

Colloque *Faut-il supprimer l'erreur* – Université Catholique de Lille
7-8 Septembre 2020
Universalité de l'erreur

Universalité de l'erreur, résumé

Bien souvent, la nature n'est pas conforme aux représentations que nous nous en faisons. Jusqu'au XVI^e siècle, nous avons vécu – en Occident tout au moins – dans un cosmos qui était celui des Grecs, dans lequel Aristote pouvait affirmer que « La nature a horreur du vide », mais en creusant des puits de plus de 10 mètres de profondeur les puisatiers ne pouvaient que constater que la colonne d'eau se désamorçait naturellement et qu'en haut il subsistait un espace qu'on ne pouvait pas qualifier autrement que de « vide ».

Notre rapport à la vérité n'a jamais cessé d'évoluer, mais dès l'époque des Sages de la Grèce nous savions distinguer les vérités démonstratives, les vérités empiriques et les simples opinions. Le savoir démonstratif avait été consigné par Euclide dans ses *Eléments* qui ont traversé les âges, jusqu'à nous. Le savoir empirique, fondé sur l'observation de la réalité, n'a cessé de se perfectionner avec les architectes et ingénieurs sous la forme des « Arts » donnant naissance à une ingénierie qui allait se fusionner avec le savoir démonstratif, les mathématiques, au XIX^e siècle. La réalité elle-même évolue car elle dépend des instruments avec lesquels on l'observe. Avec les microscope et les télescopes nous voyons des choses que notre œil ne peut pas distinguer. L'échelle de l'observation devient un critère de vérité.

Avec les opinions, la vérité repose sur un critère de consensus qui ne relève ni du savoir démonstratif, ni du savoir empirique. La formule bien connue de J-J. Rousseau « L'homme est naturellement bon, c'est la société qui le corrompt » est un exemple d'opinion dont la « vérité » est problématique. Toutefois, une opinion qui serait franchement contredite par les savoirs démonstratif et/ou empirique deviendrait ipso facto suspecte.

« La nature aime à se cacher » disait Héraclite dans un de ses célèbres fragments. Pourtant nous avons su mettre à notre service certains des processus qu'elle nous procure et construire des objets/systèmes d'une incroyable complexité, prodigieusement efficaces, qui aujourd'hui nous environnent de toute part. Mais de temps en temps cette belle mécanique nous réserve de mauvaises surprises comme celles que nous vivons présentement avec le Covid-19. Et nous redécouvrons cette vérité universelle, régulièrement oubliée, que l'erreur est partout, que nos connaissances ne sont qu'approximatives et incomplètes.

Il n'y a que dans le monde mathématique que l'on peut parler de certitude, du moins si on en respecte les règles. Pour le reste, il n'y a que des probabilités, mais pas toujours. Nous-mêmes dans nos activités quotidiennes nous faisons beaucoup d'erreurs. Nous prenons des décisions inappropriées, notre libre arbitre nous permet même de mentir, individuellement et collectivement. Des statistiques régulièrement vérifiées nous disent qu'en moyenne nous faisons entre 5 et 10 erreurs par heure d'activité, soit en arrondissant une erreur toutes les 10 minutes. La logique pour ceux qui la connaissent est d'un maniement délicat, quant à la traduction c'est un champ de mines entre faux-sens et contresens. Entre le VRAI et le FAUX, il y a les immensités océaniques de l'insignifiance.

Notre problème n'est donc pas tant de supprimer les erreurs, ce qui reviendrait à supprimer la vie, mais d'apprendre à cohabiter avec elles pour survivre aux aléas, en jouant intelligemment avec les possibilités que nous offre la nature, y compris nous-mêmes.

Universalité de l'erreur – Le message de l'ingénierie système

De la nature des erreurs

La science contemporaine, avec la mécanique quantique, nous a appris qu'au cœur de ce que nous appelons « matière », il y avait une incertitude fondamentale que les physiciens dans le premier tiers du 20^e siècle ont modélisé sous la forme de relations dites d'« incertitudes » avec entre autre Werner Heisenberg, puis Erwin Schrödinger assimilant les « particules » de la matière ordinaire que nous percevons à des ondes de probabilités. Toutefois, à notre échelle, les processus avec lesquels nous sommes en interactions paraissent extraordinairement stables et pour tout dire quasi éternel, ce qui faisait dire à Einstein, dans sa controverse avec Bohr, « Il me plait de savoir que la Lune, même si je ne la regarde pas, continue d'exister ».

L'échelle atomique, soit 0,1 nanomètre – abréviation nm –, nous est totalement inaccessible. Ce que nous percevons à notre échelle ne sont que des moyennes, des effets statistiques. Pour matérialiser ce paradoxe, rien de mieux que de réfléchir au nombre phare de cette échelle, le nombre dit d'Avogadro, car estimé pour la première fois par ce savant italien en 1811, soit dans nos meilleurs estimations : $N_A = 6,022140857(74) \dots \times 10^{23}$ par atome/molécule gramme, par exemple 18 grammes d'eau ; notons que pour la précision obtenue – 10 chiffres significatifs sûrs –, l'incertitude porte sur 14 chiffres, soit de l'ordre de cent mille milliards de particules. A partir du 20 mai 2019, ce nombre est devenu un axiome de la physique qui sert à définir la masse dans le système international des unités de mesures de la physique.

Cependant, grâce aux travaux des physiciens de la matière condensée [en anglais *solid state physics*], il a été possible de créer une ingénierie atomique, substrat des nanotechnologies, à une échelle de 10 nanomètres qui est celle des organites constitutifs de la machinerie cellulaire [ribosomes, mitochondries, etc.] ; le pas de l'hélice de l'ADN étant de 3 à 4 nanomètres. Cette ingénierie a permis à la société NVIDIA de produire une puce électronique pour sa machine Quadro GV100, utilisée en IA et en génomique, qui totalise 21 milliards de transistors sur une surface de 835 mm² de silicium, soit 25 millions par mm², une performance extraordinaire qui mérite d'être soulignée.

Les propriétés qui font du silicium le matériau phare des technologies de l'information proviennent de défauts de structure dans le réseau cristallin du silicium proche de celui du carbone dans sa forme diamant, défauts qui correctement organisés grâce à cette ingénierie nouvelle vont produire un effet spécifique dit de semi-conducteur ce qui va permettre de faire des « portes logiques », c'est-à-dire des interrupteurs, de très petite taille consommant très peu d'énergie. A cette échelle, nous pouvons imaginer nos transistors comme de petits volumes de 10 à 15 nm de pas, c'est-à-dire un cristal de silicium rassemblant un collectif de 1 à 4 millions d'atomes. Nous savons que nous sommes désormais proche d'une limite qui sera indépassable compte tenu des lois de la physique.

