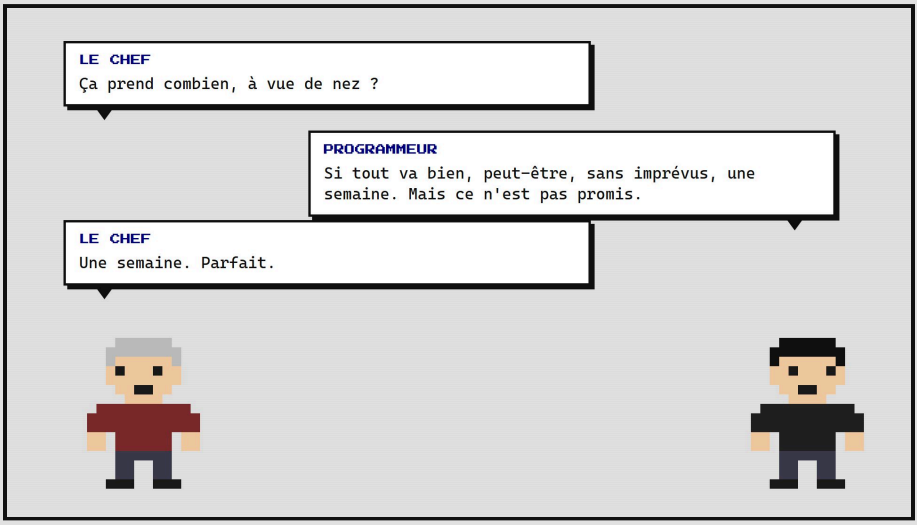
— Anonyme, *folklore informatique*

Le programmeur qui estime un travail est optimiste par constitution. Il imagine un monde sans imprévus, sans réunions, sans le client qui appelle pour demander si on peut avoir le logo un peu plus grand. Dans ce monde idéal, « deux jours », ce sont deux jours.

Puis arrive la réalité, qui a sa propre mathématique. Deux jours deviennent six virgule vingt-huit, et le zéro virgule vingt-huit, c'est le moment où l'on découvre que le serveur de test a le mot de passe de quelqu'un parti à la retraite.

Pourquoi justement pi ? Personne ne le sait. Mais c'est un nombre irrationnel et infini, donc le plus adapté pour décrire les délais. Dans mes devis, j'ai arrêté d'écrire les jours. J'écris directement le diamètre du projet, et je laisse le client faire le calcul.

LOI DE L'ESTIMATION ASSERMENTEE



Toute estimation donnée de vive voix, avec mille précautions et trois « peut-être », deviendra une échéance inscrite dans l'agenda du chef.

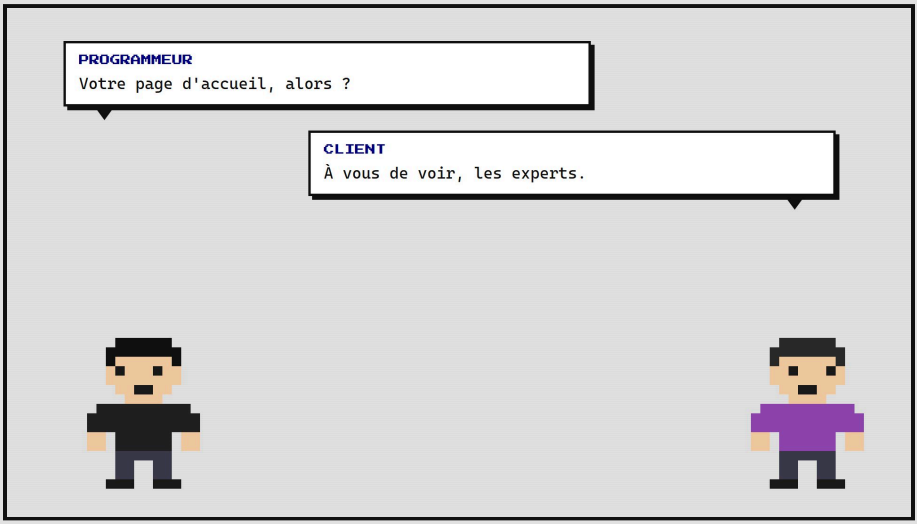
— Anonyme, *folklore informatique*

Le programmeur, interrogé dans le couloir, répond avec une prudence de diplomate : « Comme ça, à vue de nez, si tout va bien, peut-être, sans imprévus, une semaine. Mais ne le prenez pas pour argent comptant. » Le chef acquiesce. Il n'a entendu que « une semaine ».

Le lendemain, cette phrase apparaît dans un document officiel, puis dans un e-mail au client, puis dans un graphique coloré projeté en réunion. Les précautions ont été supprimées à la compression, comme les détails d'une photo envoyée par téléphone.

Le manager visionnaire, lui, la présente avec fierté : « L'équipe s'est engagée. » L'équipe ne savait pas qu'elle s'était engagée. Depuis, quand quelqu'un me demande combien de temps il faut, je réponds par une question : « C'est une question ou c'est un contrat ? » En général, le silence qui suit vaut réponse. Et devis.

LOI DU CLIENT VOYANT



Le client sait exactement ce qu'il veut au moment précis où vous lui livrez ce qu'il avait demandé.

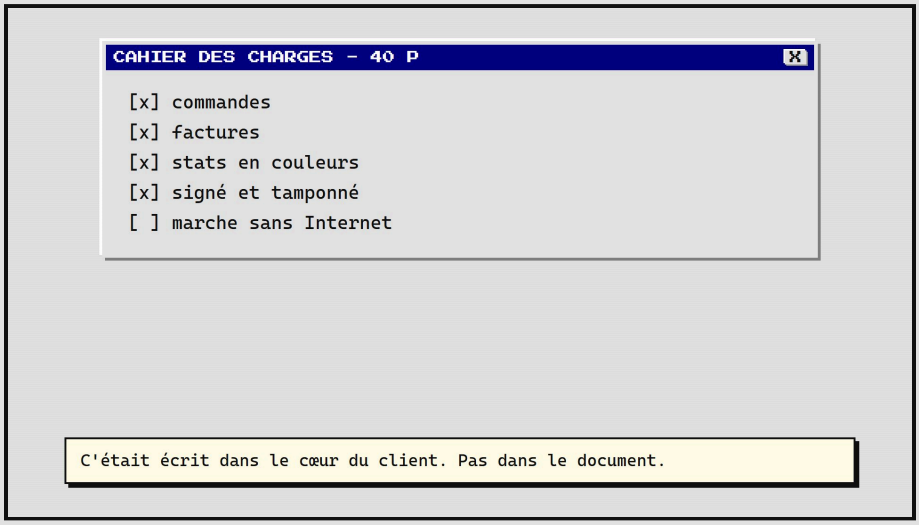
— Anonyme, *folklore informatique*

Pendant l'analyse, le client est vague, pensif, zen. « Faites comme vous voulez, c'est vous les experts. » Puis il voit le travail fini et a une révélation mystique, comme si les cieux s'ouvraient : maintenant, il sait. Il sait que ce n'était pas ça.

Le phénomène est bien connu de la littérature technique : un célèbre dessin anonyme montre une balançoire telle que la voulait le client, telle que l'a comprise le commercial, telle que l'a construite le programmeur. Aucune des versions n'est une balançoire.

J'ai trouvé une méthode infaillible. Je prépare tout de suite une version fausse, exprès, en une demi-journée. Le client la regarde, est horrifié et parle enfin clairement. C'est le seul cas en informatique où le bug est la fonctionnalité la plus utile du projet. Je la facture comme « analyse des besoins par contraste ».

LOI DE L'EXIGENCE FANTOME



L'exigence la plus importante du projet est communiquée oralement, en passant, le jour de la livraison.

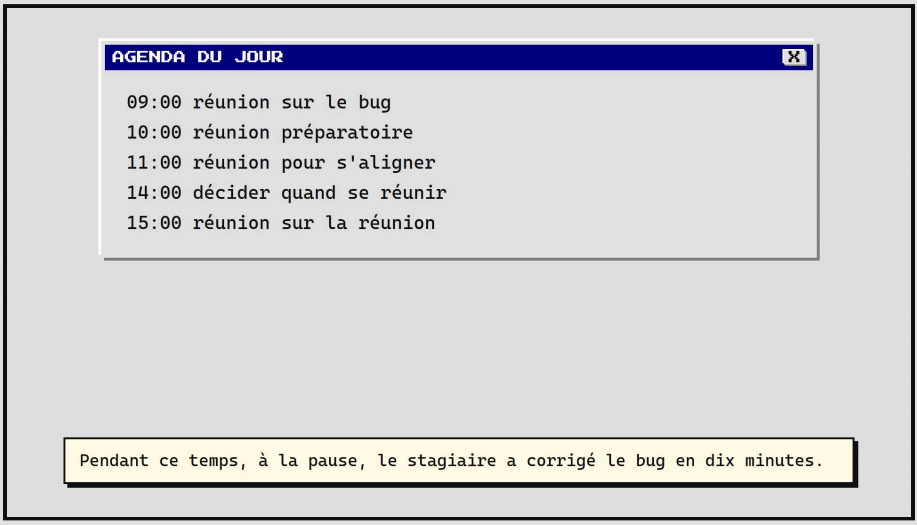
— Anonyme, *folklore informatique*

Le cahier des charges fait quarante pages, avec des tableaux, des graphiques et une table des matières digne d'un traité de paix. Tout a été discuté, signé, tamponné. Puis, le jour de la livraison, pendant qu'on se serre la main, le client ajoute : « Ah, évidemment, il faut que ça marche aussi sans Internet. »

Il ne le dit pas méchamment. Il le dit comme on dit « bonjour », avec la sérénité de celui qui croit rappeler un détail évident. Pour lui, ça a toujours été ainsi. C'était écrit dans son cœur, à défaut de l'être dans le document.

Depuis, j'ai instauré au bureau la réunion de la question finale. Avant de signer, je demande : « Y a-t-il quelque chose qui vous paraît tellement évident que vous ne l'avez pas dit ? » Le silence qui suit est toujours long. Et dans ce silence, à chaque fois, habite un nouveau module à développer.

LOI DE LA REUNION RECURSIVE



Tout problème technique complexe génère une réunion, qui génère une réunion préparatoire, qui génère le besoin d'une réunion pour décider quand faire la réunion.

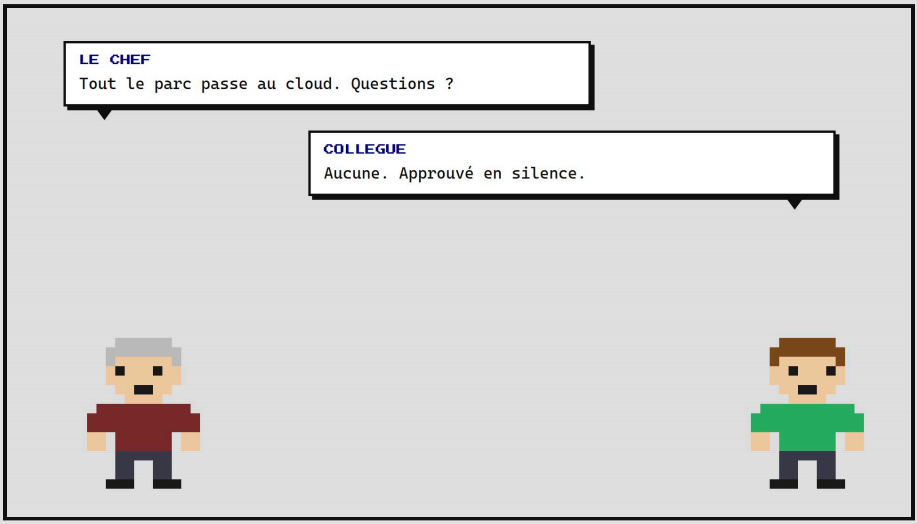
— Anonyme, *folklore informatique*

En informatique, la récursivité, c'est une fonction qui s'appelle elle-même. Dans les entreprises, c'est une réunion qui se convoque elle-même, à l'infini, jusqu'à ce que quelqu'un parte en vacances et brise le cycle par épuisement du calendrier.

Le modèle est toujours le même. Huit personnes se réunissent pour discuter d'un bug. On découvre qu'il manque des informations. On fixe une nouvelle rencontre avec quatre autres personnes. Lors de cette rencontre émerge la nécessité de « s'aligner ». Pendant ce temps, le bug a été corrigé par le stagiaire, en dix minutes, pendant la pause.