Voir les planches N°2, 3 et 4 dans l'annexe

En fait, ce que nous appelons erreurs ou défauts ne sont que des différences entre les modèles idéalisés que nous nous faisons de la réalité et cette même réalité. Pour les savants grecs le cercle était l'image de la perfection et les planètes – astres « errants », dans leur cosmos statique créé une fois pour toute – ne pouvaient se mouvoir que sur des trajectoires circulaires, que ce soit dans le modèle géocentrique ou dans le modèle héliocentrique. Pour Pythagore et les philosophes milésiens, les nombres étaient des proportions harmonieuses, comme les rapports des sons musicaux, représentés par ce qu'on appelle aujourd'hui des fractions, soit

$\frac{a}{b}$, d'où leur stupéfaction lorsqu'il découvre que la diagonale d'un carré ne pourra jamais être représentée par une fraction. Johan Kepler, un savant de la Renaissance, le « mathematicus » de l'Empereur Rodolphe, découvre que les planètes, grâce aux mesures effectuées par

l'astronome danois Tycho Brahe, ne se déplace pas sur des cercles mais sur des ellipses, courbes bien connues des mathématiciens grecs sur un plan spéculatif, mais qui ont le défaut d'être plus complexe et moins « belle » que le cercle. Le modèle géocentrique se fissure, mais il faudra attendre encore, grâce à de nouvelles mesures encore plus précises, faites par l'astronome William Herschel, pour être certain que ce sont les planètes qui tournent autour du soleil et non l'inverse, une nouvelle hypothèse seule à même d'expliquer le phénomène observé de parallaxe.

Ce qui se découvre progressivement est que le rôle dévolu aux modèles est de détecter les différences entre ce que le modèle permet de voir et la réalité de ce qui est vu. La vérité réelle ou supposée d'un modèle jugé « beau » par sa simplicité ne peut être tranchée que par l'expérimentation et la capacité de prédiction. Au 19^e siècle le modèle mécanique de l'univers initialement élaboré par Newton, perfectionné par Lagrange permet par le simple calcul à partir de variations mesurées de prédire l'existence de la planète Neptune, obtenu par Urbain Le Verrier. Le modèle permet de s'approcher progressivement de la réalité. En soi, on ne peut jamais dire que le modèle est « vrai » ; il est toujours une approximation. La perception par modèle interposé ne deviendra vraiment claire sur le plan scientifique que dans la deuxième moitié du 20^e siècle¹, à la condition expresse de ne jamais confondre le modèle et la réalité.

→ **Le modèle est une construction** dite hypothéticodéductive faite en langage mathématique dont il doit respecter rigoureusement les règles – c'est une « grammaire » explicite qui elle-même peut évoluer – pour être utilisée sans risque d'inconsistance.

Dans cette approche, le modèle apparaît comme le moyen le plus sûr pour formuler le plus clairement possible des hypothèses que l'on pourra ensuite soumettre à l'expérience, c'est-à-dire de poser des questions à la nature et d'observer les réponses données. Nous avons ainsi un critère permettant de distinguer une hypothèse, vraie ou fausse, d'une simple opinion subjective qui ne concerne que son auteur. Dans le langage des scientifiques c'est ce qui s'appelle une axiomatisation.

Valider une hypothèse

Pour illustrer ce qui vient d'être dit, donnons un exemple de raisonnements à partir de modèles hypothétiques, sous-jacents à toute construction, matérielle et/ou immatérielle, qui montrent comment, si on raisonne mal, on peut facilement sombrer dans l'erreur avérée, c'est-à-dire du FAUX que l'on croit VRAI, et/ou dans l'océan de l'insignifiance du pur non sens².

Voir la planche N°5 dans l'annexe

Qu'est-ce qu'une hypothèse ?

Concernant le sens qui doit être donné au mot hypothèse on peut se référer aux travaux du chanoine George Lemaître concernant l'« hypothèse de l'atome primitif³ », lui-même s'abritant sous l'autorité de Poincaré et de son ouvrage d'épistémologie *La science et l'hypothèse*, paru en 1902, suivi de nombreuses rééditions. Ce que Poincaré veut clarifier c'est la distinction entre une simple opinion, ce qui est du ressort du libre arbitre de chacun, a priori non démontrable, et ce que l'on peut appeler une vérité scientifique, c'est-à-dire quelque chose que l'on peut considérer comme vrai, sous certaine réserve que Poincaré va préciser ; et

¹ On peut considérer que la conférence de von Neumann donnée à l'université Columbia, en 1954, *Method in the physical sciences*, met un point final à un questionnement qui aura duré quelques siècles. Sur l'aspect historique, voir Paolo Rossi, *Aux origines de la science moderne*, une excellente synthèse, Points Science, 1999.

² Voir René Thom, *Prédire n'est pas expliquer*, Flammarion 1991, et l'article *Halte au hasard, silence au bruit*, Les Débats, 1980, N°3.

³ Ses travaux ont été regroupés dans un livre paru en 1946, *L'hypothèse de l'atome primitif – Essai de cosmogonie*, avec une préface magistrale de Ferdinand Gonseth. Pour approfondir la pensée de G. Lemaître voir les livres de Dominique Lambert, *Un atome d'univers – La vie et l'œuvre de Georges Lemaître*, 2000, et *L'itinéraire spirituel de Georges Lemaître*, 2007, tous deux chez Lessius

avec lui David Hilbert dans la formulation de son 6^{ème} problème : « Axiomatisation, fondée sur le modèle mathématique, de la physique », et plus particulièrement de la géométrie qu'il vient de réviser en détail pour incorporer à côté de la géométrie euclidienne que tout le monde connaît, les géométries dites non euclidiennes. La vérité scientifique implique un partage collectif au sein de la communauté scientifique, comme ce que feront les physiciens avec la théorie atomique, même si on ne « voit » pas les atomes, puis avec la mécanique quantique – et ses ondes de probabilité, tout aussi invisibles – qui suivra, même si cette dernière a plusieurs interprétations. La vérité scientifique a un aspect universel, transculturel si l'on préfère, que chacun peut s'approprier s'il en fait l'effort.

Pour illustrer la transformation d'une simple opinion en hypothèse scientifique, Lemaître va utiliser trois moments importants de la quête des hommes dans la recherche d'une réponse aux « grandes questions », Qui suis-je ?, Pourquoi y a-t-il quelque chose plutôt que rien ?, etc., avant de proposer son hypothèse. Pour cela il va interpellé successivement Kant, Buffon et Laplace.

Kant, dans son ouvrage *Histoire naturelle générale et théorie du ciel*, propose à ses contemporains une « théorie » totalement fantaisiste, fondée sur des erreurs de compréhension de ce qu'a dit Newton, – à la lueur de nos connaissances d'aujourd'hui c'est beaucoup plus clair – dont Lemaître dira tout en malice : « Kant tire de cette erreur les conséquences les plus extraordinaires montrant ainsi que lorsque des esprits puissants viennent à se tromper, ils ne se trompe pas à moitié »⁴. C'est du même calibre que la théorie des solides platoniciens pour expliquer l'ordonnement des planètes. En fait, Kant reste inconsciemment prisonnier de l'univers des Grecs, et il ne fait pas vraiment la différence entre une opinion et une hypothèse, ce qui prouve qu'il a mal lu Newton, que cependant il admire, dans ses *Principia*, basés sur le plan général des *Eléments* d'Euclide qui est le modèle de raisonnement des savants de l'époque, un modèle durable qui sera mis à jour par Hilbert. Ce genre de contresens fait planer un doute sur toute son œuvre, en particulier sa classification des catégories de la raison pure ; pour Kant l'espace et le temps sont des catégories indépendantes l'une de l'autre rebaptisées « catégories de la raison pure », une interprétation erronée des idées de Newton muet sur le sujet. Pour la clarification, il faudra attendre Einstein et Poincaré, mais l'ancrage chez les philosophes restera durable jusqu'à encore aujourd'hui. Sa compréhension de l'infini reste limitée, car il ne perçoit pas les paradoxes de l'infini qui inquiétaient tant Pascal.

Les spéculations de Kant sont le fruit de son imagination, et rien dans les faits, y compris ce qui est connu de son temps, ne peut les justifier ; sans parler des erreurs de logique. Pour en donner un aperçu, voici un court extrait de la conclusion où Kant spéculé sur les habitants des différentes planètes, p113 de l'édition *Forgotten Books* :

... Nous arrivons ainsi à l'expression de cette loi générale : la matière^[*1] dont sont formés les habitants des diverses planètes, les animaux aussi bien que les plantes, doit avant tout être d'une nature d'autant plus légère et subtile, l'élasticité des fibres et en même temps la conformation de leur corps doivent être d'autant plus parfaites que les astres sont plus éloignés du Soleil. ...

D'un côté nous voyons des créatures pensantes auprès desquelles le Groenlandais et le Hottentot serait des Newtons, et de l'autre des êtres qui regarderaient Newton comme un singe. ...

⁴ Une opinion également partagée par le physicien E. Schrödinger qui dans ses conférences de 1956 regroupées dans son livre *L'esprit et la matière*, disait, chapitre 5, *Science et religion* : « ... L'attitude de Kant vis-à-vis de la science était incroyablement naïve ... Ce qui est arrivé à un grand génie devrait être un avertissement pour tous les philosophes à venir. Il a tout simplement pensé avoir démontré que l'espace était nécessairement infini et a cru fermement qu'il est dans la nature de l'esprit humain de le doter des propriétés géométriques résumées par Euclide. ... », ce qui par ailleurs n'enlève rien à Kant dans d'autres domaines comme la morale ou l'éthique ; néanmoins, il faut être vigilant sur certaines de ses « démonstrations » !

Nous avons poursuivi jusqu'ici nos hypothèses^[*2] en nous appuyant uniquement sur les lois physiques de la nature, qui nous ont servi de fil conducteur pour maintenir nos déductions^[*3] dans le sentier de la vraisemblance et de la raison. Nous sera-t-il permis maintenant de nous écarter un instant de cette voie pour faire une excursion dans le domaine de la fantaisie ? ...

[*1] à cette époque, c'est plus proche de celle des grecs que ce que nous entendons aujourd'hui par 'matière', que Teilhard comprend fort bien. [*2] Aucun rapport avec ce que Lemaître entend par 'hypothèse', en suivant l'analyse de Poincaré. [*3] Très grave faute de logique car pour qu'une déduction soit valide, il faut être certain de la vérité des prémisses, une règle de logique médiévale qui faisait dire aux philosophes logiciens comme Pierre Abélard, Jean de Salisbury, Guillaume d'Ockham, etc. : « du faux on peut déduire n'importe quoi », ce qui est le cas ici.

Avec Buffon les choses sont différentes car Buffon est un vrai savant qui observe et essaye de comprendre ce qu'il voit en utilisant des mécanismes connus comme les chocs ou les variations des températures [ces mécanismes sont mieux représentés, aujourd'hui, par ce qu'on appelle en ingénierie système des « processus »]. Il imagine une comète assez puissante pour arracher un morceau de soleil d'où seraient issues les planètes par refroidissement. Il ignore bien sûr que le soleil est fait d'hydrogène qui se transforme en hélium, via une réaction atomique dite de fusion totalement inconnue à l'époque, voire impensable. Avec le phénomène du refroidissement des corps chauds, il commence à comprendre que l'âge de la terre ne peut pas être tiré de la généalogie des patriarches de la Bible. Il est assez lucide et prudent pour ne rien conclure de définitif, l'exact contraire de Kant qui lui ne doute de rien.

Avec Laplace, on change de niveau, car Laplace est un mathématicien de tout premier plan qui maîtrise parfaitement le raisonnement mathématique et la mécanique céleste. Il va imaginer une immense nébuleuse de matière, laquelle soumise aux lois de la gravitation va avoir tendance à se condenser, à partir de sa rotation initiale, pour former le système solaire dont les lois de comportement sont parfaitement connues ; la découverte de la planète Neptune, en 1846, est vécue comme un triomphe de la science positive. Chaque étoile assimilée à un soleil peut avoir son propre cortège de planètes, comme notre système solaire. D'où également, l'hypothèse déterministe de son démon infiniment intelligent dont il faudra attendre l'époque présente pour établir l'absurdité ; le *chaos déterministe* découvert par Poincaré ne sera vraiment acté que dans les années 1980.