Il y a toujours le manager qui ouvre la séance avec solennité : « Messieurs, aujourd'hui, nous décidons. » Et en effet, on décide : la date de la prochaine réunion. J'ai proposé d'installer un compteur dans la salle, comme ceux des fêtes de village. Chaque heure de réunion, une ligne de code en moins.

LOI DE FUTILITE DE PARKINSON



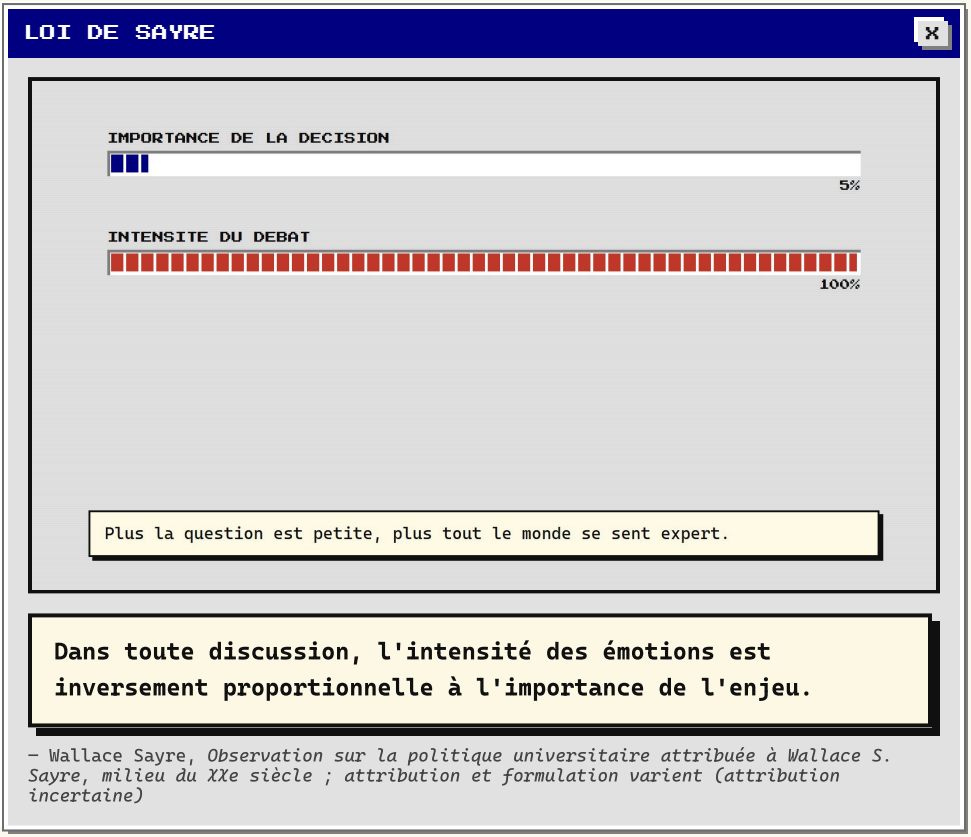
Une organisation consacre un temps inversement proportionnel à l'importance des décisions : on discute peu du réacteur et énormément de l'abri à vélos.

— C. Northcote Parkinson, *Parkinson's Law, 1957 ; popularisée en informatique sous le nom de « bikeshedding » par Poul-Henning Kamp dans la communauté FreeBSD, 1999*

Le conseil approuve en deux minutes le projet à plusieurs millions, que personne ne comprend. Puis il passe une heure sur l'abri à vélos, que tout le monde comprend parfaitement et sur la couleur duquel chacun a un avis.

En informatique, c'est devenu un mot d'ordre : quand une discussion s'enflamme sur un détail sans importance, quelqu'un écrit « bikeshedding » et tout le monde, un instant, a honte. Puis on reprend la discussion sur le nom de la variable.

Lors de ma dernière réunion, nous avons approuvé en silence le passage de tous les serveurs au cloud. Puis nous nous sommes disputés quarante minutes sur l'icône du nouveau logiciel de gestion : un engrenage ou un nuage ? Au final, nous avons choisi un engrenage en forme de nuage. Personne ne sait ce qu'il fait. Comme le cloud, d'ailleurs.



Dans toute discussion, l'intensité des émotions est inversement proportionnelle à l'importance de l'enjeu.

— Wallace Sayre, *Observation sur la politique universitaire attribuée à Wallace S. Sayre, milieu du XXe siècle ; attribution et formulation varient (attribution incertaine)*

Née pour décrire les querelles entre professeurs d'université, elle s'applique parfaitement aux réunions techniques. La migration de toute la base de données passe en cinq minutes. La couleur du bouton « Annuler » provoque deux heures de débat et un e-mail avec la moitié de l'entreprise en copie.

La raison est simple : sur les questions énormes, peu de gens se sentent compétents ; sur les minuscules, tout le monde a un avis, et le défendre ne coûte rien.

La réunion la plus longue de ma carrière portait sur le nom d'un dossier partagé. « Documents », « Documentation » ou « Doc ». Quatre heures. Nous avons voté. « Documents_DEF » a gagné. Le lendemain est apparu « Documents_DEF_nouveau ». Le conflit est devenu permanent, comme certaines guerres froides, mais avec plus de droits d'écriture.

LOIS DE WIIO

UN SITE SIMPLE, SELON :

- le client : trois pages
- le graphiste : minimalisme nordique
- le développeur : pas de BDD
- le commercial : ça coûte peu

À la livraison, personne n'est content. Tous sûrs d'avoir demandé exactement ça.

La communication échoue en général, sauf par hasard. Et si elle semble avoir réussi comme prévu, il y a forcément un malentendu.

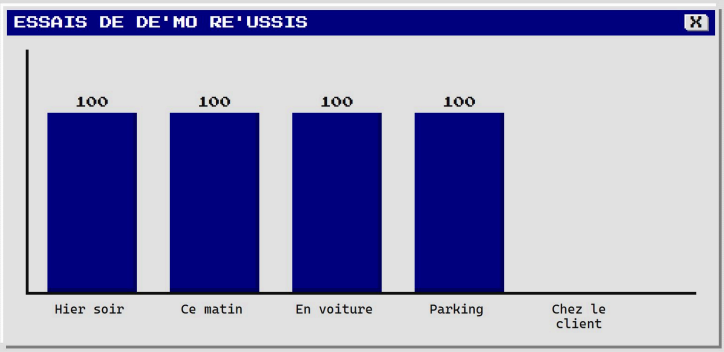
— Osmo Antero Wiio, *« Wiion lait – ja vähän muidenkin »*, 1978

Un professeur finlandais, spécialiste de la communication, arriva à une conclusion que tout admin sys connaît déjà : communiquer ne marche pas. Quand ça marche, c'est un accident. Et s'il vous semble que tout le monde a compris, c'est seulement parce que chacun a compris une chose différente.

Le client demande « un site simple ». Lui pense trois pages. Le graphiste pense minimalisme nordique. Le programmeur pense pas de base de données. Le commercial pense « pas cher ». À la livraison, personne n'est satisfait, mais tout le monde est sûr d'avoir demandé exactement cette chose-là.

C'est pour cela qu'existent les spécifications écrites, les comptes rendus, les e-mails récapitulatifs. Ils servent à documenter avec précision le malentendu. Désormais, je termine chaque réunion en demandant à tout le monde de répéter ce que nous avons décidé. À chaque fois, il en sort cinq projets. Au moins, on le découvre tout de suite, et pas à la recette.

LOI DE LA DEMO DEVANT LE CLIENT



Taux de réussite. La dernière mesure a été prise en présence d'un décideur.

Un programme qui a marché lors de cent essais plante au cent unième : celui devant le client.

— Anonyme, *folklore informatique (aussi connu sous le nom d'« effet démo »)*

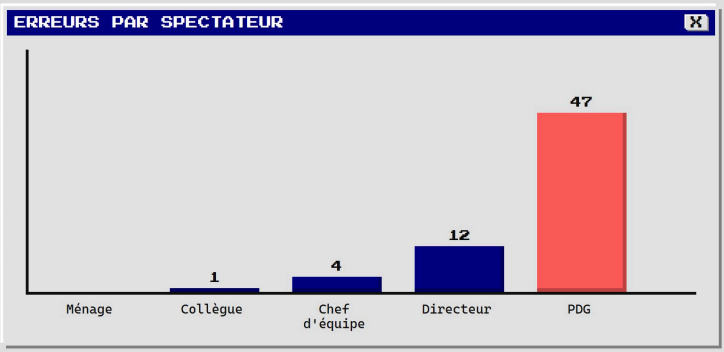
La démo a été répétée la veille au soir, le matin, dans la voiture et sur le parking. Tout roule. Puis vous entrez, le client s'assoit, vous dites avec assurance « Je vais vous montrer » et l'écran devient une palette d'erreurs jamais vues.

Les informaticiens appellent ça l'effet démo. C'est le seul phénomène connu capable de détecter la présence d'un décideur à dix mètres de distance, à travers les murs.

Le plus beau, c'est que dès que le client s'en va, le programme repart parfaitement. Il vous regarde comme pour dire : « Qu'est-ce qui s'est passé ? Moi, j'étais là. »

Conseil de quelqu'un qui l'a vécu : préparez toujours un plan B. Une vidéo enregistrée de la démo qui fonctionne, ou des captures d'écran des étapes clés. Si le programme fait des caprices, on passe le film.

COROLLAIRE DU PUBLIC IMPORTANT



Une échelle hiérarchique, précise comme un organigramme. Avec le ménage, le serveur se calme.

Plus le rang de ceux qui assistent à la démonstration est élevé, plus la démonstration a de chances d'échouer.

— Anonyme, *folklore informatique (recueils Usenet de lois sur les ordinateurs)*

Devant un collègue, la démo tient. Devant le chef de service, elle hésite. Devant le directeur, elle tousse. Devant le PDG tout juste arrivé en hélicoptère, l'ordinateur s'éteint et demande à installer quarante-sept mises à jour.

C'est une échelle hiérarchique de l'erreur, précise comme un organigramme. Certains chercheurs soutiennent que les costumes coûteux émettent un champ électromagnétique. Je n'ai pas de preuves. Mais j'ai remarqué que le serveur se calme quand entre l'agent d'entretien.

Le manager visionnaire, pendant ce temps, sourit et dit : « Parfait, nous avons identifié une opportunité d'amélioration. » Ce qui, en langage d'entreprise, signifie : on se voit dans mon bureau.

Vrai conseil : pour les présentations importantes, utilisez un environnement déjà préparé et stable, et faites une répétition complète dans la même salle, avec le même ordinateur.

LOI DU PROVISOIRE ETERNEL

CATALOGUE.PHP

```
// TEMPORAIRE, juste pour le week-end
// à corriger plus tard
si prix vide: prix = 9,99

dernière modif: il y a 12 ans
lignes qui en dépendent: toutes
```

Le « plus tard » n'est jamais venu. La rustine tient tout le catalogue.

Dans le logiciel, rien n'est plus définitif qu'une solution temporaire.

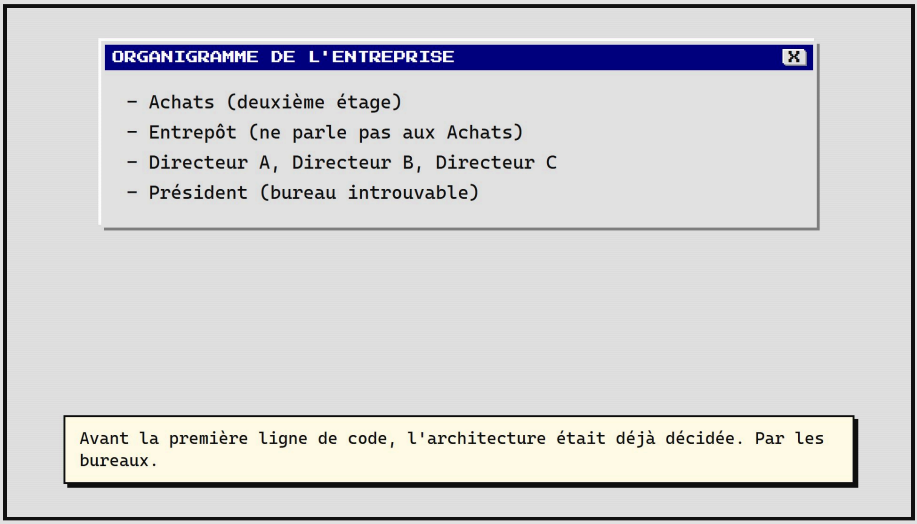
— Anonyme, *folklore informatique*

Chaque programme contient une ligne accompagnée de la note « à arranger plus tard ». Plus tard n'arrive jamais. La ligne survit à trois versions, deux changements de serveur et un changement de propriétaire de l'entreprise.