Il faudra attendre 1986 pour entendre cet extraordinaire mea culpa du mathématicien James Lighthill, alors qu'il était président de l'Union Internationale de Mécanique Pure et Appliquée, où il dit, dans son discours à la Royal academy of sciences, *The recently recognized failure of predictability in Newtonian dynamics* :

« [...] Nous sommes très conscients, aujourd'hui, de ce que l'enthousiasme que nourrissaient nos prédécesseurs pour la réussite merveilleuse de la mécanique newtonienne les a menés à des généralisations, dans le domaine de la prédictibilité [...], que nous savons désormais fausses. Nous voulons collectivement présenter nos excuses pour avoir induit en erreur le public cultivé en répandant, à propos du déterminisme des systèmes qui satisfont aux lois newtoniennes du mouvement, des idées qui se sont, après 1960, révélées incorrectes ».

C'est une réponse claire et nette au déterminisme forcené de Laplace qui disait de façon provocante qu'il n'avait pas besoin de « l'hypothèse de Dieu » pour expliquer le « Système du Monde » où tout semble déterminé à l'avance, un monde où il n'y a plus de liberté humaine, comme celui de Baruch Spinoza. Il faut prendre acte de ces excuses en bonne et due forme, assez rare dans le monde intellectuel pour être soulignées ...

Laplace ne se ridiculise pas à la façon de Kant, il essaye d'utiliser au mieux les connaissances mathématiques sur des données qui, pour ce qui concerne la matière, sont encore quasiment celles des Grecs, comme par exemple celles de Démocrite, mais ce ne sont que des opinions. L'hypothèse atomique d'Avogadro qui date de cette époque lui est peut-être connue, mais globalement il manque de données sûres ; les spectres lumineux commencent à être répertoriés et analysés, mais ils sont globalement incompréhensibles, il faudra attendre Planck et la mécanique quantique pour commencer à y voir clair. Il extrapole, certes, mais pas à la

façon de Kant qui comblait les vides par son imagination qu'il assimilait à des données et/ou des faits.

L'hypothèse de la nébuleuse de Laplace restera populaire chez les astronomes jusqu'à l'époque de Lemaître, dans les années 1930-1940, lequel va tirer toutes les conséquences des théories de son temps : la relativité, la théorie atomique et les quanta, les nouvelles données de l'observation comme les rayons cosmiques que l'on commence à comprendre, l'étude fine du spectre des étoiles qui va permettre d'identifier un décalage vers le rouge du spectre de certaines d'entre elles, etc. C'est dans ce contexte que Lemaître va élaborer son hypothèse de l'atome primitif, qui au départ sera tournée en dérision, d'où l'appellation *Big Bang*, qui finira cependant par triompher dans le modèle cosmologique standard qui verra le jour dans les années 1980⁵. Lemaître a été un vrai génie créatif qui maîtrisait la science de son temps, mais comme il était chrétien, et prêtre catholique de surcroît, il a été ostracisé par les médiocres et les envieux, mais pas par Einstein ...

L'hypothèse de l'atome primitif de G. Lemaître

La méthode de raisonnement de Lemaître est celle prônée par Hilbert et Poincaré⁶, revisitée par von Neumann en 1954 à l'occasion d'un colloque à l'université Columbia, *The unity of knowledge*, reproduit dans le volume IV des Œuvres Complètes. À partir d'un certain niveau de cohérence, un faisceau de présomptions devient une hypothèse solide – en mathématique, c'est ce qu'on appelle une conjecture – sur laquelle l'imagination peut s'appuyer pour ne pas sombrer dans les délices de l'imagination et de la poésie.

Cette méthode de raisonnement dite hypothéticodéductive était fort peu connue, voire ignorée des philosophes à l'exception de Bergson, et sans doute aussi de Bachelard, qui ne feront pas école ; avec l'émergence de la relativité et de la mécanique quantique, les philosophes « décrochent », et même Bergson, 1^{er} Prix de mathématiques au Concours Général, dans sa jeunesse, s'y cassera les dents dans son livre *Durée et simultanéité – À propos de la théorie d'Einstein*, 1922, suite à une invitation faite à Einstein au Collège de France⁷.

Lemaître, lui, comprend parfaitement la nouvelle perspective qui s'offre. Il comprend que la force de gravitation, la seule véritablement connue de Laplace, doit être compensée par une force de répulsion dont la nouvelle théorie atomique permet de comprendre le pourquoi, avec la désintégration des atomes lourds ; deux forces cosmiques antagonistes sont à l'œuvre, et non pas une seule force, pour que l'univers existe et ne s'effondre pas sur lui-même. Il comprend également grâce à la relativité générale et à l'usage qui est fait de la géométrie non euclidienne de Riemann, comment un univers peut être fini mais sans borne, comme dans la géométrie sphérique ou elliptique ; toute chose que Laplace ignorait, ce qu'évidemment on ne peut pas lui reprocher.

D'où le retournement que va opérer Lemaître qui montre qu'il faut remplacer la nébuleuse primitive par l'atome primitif, ce qui est une thèse⁸ exactement inverse de celle de Laplace. Voici ce qu'il dit :

... On voit ainsi que, dans l'état actuel de la physique, l'état le plus simple qui puisse être compris comme point de départ d'une théorie cosmogonique n'est plus une nébuleuse plus ou moins uniforme, mais que c'est un atome unique dont la désintégration radioactive aurait donné lieu, par une cascade de décompositions successives, aux atomes plus ou moins stables qui subsistent encore aujourd'hui.

Cette hypothèse de l'atome primitif permet d'expliquer immédiatement certains faits en apparence assez surprenants. ...

⁵ Pour approfondir, voir le livre de G. Cohen-Tannoudji, M. Spiro, *Le boson et le chapeau mexicain*, 2013.

⁶ En particulier dans, *La science et l'hypothèse*, 1908, déjà cité, et ses autres écrits épistémologiques.

⁷ L'édition critique de 2019 contient un dossier complet de cette controverse intéressante.

⁸ L'expérience cruciale en faveur de cette thèse sera la découverte du fond diffus cosmologique, en 1964, qui vaudra à ses deux découvreurs Penzias et Wilson un prix Nobel en 1978.

Ce que tout cela montre, à défaut de démontrer, est que avec des hypothèses bien posées, fondées sur des observations mesurables et des mécanismes explicites qui sont des modèles logicomathématiques, on peut cerner progressivement la vérité par des raisonnements convergents, et ce faisant la faire partager à la communauté qui accepte ce jeu, un jeu dont la technologie moderne – la techno-science – va démontrer l'efficacité, une efficacité jugée « déraisonnable » par le physicien et prix Nobel Eugène Wigner, collègue et ami de von Neumann ; cette efficacité a permis de bâtir une ingénierie de l'invisible, celle des nanotechnologies, d'une part, mais surtout une ingénierie de l'information et des entités immatérielles abstraites que sont les programmes, sans laquelle nos objets quotidiens comme les smartphones n'existeraient pas.

L'erreur dans les sciences de l'information

Avec les sciences de l'information l'erreur change de statut. Jusqu'au 20^e siècle l'erreur est une préoccupation centrale pour les ingénieurs, et ce depuis au moins Galilée dans son *Discours concernant deux sciences nouvelles*, de 1638, où il jette les bases de ce qui deviendra le pilier des sciences de l'ingénieur : la résistance des matériaux, c'est-à-dire la science du POURQUOI et du COMMENT ça casse. Pour faire une voûte de cathédrale ou un dôme comme celui de Florence, il faut non seulement que la structure finale « tienne » mais que tout le processus qui conduit à son édification tienne également. La science de l'expérimentation, prélude à la science moderne, date de cette époque, avant même que celle-ci ne soit théorisée tout au long du 19^e siècle avec les débuts de la Révolution industrielle et de la science des machines « à feu » qui donnera naissance, après coup, à la thermodynamique.

L'information sans laquelle aucun de nos systèmes parfois qualifiés de « complexes » ne sauraient exister, est un produit de notre intelligence au sens *faber* et *sapiens*. Ce n'est pas quelque chose qui se trouve à l'état naturel mais quelque chose qui nécessite : a) une élaboration soignée, b) des connaissances que nous savons cumuler et intégrer, c) une architecture, et ce depuis nos plus lointains ancêtres. Pour un spécialiste de la préhistoire, un silex taillé par l'activité humaine est facilement reconnaissable dans un tas de caillou, car les formes que nous élaborons sont différentes de celles que produit la nature par les seules forces physicochimiques ; l'intention, le but recherché, le projet, transparaît toujours.

Avec l'information, l'erreur va se nicher dans les représentations, c'est-à-dire au départ dans les langages, et dans l'organisation de ces représentations. L'information, pour exister, se transformer et se transmettre, a besoin d'un substrat physicochimique, donc de l'énergie, pour constituer le support matériel du message porteur de l'information, lequel est soumis aux lois générales de la résistance des matériaux, qui s'appliquent également aux processus de transformation dans les équipements appelés transducteurs.

D'où l'injonction de Richard W. Hamming, l'un des grands contributeurs de la théorie des codes et des sciences de l'information, dans son livre *Coding and Information Theory*, 1985 :

“Most bodies of knowledge give errors a secondary role, and recognize their existence only in the later stages of design. Both coding and information theory, however, give a central role to errors (noise) and are therefore of special interest, since in real-life noise is everywhere”,

que l'on peut traduire par :

« La plupart des domaines de connaissances donnent aux erreurs un rôle secondaire, et reconnaissent leur existence seulement dans les dernières étapes de la conception. La théorie des codes et la théorie de l'information donnent toutes deux une place centrale aux erreurs (bruit) et pour cette raison présentent un intérêt particulier, car dans la vie réelle le bruit est partout »,

on ne saurait mieux dire ! L'ingénieur des *Bell Labs* et du MIT, Claude Shannon, dans les années 1940, avec l'aide de John von Neumann et Norbert Wiener, démontre que l'on peut construire des codes correcteurs d'erreurs, c'est-à-dire des codes qui corrigent les fluctuations

du « bruit » dans la mesure où elles sont aléatoires – c'est le 2^{ème} théorème de Shannon – et Hamming construira effectivement de tels codes, dits *Codes de Hamming*, avec d'autres. Une nouvelle perspective s'ouvre qui ne se développera pleinement qu'avec les 1^{ères} machines entièrement transistorisées dans la décennie 1980.

L'ordinateur, dans l'architecture dite de von Neumann, est la 1^{ère} machine qui intègre complètement dans son intimité la problématique des erreurs posée par la théorie de l'information, grâce à des dispositifs dont von Neumann fera la théorie. Dans cette architecture⁹, il va introduire une distinction fondamentale entre ce qu'il appelle le LANGAGE INTERNE, un langage spécifique des structures organiques de la machine, et le LANGAGE EXTERNE, spécifique du programmeur humain où la visibilité des structures organiques est remplacée par une bibliothèque de fonctions qui en donne une image abstraite purement logique, et plus simple, pouvant comporter des centaines de fonctions appelées *API/Application Programming Interface* dans le jargon. Le programmeur communique les ordres à effectuer, ordres que la machine exécutera mécaniquement, sans fatigue, à une vitesse de plus en plus grande, 20.000 ordres à la seconde dans les années 1950, plusieurs centaines de milliards aujourd'hui sur des machines comme la NVIDIA GV100.

Voir les planches N°6, 7, 8 et 9 dans l'annexe

Depuis les années 1990, la majorité des programmeurs travaille avec l'interface du LANGAGE EXTERNE, c'est-à-dire des langages comme FORTRAN, COBOL dans les années 1950-1960, puis des langages comme C, C++, Java, Python, PHP, SQL, ... et les nombreux langages associés à Internet comme XML, JavaScript, etc. A charge de programmes spéciaux appelé COMPILATEURS, intégrés au système d'exploitation, d'établir la traduction LANGAGE EXTERNE → LANGAGE INTERNE, soit de façon statique, soit de façon dynamique, et ce de façon réversible. Les erreurs de programmation concernent le LANGAGE EXTERNE, mais leur découverte par la machine et/ou lors des tests de validation se fait sur le LANGAGE INTERNE ; il faut donc que la machine et le système d'exploitation donne la traçabilité inverse pour que les programmeurs puissent corriger leurs erreurs. On peut facilement comprendre que ce double mécanisme est crucial pour la productivité des programmeurs et pour le contrat de service vis-à-vis des usagers externes, car sans eux, toutes ces fantastiques machines seraient virtuellement inprogrammables, car beaucoup trop compliquées.