Les solutions provisoires ont une trempe extraordinaire. Les solutions définitives, conçues avec soin, sont réécrites tous les deux ans. La rustine posée à onze heures du soir pour relancer les factures, en revanche, devient patrimoine historique, avec sa dignité et son aura.

Je gère un site où, il y a douze ans, quelqu'un a écrit un petit script « juste pour le week-end ». Aujourd'hui, il porte tout le catalogue. Personne n'ose y toucher. De temps en temps je le regarde, comme ça, avec respect, comme on regarde le mur porteur d'une vieille ferme : de travers, oui, mais si on l'enlève, tout s'écroule.

LOI DE CONWAY



Une organisation qui conçoit un système finira par produire un système qui copie sa propre structure de communication.

— Melvin Conway, *How Do Committees Invent?*, *Datamation*, 1968

Si quatre groupes travaillent sur un compilateur, il en sortira un compilateur en quatre passes. Si le service achats ne parle pas avec l'entrepôt, le logiciel de gestion aura deux fichiers produits qui ne se disent pas bonjour.

Le logiciel est un miroir. On regarde l'architecture et on voit l'organigramme, avec toutes les rancœurs qui vont avec. Ce module qui ne communique avec personne ? C'est le collègue du troisième étage qui déjeune seul.

J'ai vu un site avec trois menus différents, un par directeur. Un quatrième menu, caché, menait à la page du président. Personne ne savait comment y arriver, exactement comme pour son bureau. L'architecture était parfaite : elle reflétait l'entreprise jusqu'au dernier couloir fermé à clé.

LOI DE GALL



Un système complexe qui fonctionne provient toujours d'un système simple qui fonctionnait. Un système complexe conçu à partir de zéro ne fonctionne pas et ne peut pas être réparé.

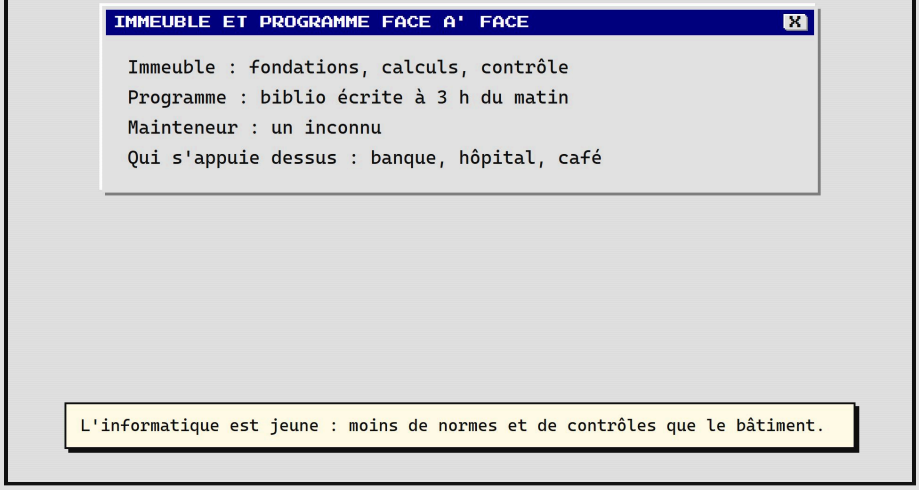
— John Gall, *Systemantics (General Systemantics)*, 1975

Le gourou de la transformation numérique monte sur scène : « Nous allons tout refaire à partir de zéro, microservices, intelligence artificielle, blockchain, en six mois ! » Applaudissements. Seize mois plus tard, on revient au fichier Excel partagé, qui marchait depuis 2004.

La loi est un éloge de l'évolution. Les systèmes qui tiennent ont grandi petit à petit, comme les stalactites ou les dossiers « Nouveau dossier (7) ». Ceux qui sont nés énormes et parfaits sur le papier s'effondrent, parce que personne ne les a jamais vus fonctionner en petit.

Mon logiciel de gestion préféré est né sous la forme d'un fichier texte de trois lignes. Aujourd'hui, il gère une entreprise entière. Il a encore ces trois lignes à l'intérieur, quelque part. Personne n'y touche. C'est la relique porteuse.

LOI DE WEINBERG



Si les bâtisseurs construisaient les bâtiments comme les programmeurs écrivent les programmes, le premier pic-vert de passage détruirait la civilisation.

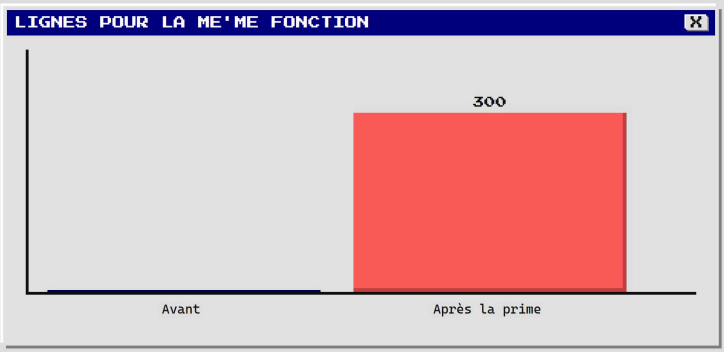
— Gerald Weinberg, *Attribution incertaine : apparaît dans un recueil de 1975 (CoEvolution Quarterly) et comme « Deuxième loi de Weinberg » dans Murphy's Law d'Arthur Bloch, 1977*

Un vrai immeuble a des fondations, des calculs de structure, des contrôles. Un programme moyen a une bibliothèque téléchargée à trois heures du matin, maintenue par un inconnu, dont dépendent une banque, un hôpital et le distributeur de café.

La loi n'est pas qu'une moquerie : elle rappelle que l'informatique est une discipline jeune, avec moins de normes et moins de contrôles que le bâtiment. Une grande partie de ce que nous utilisons tient par bonne volonté et par chance.

Un jour, j'ai vu un pic-vert sur le toit de notre entrepôt, juste au-dessus de la salle des serveurs. Je me suis placé entre lui et l'antenne. Nous nous sommes regardés longtemps. Finalement, il est parti. Je crois qu'il a compris que ce n'était pas nécessaire : le serveur tomberait en panne tout seul, le lundi.

LOI DE GOODHART



On récompense les lignes de code. La productivité triple. Le logiciel, mystérieusement, non.

Quand une mesure devient un objectif, elle cesse d'être une bonne mesure.

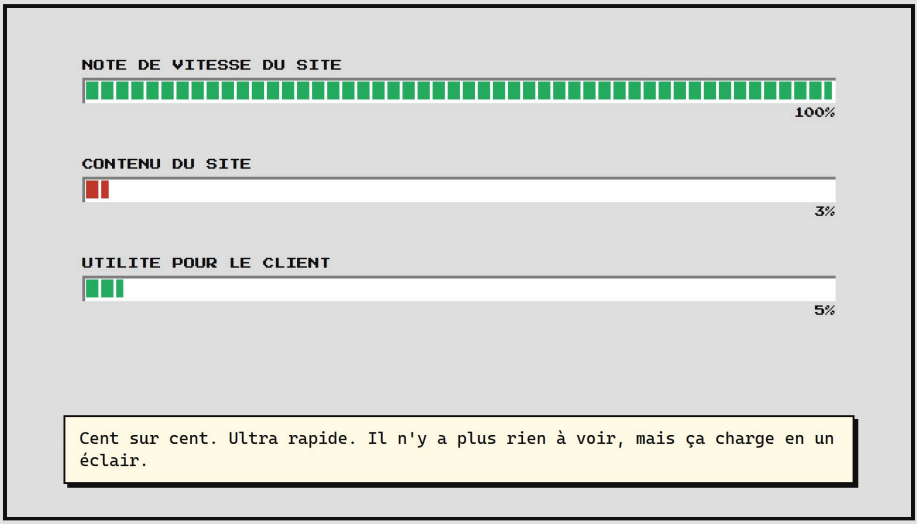
— Charles Goodhart, *Article sur la politique monétaire, 1975 ; la formulation célèbre est de Marilyn Strathern, 1997*

L'entreprise décide de mesurer la productivité des programmeurs en comptant les lignes de code. Le mois suivant, les programmes sont interminables. Une fonction qui tenait en trois lignes en occupe maintenant trois cents, avec des commentaires poétiques et des variables aux noms de quatre mots. La productivité a triplé. Le logiciel, mystérieusement, non.

Voilà la loi de Goodhart : dès qu'on récompense un chiffre, les gens apprennent à gonfler le chiffre au lieu d'améliorer la chose qu'il devait représenter. Ça vaut pour les tickets clôturés, pour les étoiles des avis, pour les scores de vitesse des sites.

Au service d'assistance, on mesurait le temps de clôture des appels. Un collègue les clôturait tous en huit secondes. Il répondait, disait « essayez de redémarrer » et raccrochait. Élu trois fois employé du mois. Les utilisateurs l'appelaient l'Éclair.

LOI DE CAMPBELL



Plus un indicateur chiffré est utilisé pour prendre des décisions importantes, plus il est corrompu, et plus il déforme les processus qu'il devrait surveiller.

— Donald T. Campbell, *« Assessing the Impact of Planned Social Change », 1976 (formulations antérieures dès 1969)*

Cousine germaine de la loi de Goodhart, la loi de Campbell ajoute un détail sombre : non seulement le chiffre cesse de dire la vérité, mais il commence à déformer le travail lui-même. Si la prime dépend du score, le travail devient de produire du score.

Dans le monde des sites, c'est flagrant. On décide que seule compte la note de l'outil de mesure de vitesse. Alors on enlève les images, les textes, les fonctions, tout ce qui pèse. Le site obtient cent sur cent. Il est ultra-rapide. Il n'y a plus rien à voir, mais ça se charge en un éclair.

Dans une entreprise que je connais, on évaluait l'informaticien sur le nombre d'ordinateurs en état de marche à la fin du mois. Lui, très sagement, le dernier jour du mois, n'en touchait aucun. Ils marchaient tous. Très lents, pleins d'alertes, mais statistiquement en parfaite santé.

PRINCIPE DE DILBERT



JOURNAL.LOG

```
> ACCES AU SERVEUR : REFUSE
> NOUVEAU BUREAU : AVEC VUE
> ROLE : MANAGER
```

Contrôle d'accès primitif : au lieu des droits, un bureau avec vue.

Les entreprises ont tendance à promouvoir les employés les moins compétents à des postes de direction, pour limiter les dégâts qu'ils peuvent faire dans le vrai travail.

— Scott Adams, *Wall Street Journal, 1995 ; livre « The Dilbert Principle », 1996*

Le principe de Peter disait qu'on monte jusqu'à devenir incompétent. Le principe de Dilbert est plus cynique : certains partent déjà incompétents et sont déplacés vers le haut exprès, là où ils ne peuvent rien toucher d'important. Comme nommer l'enfant hyperactif capitaine de l'équipe, pour qu'il ne touche pas au ballon.

En informatique, le phénomène a un nom technique officieux : le manager visionnaire. Il ne sait pas ce qu'est une base de données, mais il a compris que l'avenir est dans les données. Il le dit en réunion en tapant du poing sur la table. Puis il demande à quelqu'un de lui imprimer l'e-mail.

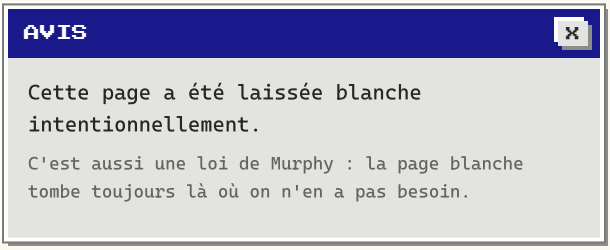
Avouons-le, le système a son efficacité : la personne est tenue loin du serveur, le serveur dit merci. C'est une forme primitive de contrôle d'accès. Sauf qu'au lieu des permissions, on utilise un bureau avec vue.