L'activité de programmation se déroule au sein d'équipes qui peuvent regrouper plusieurs centaines de programmeurs, ce qui fait que, outre l'aspect individuel de l'acte, les aspects collectif et social doivent être impérativement pris en compte. Les programmeurs communiquent beaucoup entre eux pour négocier les API spécifiques de leurs programmes, et ils communiquent avec toutes les parties prenantes impliquées dans le système qu'il réalise. Tous les biais cognitifs propres à la communication humaine sont donc présents dans l'activité de programmation, d'où les taux d'erreurs observés après livraison des systèmes, et ce en dépit des phases de vérification/validation qui accompagnent tout le cycle de développement des systèmes.

Voir les planches N°10, 11, 12 et 17 dans l'annexe

Les taux d'erreurs individuelles propres à l'activité psychocognitive¹⁰ – cf. tableau ci-dessous – sont statistiquement bien connus, depuis les premières réalisations de systèmes complexes dans les années 1940-1950 car c'est un paramètre ergonomique fondamental. Ils se situent aux alentours de 5 à 10 erreurs par heure d'activité. L'analyse du produit de cette activité pour ce qui concerne les programmeurs conduit à identifier trois types de textes aujourd'hui normalisés :

⁹ Pour le détail, voir le livre *Survivrons-nous à la technologie ?*, à l'occasion de mes conférences à l'UC Lille en 2019.

¹⁰ Nombreux ouvrages sur le sujet, voir https://en.wikipedia.org/wiki/Cognitive_psychology, et le *Handbook of Human Factors and Ergonomics*, Gavriel Salvendy, toujours utilisé.

1. Un texte qui définit ce que le système DOIT FAIRE, en prenant le point de vue des usagers et/ou utilisateurs, et d'une façon plus générale de toutes les parties prenantes. Ce texte purement intentionnel est écrit dans le langage des usagers ; il peut comporter des études de cas pour illustrer le besoin. Il doit être le plus concis possible pour pouvoir être partagé par le maximum de personnes.
2. Un texte qui définit le plan général de construction du système, une nomenclature précise des API spécifiques, les contraintes techniques à prendre en compte, taille, consommation d'énergie, etc. Ce texte est le référentiel du COMMENT FAIRE qui doit être partagé par tous les acteurs de l'ingénierie. Il doit être non ambiguë, et de ce fait écrit dans des langages formels et/ou semi-formels qui serviront à constituer les dossiers de programmation remis aux programmeurs.
3. Un texte final PROGRAMME–TEST qui est la programmation proprement dite, texte formel destiné à être exécuté sur les machines constitutives du système, texte écrit dans le ou les langages de programmation choisis par les parties prenantes et leurs API associées, texte qui servira à valider que le système ainsi construit satisfait aux besoins formulés dans le texte N°1 ; d'où un texte complémentaire indispensable qui est une « preuve » de bon fonctionnement. La programmation a donc un double aspect : un aspect affirmatif et/ou impératif, et un aspect démonstratif, tout à fait analogue à un théorème et sa démonstration, la démonstration pouvant être beaucoup plus longue que l'énonciation proprement dite¹¹.

Voir la planche N°10 dans l'annexe

Chacun de ces textes contient des erreurs, mais le texte final doit en contenir un nombre minimal permettant de satisfaire le contrat de service négocié avec les usagers. On peut concevoir que ce nombre va pouvoir varier entre des logiciels de contrôle/commande en avionique ou pour le nucléaire, et des systèmes d'information grand public comme les logiciels de paye, de comptabilité ou pour le recouvrement des impôts.

Ce nombre s'exprime par une mesure explicite appelée disponibilité du système qui est un rapport entre des durées de vie en bon fonctionnement et des durée de pannes, c'est-à-dire d'indisponibilité, où le système ne rend plus le service attendu ; toutes ces mesures sont statistiques.

→ La **programmation** apparaît pour ce qu'elle est véritablement : **une chasse aux erreurs** qui peut consommer jusqu'à 80% de l'énergie humaine investie dans le projet de développement.

La programmation : nouvelle « matière », nouvel outil ?

Rarement dans l'histoire humaine, sauf peut-être dans l'histoire des techniques, nous avons su dater avec précision les points de bascules qui caractérisent un changement d'époque – une transition de phase, comme disent les physiciens – où l'après n'est pas réductible à l'avant. Un événement apparaît ... et tout change.

La programmation naît dans ce creuset de la modernité que fut le projet Manhattan, où l'élite scientifique occidentale est mobilisée contre les totalitarismes, premier grand projet à l'échelle planétaire qui servira de modèle à tous ses successeurs qui seront les vecteurs de la mondialisation.

Avant 1940, rien ne ressemble ni de près ni de loin à ce qu'on appellera « la programmation » des systèmes ; le métier de programmeur n'existe tout simplement pas, mais en trois ou quatre générations il va submerger la planète au point de ne plus savoir, aujourd'hui, dire comment c'était « avant ».

La programmation intègre trois types d'activités et/ou domaines de connaissances distincts mais indissociables :

¹¹ La démonstration du grand théorème de Fermat – une ligne – reste emblématique avec sa preuve qui fait plusieurs centaines de pages.

1. Une machine entièrement nouvelle par sa complication/complexité intitulée *Computing instrument* – par von Neumann – ou plus simplement *Computer* – un vieux mots français fréquent dans les écrits alchimiques, passé à l'anglais –, et en français : ordinateur ; la machine est caractérisée par son LANGAGE INTERNE, qui définit son jeu d'instructions basiques : plusieurs centaines dans les machines d'aujourd'hui.
2. Un « humain qui calcule » selon l'expression de Wittgenstein, c'est-à-dire celui que nous appelons depuis les années 1960 un programmeur, dont le rôle est de fabriquer un programme qui va produire un effet dans le monde réel à partir du jeu d'instructions et des organes périphériques de la machine.
3. Une interface entre la machine et le programmeur qui prend en compte les contraintes psychocognitives des programmeurs que von Neumann appellera LANGAGE EXTERNE, c'est-à-dire tous nos langages de programmation et leurs bibliothèques de fonctions intégrées.

Il n'y a pas de relation de préséance entre ces trois « choses » qui doivent être comprises comme complémentaires mais non réductibles l'une à l'autre, analogue à ce phénomène étrange révélé par la physique de l'atome que la mécanique quantique qui vient de naître appellera relation de dualité, comme la dualité onde-particule : pas d'ordinateur sans programmeur, pas de programmeur sans la machine et son langage interne, et sans surtout de langages externes sans lesquels la programmation n'aurait jamais pu prendre l'ampleur que nous lui connaissons.