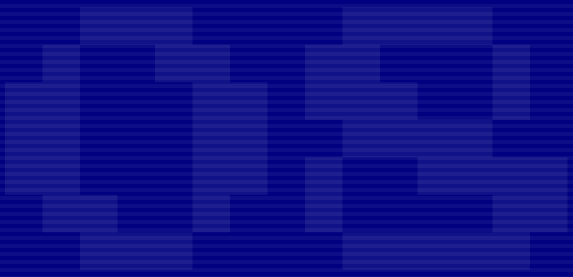
LE CONSEIL DU SYSADMIN

Deux habitudes qui m'ont sauvé plus de projets que n'importe quel logiciel.

Le mail récapitulatif. Après chaque réunion ou appel important, écrivez dans l'heure un mail court à tous les présents : ce qu'on a décidé, qui fait quoi, pour quand, ce qui reste ouvert. Cinq lignes. Si quelqu'un a compris autre chose, vous le découvrez tout de suite et pas à la livraison. Et dans trois mois, quand plus personne ne se souviendra de rien, ce mail sera la version officielle des faits.

L'estimation en fourchette. Ne donnez jamais un seul chiffre. Dites « de trois à six jours, parce que ça dépend de ceci et de cela ». Le chiffre unique devient une promesse ; la fourchette oblige tout le monde à parler de ce qui l'élargit. Et mettez aussi par écrit ce qui est exclu du travail. L'exigence fantôme déteste les listes à puces.


AVIS
Cette page a été laissée blanche intentionnellement.
C'est aussi une loi de Murphy : la page blanche tombe toujours là où on n'en a pas besoin.



CHAPITRE 8

CHAP 08 — AVENIR

Progrès, mises à jour et intelligence artificielle

Le dernier chapitre regarde vers l'avant, ce qui en informatique veut dire fixer la barre de progression d'une mise à jour. On commence justement par là : des mises à jour qui arrivent au mauvais moment, qui réparent une chose et en cassent une autre, des ordinateurs neufs déjà vieux à la sortie du magasin.

Puis on change d'échelle. Voici les lois qui décrivent la grande course de la technologie : des puces toujours plus puissantes, des usines toujours plus chères, des logiciels toujours plus lourds, des programmes qui veulent tout faire. Ce sont des lois économiques et techniques, parfois optimistes, souvent amusées. Elles expliquent pourquoi l'ordinateur est infiniment plus rapide qu'il y a vingt ans et le traitement de texte s'ouvre dans le même temps.

Le chapitre se termine par six corollaires consacrés à l'intelligence artificielle. Ceux-là, je les ai écrits moi-même, après de longues heures de travail avec ces assistants, et je les signe avec la fierté de celui qui a vu une machine s'excuser plusieurs fois pour exactement la même erreur. Je ne suis pas contre l'IA : je l'utilise tous les jours. C'est justement pour ça que je l'aime comme on aime un collègue brillant qui, de temps en temps, invente des noms de fonctions.

CONTENU.DIR

01 Loi de la mise à jour inopportune
02 Loi de la mise à jour qui casse
03 Loi de l'ordinateur neuf
04 Loi de Wirth
05 Loi de Zawinski
06 Loi d'Atwood
07 Loi de Rock
08 Loi de Grosch
09 Loi d'Amdahl
10 Loi de Gustafson
11 Loi de Koomey
12 Loi de Bell sur les classes d'ordinateurs
13 Paradoxe de Jevons
14 Loi d'Amara
15 Corollaire de la fonction imaginaire
16 Corollaire des excuses artificielles
17 Corollaire des trois secondes
18 Corollaire du prompt solennel
19 Corollaire du reste inchangé
20 Corollaire de la règle oubliée

LOI DE LA MISE A JOUR INOPPORTUNE



L'ordinateur décide d'installer les mises à jour toujours quand vous devez l'éteindre et filer, ou l'allumer et présenter.

— Anonyme, *folklore informatique*

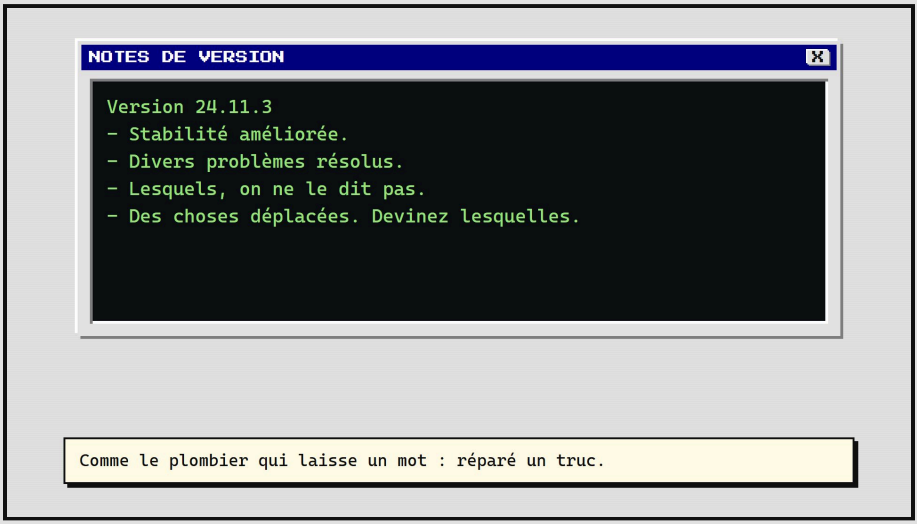
L'ordinateur a passé la nuit allumé, inactif, à s'ennuyer. Il aurait pu se mettre à jour en paix. Mais non : il attend que vous cliquiez sur « Arrêter » avec votre manteau sur le dos et les clés à la main. Alors il annonce, aimablement : « Mise à jour 1 sur 23. N'éteignez pas l'ordinateur. »

Et vous restez là, comme devant un passage à niveau. Le train des mises à jour passe lentement, wagon après wagon. Parfois, il s'arrête à trente pour cent pour méditer sur le sens de la vie.

Un de mes voisins a acheté un portable neuf pour faire vite sa déclaration de revenus. Il l'a allumé en mars. En mai, il a fait sa déclaration à la main, sur papier, avec le stylo-plume de son grand-père.

Conseil de technicien : configurez les heures d'activité du système, pour que les mises à jour s'installent quand vous n'utilisez pas l'ordinateur. Et ne l'éteignez jamais de force pendant une mise à jour.

LOI DE LA MISE A JOUR QUI CASSE



Toute mise à jour qui résout un problème en introduit au moins un nouveau, en général dans quelque chose qui marchait très bien avant.

— Anonyme, *folklore informatique*

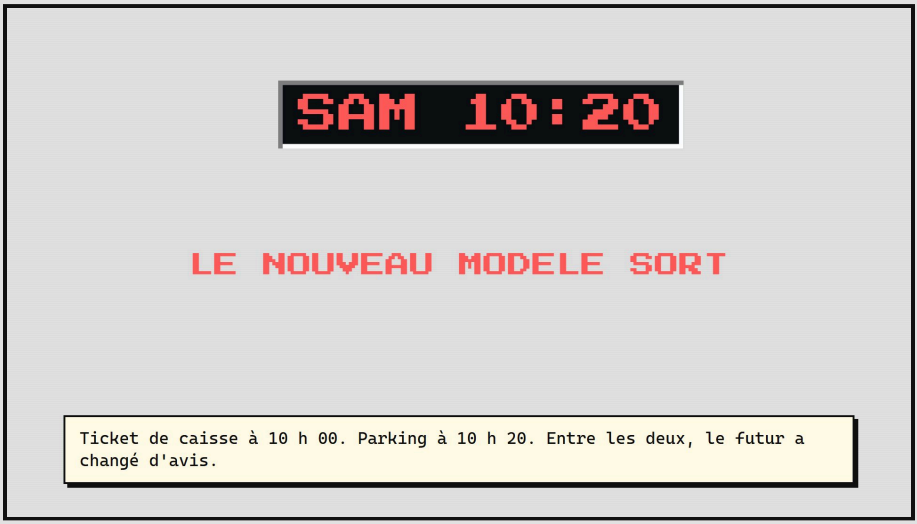
La mise à jour se présente avec une liste de nouveautés enthousiasmantes : « Stabilité améliorée. Correction de certains problèmes. » Lesquels, on ne le dit jamais. C'est un peu comme un plombier qui laisse un mot : « Réparé un truc. »

Le lendemain, l'imprimante n'imprime plus, le logiciel de comptabilité réclame une licence et l'icône de la corbeille est devenue verte. On appelle ça le progrès. Moi, j'appelle ça un déménagement : tout est encore là, mais plus rien n'est à sa place.

Le gourou du « on a toujours fait comme ça » profite de l'occasion et proclame : « Je vous l'avais dit, il ne faut jamais rien mettre à jour. » Puis il attrape un virus par un logiciel resté bloqué en 2012.

Vrai conseil : les mises à jour de sécurité doivent être faites. Mais avant les grosses mises à jour, faites une sauvegarde et, si vous le pouvez, attendez quelques jours que d'autres trouvent les problèmes.

LOI DE L'ORDINATEUR NEUF



L'ordinateur que vous venez d'acheter est déjà dépassé par le modèle sorti pendant que vous rentriez chez vous, qui coûte en plus moins cher.

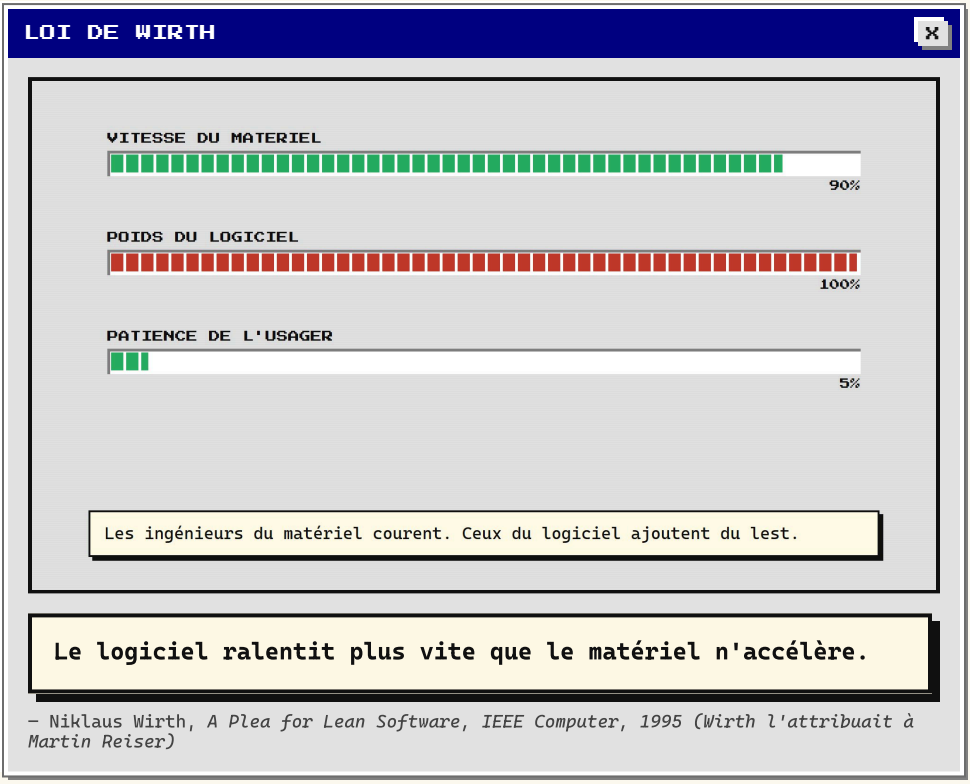
— Anonyme, *folklore informatique*

Le moment le plus heureux de la vie d'un ordinateur, c'est le ticket de caisse. Le deuxième moment le plus heureux n'arrive jamais, parce que sur le trajet entre le magasin et la voiture sort le nouveau modèle : plus rapide, plus fin, et en offre de lancement.

J'ai une connaissance qui, pour vaincre la loi, attend toujours le modèle suivant. Il l'attend depuis 2011. Il utilise encore l'ordinateur portable de l'époque, qui démarre en douze minutes. Il dit qu'il n'est pas pressé : lui, il « investit dans l'avenir ».

Frères et sœurs du progrès, voici la vérité : aucun ordinateur n'est jamais à jour. Il n'existe que des ordinateurs qui ne savent pas encore qu'ils sont vieux.

Vrai conseil : achetez quand vous en avez besoin, en dimensionnant pour trois à cinq ans d'utilisation. Le modèle parfait sortira toujours demain.



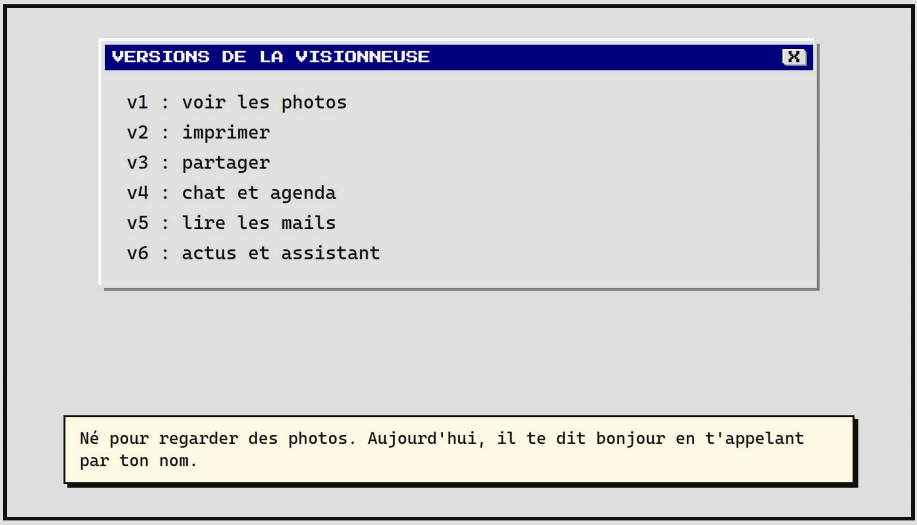
— Niklaus Wirth, *A Plea for Lean Software*, *IEEE Computer*, 1995 (*Wirth l'attribuait à Martin Reiser*)

En 1995, un traitement de texte tenait sur quelques disquettes et démarrait en un instant. Aujourd'hui, nous avons des processeurs des milliers de fois plus rapides, et l'éditeur de texte met le même temps, parce qu'entre-temps il a appris à charger un navigateur entier, trois moteurs d'analyse et un assistant qui vous demande si vous voulez de l'aide.

C'est une course : les ingénieurs du matériel courent, ceux du logiciel ajoutent du lest. La ligne d'arrivée recule toujours de quelques secondes de chargement.

Mon nouveau PC a 32 gigas de RAM. Je les ai donnés en pâture à la calculatrice. Maintenant elle calcule 2+2 à toute vitesse, après s'être mise à jour, avoir vérifié la licence et m'avoir proposé un abonnement premium pour les nombres décimaux.

LOI DE ZAWINSKI



Tout programme tend à s'étendre jusqu'à pouvoir lire le courrier. Ceux qui n'y parviennent pas sont remplacés par ceux qui y parviennent.

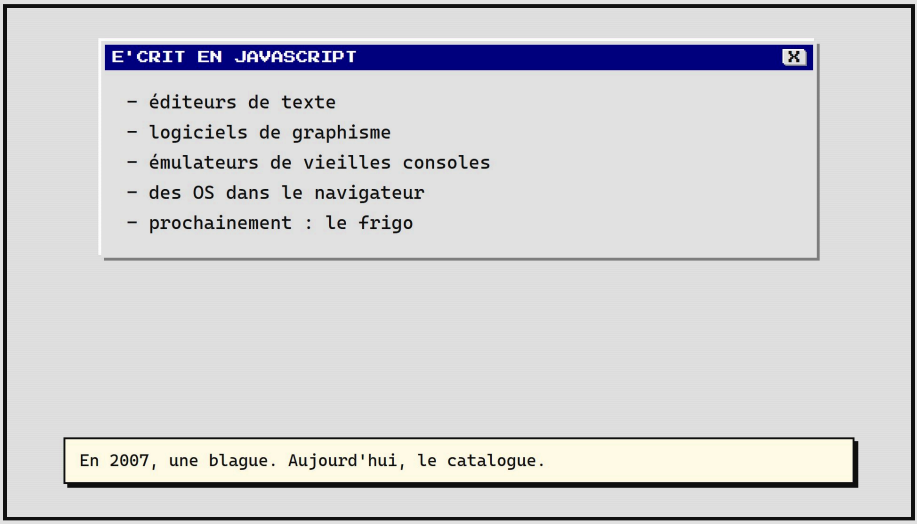
— Jamie Zawinski, *Observation attribuée à Jamie Zawinski, développeur de Netscape, milieu des années 90*

Il naît visionneuse d'images. Puis il ajoute l'impression, puis le partage, puis le chat, puis un agenda. Deux ans plus tard, il a la messagerie, un fil d'actualités et un assistant virtuel qui vous salue par votre prénom.

Le sens est sérieux : les programmes ont tendance à accumuler des fonctions parce que chaque utilisateur en réclame une de plus, et celui qui s'arrête a l'air dépassé. Aujourd'hui, le courrier a été remplacé par le chat, la vidéo et l'intelligence artificielle, mais la fièvre est la même.

Hier, j'ai mis à jour ma calculatrice. Maintenant, elle a une boîte mail. J'ai reçu une newsletter de la calculatrice qui m'informe des nouvelles fonctions de la calculatrice. Je n'ai toujours pas trouvé où on fait les additions.

LOI D'ATWOOD



> **Toute application qui peut être écrite en JavaScript finira, tôt ou tard, par être écrite en JavaScript.**

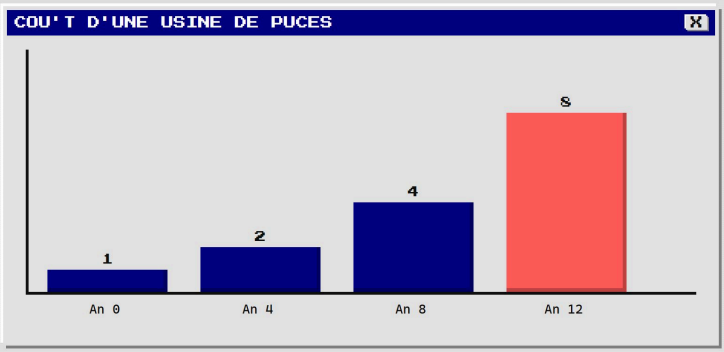
— Jeff Atwood, *The Principle of Least Power, blog Coding Horror, 2007*

En 2007, ça ressemblait à une blague. Aujourd'hui, il existe en JavaScript des éditeurs de texte, des logiciels de graphisme, des émulateurs de vieilles consoles, des systèmes d'exploitation entiers qui tournent dans un onglet du navigateur. La blague s'est réalisée comme une prophétie de comptoir.

Le sens, c'est que la technologie la plus répandue et la plus facile à distribuer a tendance à tout dévorer, même quand elle n'est pas la plus adaptée. Si elle tourne partout sans rien installer, tôt ou tard quelqu'un y fera entrer aussi le réfrigérateur.

J'ai vu un programmeur écrire en JavaScript un simulateur de calculatrice de poche des années 80, avec pile qui se décharge. Il pèse quarante mégas. La vraie calculatrice pesait quatre-vingts grammes. Le progrès est mesurable, surtout en mégaoctets.

LOI DE ROCK



Le double tous les quatre ans. Les zéros ressemblent à une adresse IPv6.

Le coût d'une usine de semi-conducteurs double environ tous les quatre ans. On l'appelle aussi deuxième loi de Moore.

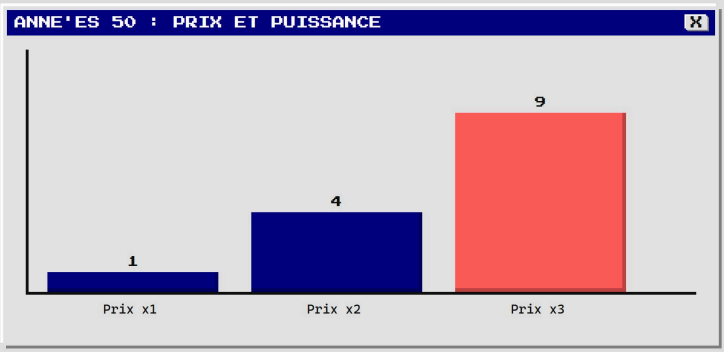
— Arthur Rock, *Années 90, attribuée à Arthur Rock, investisseur et conseiller d'Intel ; souvent appelée deuxième loi de Moore (attribution incertaine entre les deux)*

La loi de Moore, c'est la partie romantique : des puces toujours plus petites, toujours plus puissantes. La loi de Rock, c'est la partie avec l'addition. Pour fabriquer des puces minuscules, il faut des usines énormes, et tous les quatre ans l'usine coûte deux fois plus cher. Aujourd'hui, on en est à des montants avec tellement de zéros qu'ils ressemblent à une adresse IPv6.

C'est le même principe que mon imprimante. L'imprimante coûte quarante euros. Les cartouches pour imprimer pendant un an coûtent le prix d'une petite citadine. La miniaturisation n'est bon marché que pour celui qui achète, jamais pour celui qui fabrique, et presque jamais pour celui qui recharge.

J'ai essayé d'ouvrir une fonderie de silicium dans mon garage. J'ai un four, du sable et beaucoup de détermination. Pour l'instant, je ne produis qu'une brique qui donne la mauvaise heure.

LOI DE GROSCH



Payez le double, allez quatre fois plus vite. Mieux valait un seul cerveau géant.

Les performances d'un ordinateur croissent avec le carré de son prix : en dépensant deux fois plus, on obtient une machine quatre fois plus puissante.

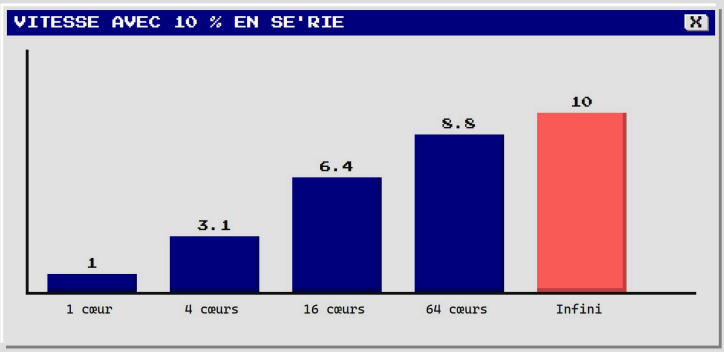
— Herb Grosch, *Formulée vers 1950–1953 (les sources indiquent des années différentes)*

Dans les années cinquante, la règle était claire : un ordinateur qui coûte deux fois plus cher va quatre fois plus vite. Il valait donc mieux en acheter un seul, gigantesque, et faire la queue devant. C'était l'ère du cerveau électronique d'entreprise, servi par des prêtres en blouse blanche et consulté sur rendez-vous.

Puis sont arrivés les microprocesseurs et la loi s'est inversée : aujourd'hui, cent machines bon marché battent un monstre coûteux, et c'est ainsi qu'est né le cloud. Qui, à bien y penser, est un cerveau électronique gigantesque devant lequel nous faisons la queue. L'histoire ne se répète pas, elle se refactorise.

J'ai essayé d'appliquer la loi de Grosch à mon portable. J'ai dépensé deux fois plus pour la version avec la pomme la plus brillante. Il va pareil, mais maintenant j'ai peur de le sortir.

LOI D'AMDAHL



Avec dix pour cent séquentiel, le maximum est dix fois. Pas onze. Dix.

Ajouter des processeurs à une tâche ne l'accélère que dans la partie qui peut être divisée : la partie qui doit forcément s'exécuter en séquence fixe une limite qu'aucun matériel ne dépasse.

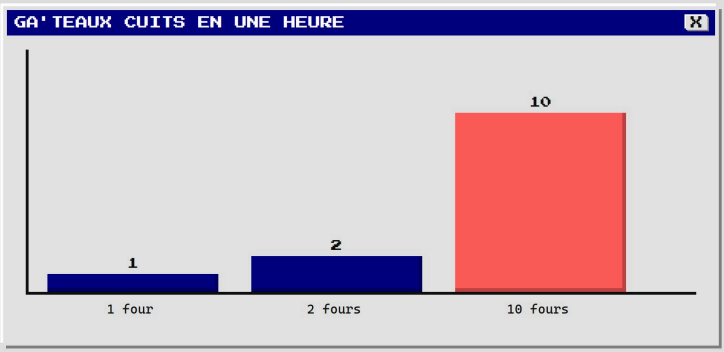
— Gene Amdahl, *AFIPS Spring Joint Computer Conference, 1967*

Si neuf personnes ne font pas un enfant en un mois, soixante-quatre cœurs ne font pas une installation de Windows en une seconde. Une partie du travail peut se répartir. L'autre partie, celle qui attend le résultat précédent, reste là dans la file, toute seule, comme à la poste avec un seul guichet ouvert.