En quelques décennies la programmation est devenue un bien commun comme l'air qu'on respire, mais les circonstances de sa création restent spécifiques : sans programmeur, sans cet « humain qui calcule », sans rigueur logique, il n'y a pas de programmation ; c'est un peu comme l'oxygène de l'air qui est le produit de la vie végétale sur notre terre.

La programmation apparaît comme une nouvelle matière dans le droit fil des savants grecs qui se demandait : « pourquoi y a-t-il quelque chose plutôt que rien ? » ; reformulée en : pourquoi y a-t-il de la programmation ?, car après tout nous avons vécu sans pendant les siècles qui l'ont précédée, et ce dans toute les grandes zones culturelles. Pourquoi est-elle née en Occident ?, si tant est que la question ait un sens.

La programmation naît d'un besoin de calcul suscité par les problèmes que se posent les sociétés humaines pour leur survie. En ce sens la programmation apparaît comme un méta outil que l'« humain qui calcule » va façonner pour résoudre les problèmes qui se posent. La programmation est toujours intentionnelle, elle a un but, une finalité qui est la résolution de tel ou tel problème.

Les problèmes d'organisation immenses que soulève la deuxième guerre mondiale aux démocraties en lutte à mort contre les totalitarismes vont très rapidement venir buter sur les capacités de calculs disponibles de l'époque. Telle est la situation face à laquelle Turing va devoir se confronter pour décrypter les messages de la machine de codage ENIGMA de l'armée allemande. Tel est le cas des physiciens et ingénieurs de Los Alamos confrontés aux volumes des calculs nécessaires aux modélisations des premières bombes atomiques. Tel est également le cas des organisateurs et planificateurs du débarquement de Normandie le 6 juin 1944 pour anéantir la puissance militaire et industrielle de l'Allemagne sur le front Ouest.

Résoudre les problèmes qui se posent est une activité que les savants européens connaissent bien. On peut même dire qu'ils sont comme fascinés par ce désir de dominer la nature, ou du moins de faire jeu égal avec elle, car dans le monde chrétien qui est le leur ils se sentent co-créateurs de cette même nature. Dans leur imaginaire, c'est une façon d'honorer le créateur de toute chose, selon les termes du Credo, avant de se retourner contre lui au 19^e siècle. Dans la perspective du Monde 3, celui de la connaissance objective selon les termes de Karl Popper, que rajoute la programmation ?

Jusqu'à l'apparition des premiers ordinateurs, l'effet moteur de cette connaissance accumulée est un être humain qui fait l'effort d'acquérir tout ou partie de cette connaissance calculatoire selon l'utilité qu'il peut en avoir, mais cet effort est un éternel recommencement ; chaque génération doit prendre le relai de celles qui l'ont précédée, en l'améliorant si nécessaire. Avec l'ordinateur, l'« humain qui calcule » va se créer un auxiliaire de vie universel car les connaissances organisées en vue d'un calcul sur une machine vont pouvoir être réutilisées par d'autres, autant de fois que nécessaire, et ce à coût marginal. Il suffit pour cela de disposer de machines en nombre suffisant, et de systèmes d'exploitation qui garantissent la portabilité des logiciels, lesquels deviennent alors un bien commun, mais aussi un élément de puissance qui va engendrer des rivalités.

Reste les imperfections que peuvent contenir les logiciels, c'est-à-dire les connaissances qui les ont fait naître, car si le cerveau humain, grâce à son bon sens est tolérant aux erreurs, c'est exactement l'inverse pour les machines qui, elles, sont intolérantes aux erreurs. Nos connaissances sont incomplètes, approximatives, souvent voire toujours fausses car la « vérité » que nous attribuons aux phénomènes dépend presque toujours de l'échelle de l'observation¹². Ce qui paraît vrai à notre échelle peut s'avérer faux à l'échelle atomique ou à l'échelle cosmique.

L'activité psychocognitive qui est sous-jacente à l'activité de programmation fait que la « matière » informationnelle ainsi créée contient des défauts tant au niveau déclaratif qu'au niveau procédural, ce qui fait que leur mise sous tension dans les machines peut occasionner toutes sortes de défaillances dont certaines mortelles pour les usagers. Pour espérer les contrôler, il faut donc d'abord les analyser en détail.

Quelques causes de défaillances du processus psychocognitif

Le processus de fabrication des textes N°2 et 3 nécessite une maîtrise complète des langages externes et de l'environnement de développement, bibliothèques d'API, outillages, etc. L'apprentissage nécessite une formation de base de niveau Bac +5 à +8, à laquelle il faudra venir ajouter la durée nécessaire à la pleine connaissance des bibliothèques qui totalisent aujourd'hui des centaines de fonctions, à raison de 10-15 heures de travail par fonction en moyenne. Un programmeur expérimenté, 3 à 5 ans de pratiques, peut espérer maîtriser quelques centaines de fonctions ; par exemple 500 fonctions nécessitent au minimum 5.000 heures de travail soit un équivalent temps plein de 3 années. Les occasions de se tromper sont multiples et variées. D'où une productivité constatée de 2 à 3 lignes par heure travaillée, en moyenne, mais 1 ligne/heure pour les logiciels critiques.

Cette liste, utilisée dans mes ouvrages antérieurs n'est pas exhaustive ; elle est cependant suffisamment détaillée pour apprécier l'ampleur du phénomène des causes d'erreurs.

D1 : Erreur de perception, manque de discrimination, confusion entre sémantique et syntaxe.

D2 : Erreur de codage/décodage.

D3 : Erreur de représentation et/ou symbolique non adaptée (interopérabilité entre systèmes).

D4 : Représentation, compréhension et interprétation erronées de phénomènes dynamiques et/ou combinatoires (plusieurs flots d'exécution en parallèle qui interfèrent, perception du temps vrai versus temps psychologique, synchronisation) ; modèle mental erroné du phénomène.

D5 : Impasse, non-exhaustivité, oubli pur et simple.

D6 : Illusions (dans notre cas ce sont surtout les paradoxes de type logique, confusion des niveaux d'abstraction ; non prédicativité).

D7 : Acceptation comme vraie d'une hypothèse fausse.

D8 : Acceptation comme fausse d'une hypothèse vraie.

D9 : Attribution de propriétés inutiles et/ou erronées.

¹² Sur cet aspect important, voir le livre de l'informaticien Leslie Valiant, Turing Award 2010, *Probably approximately correct*.