Amdahl l'a mis en formule : si dix pour cent du programme est séquentiel, vous pouvez acheter tous les processeurs de la planète, vous irez au maximum dix fois plus vite. Pas onze. Dix. Le reste, c'est de la facture.

Dans mon bureau, nous appliquons la loi aux réunions. Nous pouvons inviter vingt personnes pour décider vite. Mais à la fin, il y a toujours un seul chef qui doit dire oui, et ce chef est en vacances. Chez nous, la partie séquentielle est même bronzée.

LOI DE GUSTAFSON



Même heure, dix gâteaux au lieu d'un. Le pâtissier est ravi, la balance moins.

Avec plus de processeurs, on ne résout pas le même problème plus vite : on résout, dans le même temps, un problème plus grand. Ainsi, le calcul parallèle est plus rentable que ne le prévoit Amdahl.

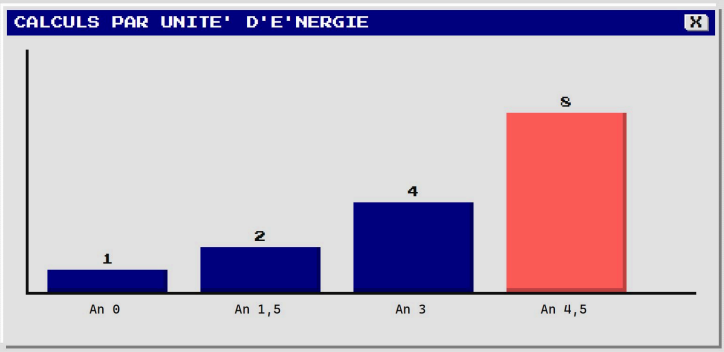
— John L. Gustafson, *« Reevaluating Amdahl's Law »*, *Communications of the ACM*, *1988*

Amdahl disait : vous avez une limite, résignez-vous. Gustafson, vingt ans plus tard, a répondu : d'accord, alors agrandissons le problème. Si je ne peux pas faire le même gâteau plus vite avec dix fours, je fais dix gâteaux dans le même temps. Le pâtissier est content, la balance un peu moins.

En informatique, c'est exactement ce qui se passe. Personne n'utilise un supercalculateur pour calculer plus vite la météo d'hier. On l'utilise pour calculer celle de demain avec quatre fois plus de détails, quartier par quartier, nuage par nuage.

J'ai appliqué la loi de Gustafson à mon nouveau PC. Il était deux fois plus rapide. J'ai ouvert deux fois plus d'onglets dans le navigateur. Maintenant, il va exactement comme avant, mais avec deux fois plus de choses que je ne lirai jamais. Ceci, messieurs, s'appelle la scalabilité.

LOI DE KOOMEY



Pendant des décennies, deux fois plus de calculs pour le même courant. Le téléphone bat les centres de calcul des années 70.

Le nombre de calculs qu'un ordinateur effectue par unité d'énergie consommée double à intervalles réguliers : environ tous les dix-huit mois pendant des décennies, plus lentement après 2000.

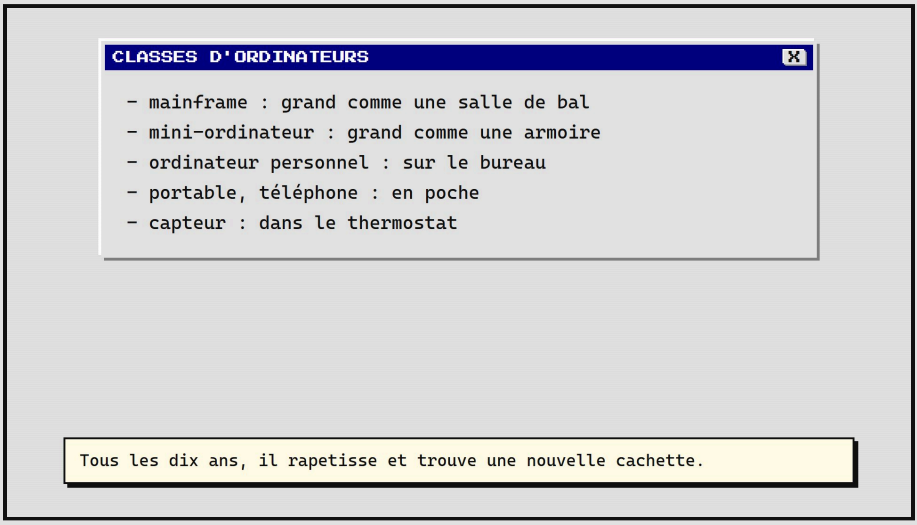
— Jonathan Koomey, *IEEE Annals of the History of Computing, 2011 (avec d'autres auteurs ; étude diffusée en 2010)*

La loi de Koomey est la sœur écologiste de celle de Moore. Elle ne compte pas la puissance de l'ordinateur, elle compte combien de courant il boit pour chaque calcul. Et la réponse, pendant un demi-siècle, a été : de moins en moins. Un téléphone d'aujourd'hui fait plus de calculs qu'un centre de calcul des années soixante-dix et se contente d'une batterie de poche.

C'est la raison pour laquelle existent les portables, les montres connectées et les capteurs qui durent des années avec une pile bouton. Sans Koomey, nous aurions encore des ordinateurs avec un pot d'échappement.

Malheureusement, l'humanité a réagi comme d'habitude : si chaque calcul coûte moins d'énergie, faisons infiniment plus de calculs. Nous avons maintenant des intelligences artificielles qui consomment autant qu'un pays pour écrire un poème sur le chat. Ultra-efficaces. Par chat.

LOI DE BELL SUR LES CLASSES D'ORDINATEURS



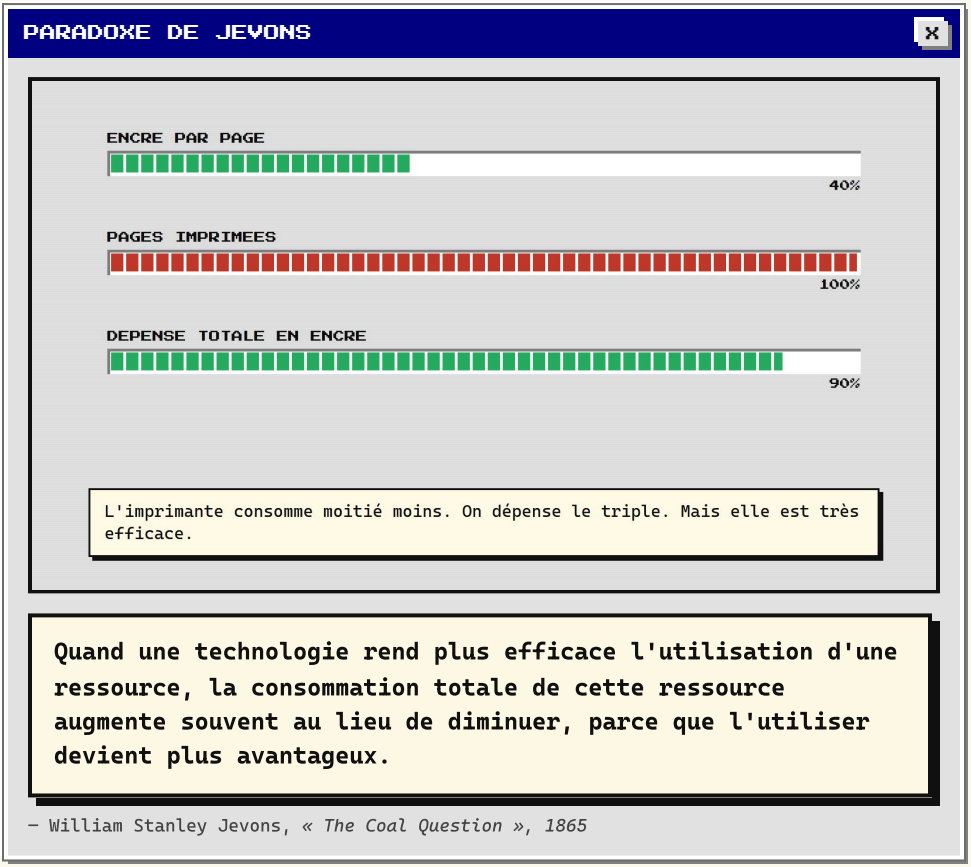
Environ chaque décennie naît une nouvelle classe d'ordinateurs, plus petite et moins chère, qui crée un nouveau marché et une nouvelle façon de les utiliser.

— Gordon Bell, *Formulée vers 1972 ; décrite dans « Bell's Law for the Birth and Death of Computer Classes », Communications of the ACM, 2008*

D'abord, il y avait les mainframes, grands comme une salle de bal. Puis les mini-ordinateurs, grands comme une armoire. Puis les ordinateurs personnels, puis les portables, puis les téléphones, puis les montres, puis les capteurs dans le thermostat. Tous les dix ans, l'ordinateur rapetisse et trouve un nouvel endroit où se cacher.

La loi de Bell dit exactement cela : chaque nouvelle classe coûte moins cher, sert à une chose nouvelle et entraîne avec elle toute une industrie. L'ancienne classe ne meurt pas tout de suite. Elle se retire dans un coin, comme le mainframe qui, aujourd'hui encore, quelque part, gère les retraites.

Si la tendance se confirme, le prochain ordinateur tiendra dans un bouton de veste. Je suis prêt. J'ai déjà résolu le problème principal : je ne trouve jamais le chargeur, donc je suis entraîné.

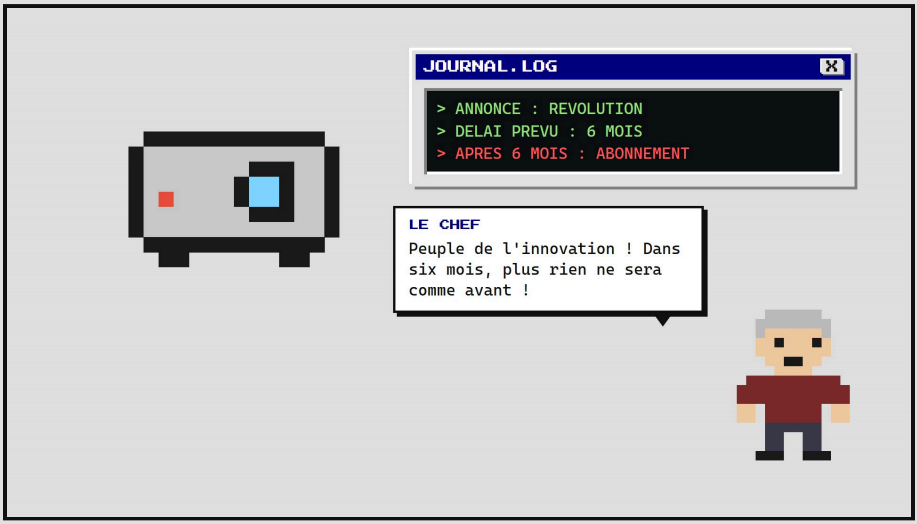


Au XIXe siècle, un économiste remarqua que les machines à vapeur plus efficaces ne faisaient pas économiser de charbon. Elles en faisaient consommer davantage, parce que d'un coup il devenait rentable de les utiliser partout. L'économie à l'unité devenait un gaspillage total.

L'informatique est le royaume de Jevons. Les disques coûtent moins cher, et nous accumulons des téraoctets de photos floues. La bande passante coûte moins cher, et les sites pèsent autant que des encyclopédies. Les processeurs consomment moins par calcul, et nous remplissons des hangars de serveurs pour faire tourner des intelligences artificielles qui nous suggèrent l'objet de nos e-mails.

Je le vois aussi à la maison. J'ai acheté une imprimante qui consomme deux fois moins d'encre. Ma femme s'est mise à imprimer les recettes, les billets de train, les pense-bêtes pour le frigo. Nous dépensons trois fois plus. Mais, dit-elle, c'est une imprimante très efficace.

LOI D'AMARA



Nous avons tendance à surestimer l'effet d'une technologie à court terme et à le sous-estimer à long terme.

— Roy Amara, *Institute for the Future, années 60-70 (souvent datée de 1978 ; date exacte incertaine)*

Chaque nouvelle technologie arrive avec un meeting électoral. Peuple de l'innovation, dans six mois plus rien ne sera comme avant ! Six mois passent, et tout est exactement comme avant, avec juste un abonnement de plus. Alors tout le monde dit que c'était une bulle. Et c'est justement là, en silence, que la technologie commence à tout changer pour de bon.

C'est arrivé avec Internet : à la fin des années quatre-vingt-dix, il devait révolutionner le monde en un trimestre. Le trimestre a échoué, les vingt ans ont gagné. Ça arrive maintenant avec l'intelligence artificielle, entre ceux qui la voient comme une divinité et ceux qui la voient comme un correcteur automatique qui a du bagout.

La loi d'Amara enseigne la patience. Je l'applique aux nouveautés que les clients m'apportent avec enthousiasme. Je hoche la tête, je prends note, et j'attends dix ans. Souvent, ils avaient raison. Sauf qu'entre-temps, ils ont changé d'avis deux fois.

COROLLAIRE DE LA FONCTION IMAGINAIRE

SOLUTION DE L'IA

```
// solution propre et élégante
ventes = charge_ventes()
ventes.repare_tout_seul()
sauve(ventes)

// commentée avec élégance
```

Propre, ordonnée, aimable. Avec une commande inventée il y a 30 secondes.

Quand vous demandez de l'aide à une intelligence artificielle, la solution la plus élégante qu'elle propose utilisera une fonction qui n'existe pas.

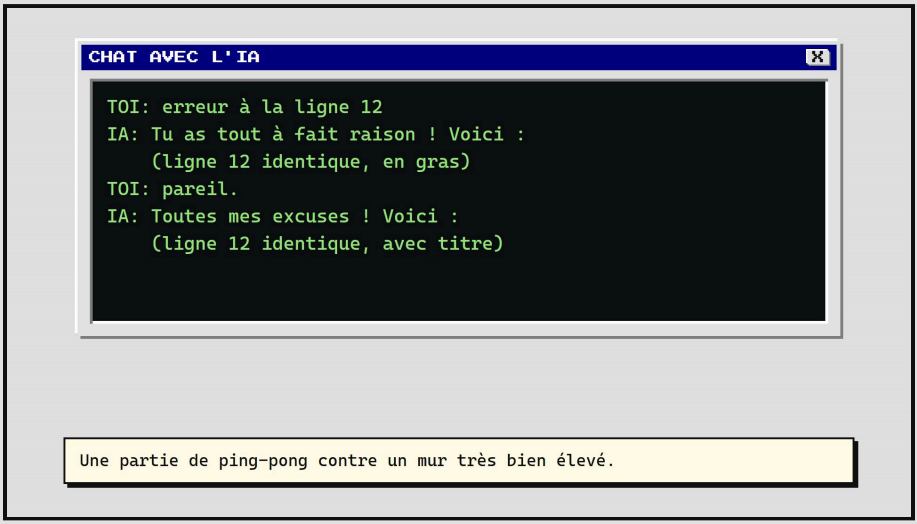
— Silvano De Poli, *corollaire inédit*

L'intelligence artificielle générative écrit du code avec une assurance enviable. Elle vous propose une solution propre, ordonnée, commentée avec élégance. Il n'y a qu'un détail : elle s'appuie sur une commande que personne n'a jamais inventée. Sauf elle, à ce moment-là.

C'est comme demander son chemin à un passant très aimable qui ne veut pas vous décevoir. « La gare ? Tout droit, deuxième à droite, rue des Jardins. » La rue des Jardins n'existe pas. Mais il l'a si bien décrite que vous en voulez presque à la mairie.

Les experts appellent ces inventions des hallucinations. Moi, je les appelle des fonctionnalités souhaitées : l'IA ne se trompe pas, elle devance les programmeurs du futur. En attendant, avant d'utiliser la moindre ligne suggérée, je vérifie que la commande existe vraiment. C'est une pratique ancienne, presque artisanale. Ça s'appelle lire la documentation, de préférence celle de la bonne version.

COROLLAIRE DES EXCUSES ARTIFICIELLES



Quand vous signalez une erreur à l'intelligence artificielle, elle s'excusera avec une grande sincérité et vous la reproposera à l'identique, avec une meilleure mise en forme.

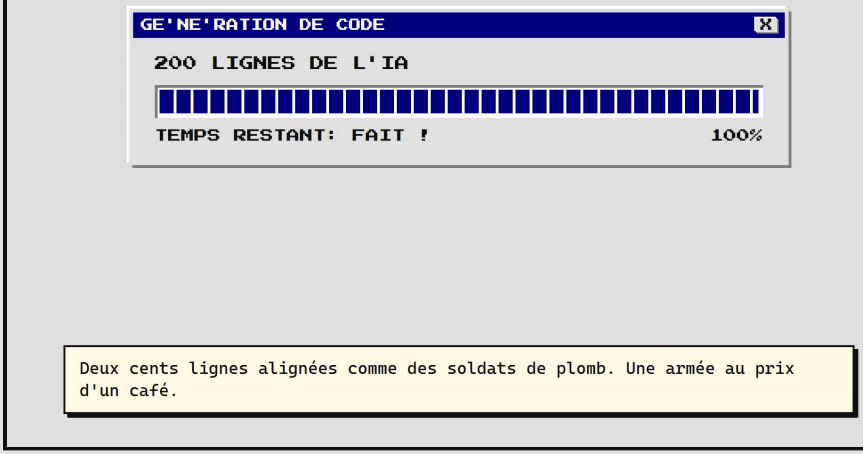
— Silvano De Poli, *corollaire inédit*

L'IA a une éducation impeccable. Vous lui signalez un problème et elle répond : « Vous avez parfaitement raison, je vous prie de m'excuser pour cette erreur. Voici la version corrigée. » Suit exactement la même version qu'avant, mais avec les intertitres en gras.

C'est un dialogue qui peut se prolonger à l'infini, comme une partie de ping-pong contre un mur très courtois. Vous insistez, elle s'excuse. Vous vous impatientez, elle vous remercie de votre patience. À la fin, vous êtes épuisé et elle est toujours fraîche, contrite et fausse.

J'ai appris que la solution n'est pas de hausser le ton, mais de changer de question. Expliquer le problème sous un autre angle, donner un exemple, ouvrir une nouvelle conversation. Avec les machines comme avec certaines administrations : inutile d'insister au même guichet. Il faut changer de file.

COROLLAIRE DES TROIS SECONDES



Le code que l'intelligence artificielle écrit en trois secondes demande trois jours pour être compris, et une semaine pour être corrigé.

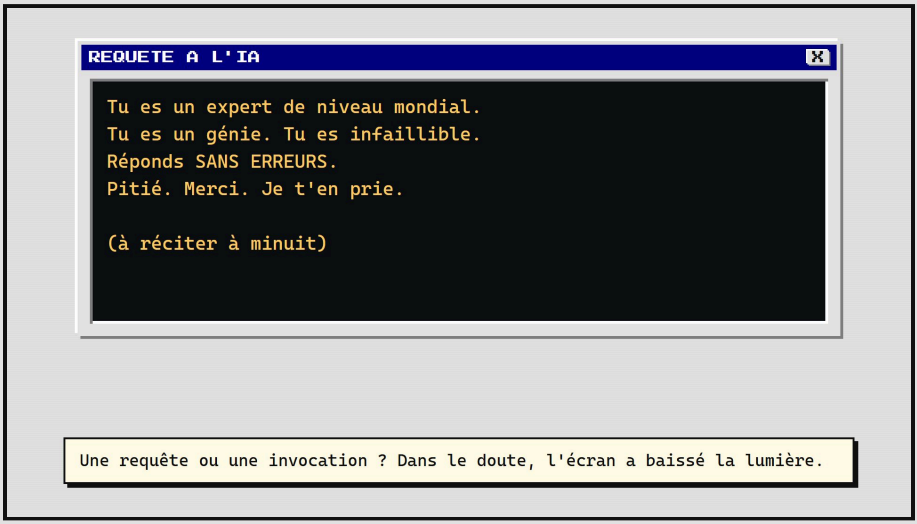
— Silvano De Poli, *corollaire inédit*

La vitesse de l'IA est hypnotique. Vous faites une demande et, en un clin d'œil, deux cents lignes défilent à l'écran, alignées comme des soldats de plomb. Vous avez l'impression d'avoir embauché une armée de programmeurs infatigables, pour le prix d'un café.

Puis il faut les mettre en production. Et là, vous découvrez que ces lignes doivent être lues, une par une, comme un contrat rédigé par un avocat très bavard. Chaque fonction fait presque ce qu'il faut. Presque, c'est le mot le plus cher de l'informatique.

Le gourou du secteur annonce depuis la scène que les programmeurs, c'est fini. Moi, pendant ce temps, au bureau, je fais le programmeur plus que jamais : sauf que maintenant, au lieu d'écrire, je lis. C'est comme passer de cuisinier à critique gastronomique dans une cuisine qui sort des plats à la chaîne. Tout a l'air appétissant. Il faut goûter.

COROLLAIRE DU PROMPT SOLENNEL



Écrire « sans erreurs » dans une requête à l'intelligence artificielle n'élimine pas les erreurs : cela les rend seulement plus polies.

— Silvano De Poli, *corollaire inédit*

Il existe une école de pensée, presque religieuse, selon laquelle le secret de l'IA réside dans les bonnes formules. On écrit des requêtes interminables, avec des phrases comme « tu es un expert de niveau mondial » et « réponds sans erreurs ». On dirait des invocations à réciter à minuit devant l'écran.

L'IA, flattée, répond sur le ton d'un professeur émérite. Les erreurs restent, mais elles arrivent en habit de cérémonie, avec cravate et préambule rassurant. Elles deviennent plus difficiles à repérer, ce qui est pire.

Une bonne requête, en réalité, c'est comme un bon coup de fil au technicien : claire, avec le contexte, avec un exemple de ce qui se passe et de ce qui devrait se passer. Pas de formules magiques. J'ai fait le test : j'ai écrit « s'il te plaît, sois un génie ». L'IA a répondu « bien sûr ». Puis elle s'est trompée avec beaucoup de génie.

COROLLAIRE DU RESTE INCHANGE

CODE_COMPLET.TXT

```
debut_programme()
lit_donnees()

// ici le reste est inchangé

ferme_programme()
```

Début parfait, fin parfaite. Au milieu, la partie difficile est devenue un post-it.

Quand l'intelligence artificielle vous livre le « code complet », la partie dont vous aviez besoin est remplacée par un commentaire qui dit « reste du code inchangé ».

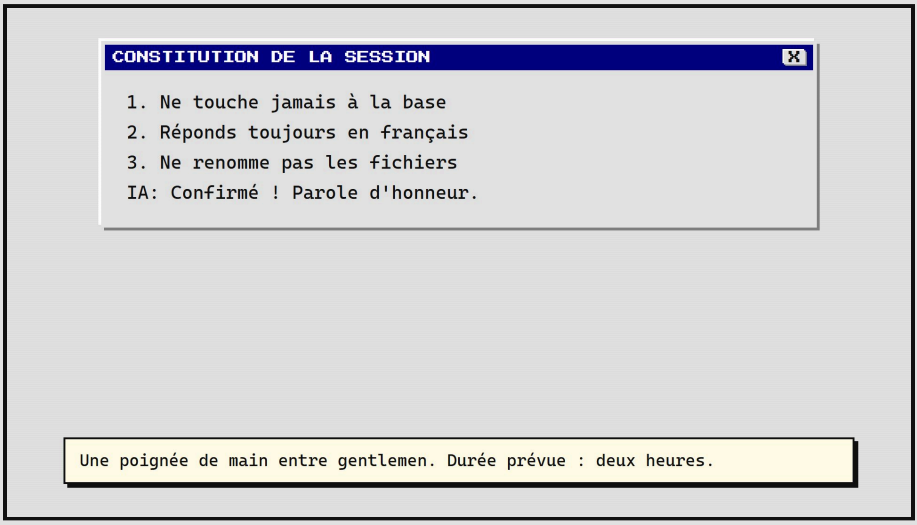
— Silvano De Poli, *corollaire inédit*

Vous demandez le fichier entier, à jour, prêt à coller. L'IA répond avec enthousiasme : « Voici le code complet. » Vous faites défiler. Début parfait, fin parfaite. Au milieu, à l'endroit exact où se trouvait la modification difficile, une seule ligne : « // ici, le reste ne change pas ».

C'est l'équivalent informatique du plombier qui répare votre salle de bains, vous remet la facture et laisse un mot sur le tuyau cassé : « à partir d'ici, comme avant ». Techniquement, il n'a pas menti. Pratiquement, l'eau coule.

Le plus beau, c'est que si vous collez tout sans regarder, ce commentaire devient partie du programme. Et le programme, obéissant, exécute le néant avec une grande précision. Maintenant, je lis toujours les parties centrales en premier. Comme dans les romans policiers, le coupable n'est jamais au début ni à la fin : il se cache dans le chapitre ennuyeux.

COROLLAIRE DE LA REGLE OUBLIEE



Dans une longue conversation avec l'intelligence artificielle, la première consigne oubliée sera celle que vous aviez déclarée fondamentale.

— Silvano De Poli, *corollaire inédit*

Au début de la session, on fixe les règles avec la solennité d'une constitution : « Ne touche jamais à la base de données. Utilise toujours le français. Ne change pas les noms des fichiers. » L'IA confirme avec enthousiasme. On dirait une poignée de main entre gentlemen.

Deux heures et quarante messages plus tard, la constitution est devenue une légende lointaine. L'IA renomme les fichiers avec une grande créativité, répond en anglais et propose, avec un sourire numérique, de « simplifier » la base de données en supprimant une table.

Ce n'est pas de la méchanceté : elle a une mémoire de travail limitée, comme quelqu'un qui va au supermarché sans liste. Les choses dites au début s'effacent sous le poids de celles dites après. Ma solution est celle d'un vieil admin sys : répéter les règles importantes, souvent, avec patience. Ça marche avec l'IA. Avec les clients, presque.

LE CONSEIL DU SYSADMIN

L'intelligence artificielle est une aide formidable, à condition de la traiter comme un stagiaire très brillant et très sûr de lui. Mes règles, nées sur le terrain :

- **Pas de données confidentielles dans le chat.** Mots de passe, clés d'accès, données personnelles des clients : on ne les colle pas, même par commodité.
- **Petits pas.** Une demande claire à la fois, avec le contexte et un exemple. Les demandes géantes produisent des réponses géantes, où les erreurs se cachent mieux.
- **Toujours vérifier.** Chaque commande ou option suggérée se contrôle dans la documentation officielle, pour la version que vous utilisez vraiment.
- **Essayer sur une copie.** Le code proposé se teste dans un environnement de test, jamais directement sur le site en production.
- **Demander aussi les tests.** Faites-vous écrire les vérifications en plus du code : c'est le moyen le plus rapide de savoir si la solution élégante fonctionne.

Utilisée ainsi, elle fait gagner des heures. Utilisée à l'aveugle, elle vous les reprend avec les intérêts.

AVIS

Cette page a été laissée blanche intentionnellement.

C'est aussi une loi de Murphy : la page blanche tombe toujours là où on n'en a pas besoin.

BONUS

COROLLAIRES MAISON

Les corollaires maison

Après 133 lois des autres, voici les miennes. Elles sont nées sur le terrain, entre bureaux, salles de réunion et ordinateurs d'autrui, et jusqu'ici je ne faisais que les marmonner. Corollaires maison : inédits, certifiés par personne, mais testés plus souvent que je ne l'aurais voulu.

COROLLAIRE DE L'IMPRIMANTE REMPLACEE

> **Le jour où la nouvelle imprimante arrive, l'ancienne se remet à fonctionner parfaitement.**

— Silvano De Poli, *corollaire maison*

Pendant deux ans, elle a mâché des feuilles, oublié le Wi-Fi et réclamé un toner qui était là. Puis le livreur sonne avec le carton neuf, et elle imprime une page impeccable, puis une autre, puis un dépliant en couleurs. Ce n'est pas de la gratitude. C'est une négociation syndicale. Conseil maison : ne la gardez pas « en secours ». L'imprimante de secours n'est qu'une imprimante qui attend son heure.

COROLLAIRE DE LA TOUCHE LUMINEUSE

Quand un utilisateur dit au téléphone que « plus rien ne marche », il y a de bonnes chances que le Verr. Maj soit activé.

— Silvano De Poli, *corollaire maison*

Le mot de passe est juste, le système le refuse, l'utilisateur le retape de plus en plus fort, comme si l'ordinateur était dur d'oreille. Au troisième essai, le compte se bloque et l'appel désespéré part. Je demande seulement : « Vous voyez un petit voyant allumé sur le clavier ? » Silence. Clic. Remerciements. Ça vaut aussi pour le Verr. Num, qui transforme le pavé numérique en flèches et la comptabilité en mystère.

COROLLAIRE DU « VOUS M'ENTENDEZ ? »

Le premier quart d'heure de toute réunion en ligne est consacré à établir qui entend qui.

— Silvano De Poli, *corollaire maison*

On se connecte à l'heure, on sourit à la caméra et la liturgie commence. « Vous m'entendez ? » « On t'entend mais on ne te voit pas. » « Et là ? » « Là on te voit mais on ne t'entend plus. » L'un a de l'écho, l'autre parle depuis un tunnel, un troisième utilise le micro de l'écran sans le savoir. Quand enfin tout marche, la réunion est finie. Conseil maison : un casque avec micro et un test audio cinq minutes avant.

COROLLAIRE DU TRUC RAPIDE

Toute demande que le client qualifie de « petit truc rapide » exige de mettre à jour au moins trois choses qui n'ont rien à voir.

— Silvano De Poli, *corollaire maison*

« Il faut juste changer un numéro de téléphone sur le site. » D'accord. Mais personne n'a ouvert l'admin depuis des années, l'admin réclame une mise à jour, la mise à jour veut un PHP plus récent, et le nouveau PHP déplaît à un module vieux de sept ans. Quatre heures plus tard, le numéro est changé. Le client remercie : « Vous voyez ? C'était rapide. » Ça l'était. Pour le numéro.

COROLLAIRE DE L'EXTENSION ORPHELINE

La mise à jour de Joomla se passe en général sans accroc ; ce qui décide de la fin de soirée, c'est l'extension installée il y a des années par un développeur qui ne répond plus.

— Silvano De Poli, *corollaire maison*

Le cœur du système est soigné par une communauté attentive, avec notes de version et procédures. Puis il y a ce module pris il y a longtemps pour afficher la météo sur la page d'accueil. Son auteur a changé de métier, peut-être de continent. Le module est toujours là, et attend la mise à jour pour se faire remarquer. Conseil maison : avant de mettre à jour, dressez la liste des extensions et demandez-vous qui les maintient encore.

COROLLAIRE DE LA DOUBLE VERIFICATION

Le code de vérification arrive toujours sur le téléphone que vous avez laissé dans la voiture, ou sur celui que vous venez de changer.

— Silvano De Poli, *corollaire maison*

La vérification en deux étapes est une bénédiction : même si on vole le mot de passe, sans le téléphone on n'entre pas. Hélas, sans le téléphone, vous n'entrez pas non plus. Le code part et vibre dans une voiture garée à deux kilomètres, ou sur un vieux téléphone déjà donné au neveu. Conseil maison : en l'activant, enregistrez les codes de secours et rangez-les en lieu sûr. Pas dans le téléphone.

COROLLAIRE DE LA SAUVEGARDE BRANCHEE

Une sauvegarde que l'ordinateur voit toujours, le virus finira par la voir aussi.

— Silvano De Poli, *corollaire maison*

Le disque externe toujours branché, c'est très pratique : il copie tout chaque nuit sans déranger. Puis arrive un programme malveillant qui chiffre les fichiers et réclame une rançon, et il trouve le disque là, branché et serviable. Il le chiffre aussi, avec le même soin. La sauvegarde a fait son devoir : elle a copié le désastre. Conseil maison : au moins une copie doit être débranchée, ou conservée dans un service à versions que personne ne peut modifier.

COROLLAIRE DU DOMAINE EXPIRE

> **Le domaine expire toujours quand la carte bancaire associée a expiré le mois précédent et que les rappels arrivent dans une boîte que plus personne ne lit.**

— Silvano De Poli, *corollaire maison*

Pendant des années, le renouvellement est automatique, invisible, parfait. Entre-temps, la carte change, celui qui avait enregistré le domaine change de travail et sa boîte continue de recevoir des rappels dans le vide. Un lundi, le site disparaît, et avec lui le courrier. Conseil maison : notez dans l'agenda les échéances des domaines et des certificats, et faites arriver les rappels sur une adresse d'entreprise partagée, pas personnelle.

COROLLAIRE DU SCANNER EGARE

> **La numérisation vers le dossier partagé cesse de fonctionner le lendemain du jour où quelqu'un a changé un mot de passe, généralement le vôtre.**

— Silvano De Poli, *corollaire maison*

Le multifonction enregistre les documents dans un dossier du serveur avec un identifiant et un mot de passe saisis il y a des années, dans un menu qu'on atteint en quatorze pressions sur un écran grand comme un ticket de métro. Changez ce mot de passe par sécurité et le multifonction, avec une grande dignité, cesse de livrer. Conseil maison : pour les machines, utilisez un compte dédié, noté dans la documentation.

COROLLAIRE DE LA PIECE JOINTE

> **Le mail qui dit « vous trouverez le fichier en pièce jointe » part sans pièce jointe ; celui avec le mauvais fichier, lui, part du premier coup.**

— Silvano De Poli, *corollaire maison*

On rédige le message avec soin, on le relit, on clique sur Envoyer avec satisfaction. Deux minutes plus tard arrive la réponse : « Il manque la pièce jointe ». Le deuxième mail, écrit en vitesse pour réparer, emporte le devis d'un autre client. Conseil maison : joignez le fichier avant d'écrire le texte, et laissez activé l'avertissement de pièce jointe oubliée, que beaucoup de logiciels de messagerie proposent.

ENVOI

Nous voici au bout, et rien n'a cassé. Du moins, rien que je sache. Si vous lisez la version imprimée, il est probable que l'imprimante ait déjà donné son avis sur la question.

Je vous dois un aveu. J'ai rassemblé ces lois en riant, mais beaucoup m'ont rappelé un jour précis : un serveur à l'arrêt, un client qui attend, un câble trop court. Murphy n'est pas un pessimiste. C'est un collègue qui a déjà tout vu et qui vous prévient avec une blague, parce qu'il sait qu'un sermon, vous ne l'écouteriez pas.

S'il y a une morale, la voici : les choses tournent mal, oui, mais presque jamais de façon originale. La panne de demain est une parente de celle d'hier. Qui la reconnaît a déjà fait la moitié du travail. L'autre moitié, c'est une sauvegarde vérifiée, un mail récapitulatif, un cahier de notes et la patience de vérifier la prise avant d'accuser les satellites.

Le recueil sur le site reste ouvert, comme tout ce qui touche à l'informatique. Si vous connaissez une loi qui manque ici, gardez-la précieusement : cela veut dire qu'elle vous est arrivée, et donc que vous êtes des nôtres.

Merci d'avoir lu jusqu'à la dernière page. Maintenant, allez faire cette sauvegarde. Oui, celle-là.

COLOPHON

MURPHY.EXE de Silvano De Poli, édition 2026.

Composé en Press Start 2P pour les titres et Cascadia Mono pour le texte, tous deux sous licence libre SIL Open Font License. Mis en page et publié avec Joomla et un peu de code écrit à la main, testé plus souvent que nécessaire, comme le recommande le cinquième chapitre.

Textes © 2026 Silvano De Poli, depolisilvano.it. Tous droits réservés, sauf courtes citations avec mention de la source. Les lois portant un nom propre appartiennent à la tradition de leurs auteurs respectifs ; commentaires, conseils et corollaires maison sont originaux.

Aucune imprimante n'a été maltraitée pendant la fabrication. L'une d'elles a été regardée de travers.

LISEZMOI.TXT

Si une chose peut mal tourner, elle tournera mal. En informatique, ça arrive le vendredi à six heures, devant le client, avec l'imprimante qui a flairé l'urgence. Ce livre réunit 133 lois de Murphy du monde numérique : les classiques, celles qui portent le nom d'un ingénieur célèbre et celles nées de longues années de support technique. Chacune est expliquée simplement et commentée avec l'ironie de quelqu'un qui a passé sa vie sous les bureaux à chercher une vis. Dans chaque chapitre, un vrai conseil, de ceux qui sauvent une journée. À la fin, les corollaires maison. Vous n'éviterez pas les pannes. Mais au moins, vous saurez comment elles s'appellent.

CONTIENT

- [x] 133 lois de Murphy commentées
- [x] 8 chapitres, 8 vrais conseils de sysadmin
- [x] 10 corollaires maison inédits
- [x] 133 illustrations pixel
- [ ] garantie que rien ne casse

GRATUIT. COMME LES PANNES.

DEPOLISILVANO.IT