- D10 : Hypothèse superflue et/ou non appropriée (Exp. : tel événement se produit rarement alors qu'il est fréquent).
- D11 : Erreur de communication entre les acteurs humains, de traduction lors d'un changement de code (contre sens, faux sens, etc.).
- D12 : Non respect d'une procédure ou d'une règle.
- D13 : Non prise de décision en temps voulu (logiques temporelles, logiques capacitaires)
- D14 : Action non adaptée au contexte, action contradictoire et/ou antinomique vis à vis d'autres actions.
- D15 : Itération, répétition inutile d'une action (i.e. propriété d'idempotence des actions).
- D16 : Absence d'information qui entraîne une action par défaut non adaptée (non perception d'un manque ou d'une absence de quelque chose).
- D17 : Erreur de raisonnement, raisonnement circulaire, généralisation abusive.
- D18 : Défaut ou excès de généralité, abstraction mal construite, définition ambiguë (non prédicative).
- D19 : Confusion langage / métalangage, concept / méta-concept (mélange de types au sens logique, ou équations de dimensions incohérentes en physique).
- D20 : Saturation de la bande passante cognitive (Exp. : trop de décisions simultanées, interruptions continues, excès de parallélisme) d'où perturbation de l'attention.
- D21 : Saturation de la mémoire de travail.
- D22 : Analogies, associations erronées et/ou inadaptées au contexte, corrélations fallacieuses.
- D23 : Confusion/Interférence proactive et/ou rétroactive.
- D24 : Changement de tâches fréquent, « papillonnage », toute perturbation qui augmente la probabilité de confusion et d'interférence entre les tâches.
- D25 : Complexité mal maîtrisée, violation de la règle du « Nombre magique 7 ± 2 » limitant le nombre de situations élémentaires que le cerveau conscient peut appréhender, non linéarité perçue comme linéaire.

Vivre et survivre avec les erreurs

L'exécution d'un défaut va toujours occasionner une défaillance plus ou moins grave compte tenu du contexte, défaillance qualifiée familièrement de *Bug* par les programmeurs, lequel peut se manifester de deux façons caractéristiques :

1. Soit il est répétable, et le programmeur sait le reproduire, ce qui permettra de l'analyser et de le corriger, soit immédiatement, soit de façon différée en phase de maintenance du programme ;
2. Soit il est non reproductible compte tenu des circonstances combinatoires trop complexes qui ont occasionné sa survenue, et le programmeur se contentera de le répertorier soigneusement.

L'informaticien Jim Gray, Turing Award 1998, appelait les premiers *Bohrbugs*, en souvenir de l'atome de Bohr assimilé à un petit système planétaire régi par les lois déterministes de Newton, et les second *Heisenbugs*, car en conformité avec l'indéterminisme quantique modélisé par les relations d'incertitude du physicien Heisenberg.

Il y a une « magie » de l'aléatoire bien mise en évidence et théorisée par von Neumann, car si un phénomène est véritablement aléatoire, alors il ne se répétera jamais identiquement à lui-même, par définition. C'est donc une propriété intéressante que l'on va pouvoir exploiter dans l'ingénierie des systèmes où ces phénomènes apparaissent. Ainsi fonctionnent les Codes Correcteurs d'Erreurs, sans lesquels nos machines ne fonctionneraient pas plus de quelques minutes, et les systèmes dits parallèles qui exécutent des milliers de transactions en concurrence comme si elles étaient seules. Pour que tout cela « marche », il faut quand même

détecter les défaillances lors de leur survenue de façon à interrompre l'exécution et rejouer la séquence qui a conduit à la défaillance, le contexte étant alors différent¹³.

Le but ultime de la programmation est donc a) d'éliminer progressivement tous les Bohrbugs des programmes, et b) de monitorer les transactions qui peuvent conduire à des Heisenbugs de façon à pouvoir les rejouer en cas de problème. Pour que ce but soit atteint, il faut que tout ce qui se passe dans la machine soit surveillé par des dispositifs ad hoc intégrés au système d'exploitation pour détecter toute situation jugée anormale et rejouer la séquence si nécessaire. Ces dispositifs sont présents dans tous les systèmes quoique invisibles pour les programmeurs d'application « externes ».

Le taux d'erreurs résiduelles apparaît alors comme une bonne mesure de la qualité de la programmation et du niveau de maturité et/ou de rigueur de l'équipe qui a réalisé le programme. Depuis les années 1970 et les premiers programmes spatiaux comme Apollo, la NASA a proposé de rapporter ce taux à la taille des logiciels livrés pour accomplir la mission, une mesure aujourd'hui normalisée par le SEI¹⁴ en usage dans toute la profession.

Un taux de 0,1 erreur par millier de lignes de code est considéré comme excellent, ce qui dans les normes de l'édition correspondrait à une erreur toutes les 200 pages en moyenne. C'est le taux exigé pour les logiciels dits « critiques » car leur défaillance fait courir un risque jugé insupportable socialement ; c'est le cas en avionique, dans le nucléaire et d'une façon générale dans les industries dites à risques.

Un taux de 1 erreur par millier de lignes de code est considéré comme acceptable dans de nombreuses situations de la vie courante mais force est de constater que de nombreux systèmes sont bien au-delà de ce taux, ce qui peut conduire à des situations inacceptables pouvant occasionner le retrait de service comme ce fut le cas pour le système LOUVOIS¹⁵. Le cas récent le plus emblématique est celui des logiciels de pilotage des Boeing 737 MAX qui ont occasionné deux crashes tuant 346 personnes, cas d'une extrême gravité car Boeing a cru bon de manipuler et corrompre les autorités de certification, en l'occurrence la FAA¹⁶, et les normes en vigueur dans la profession comme la DO-178C, *Software Considerations in Airborne Systems and Equipment Certification*, qui fixe les règles à suivre pour la programmation de ces systèmes.

Voir les planches N°13 et 14 dans l'annexe

Ce qu'il faut dire cependant à propos de ces graves manquements est que, chaque fois, les dirigeants des équipes de développement de ces systèmes ont violé les règles de prudence, parfois sciemment, qui sont à la base de l'ingénierie, règles qui sont issues des savoir-faire et des leçons retenues des accidents du passé. Les normes issues de ces expériences cumulées ne sont pas une garantie en soi, car il ne faut jamais abdiquer son bon sens, ni son intelligence, mais chaque fois qu'on les viole la catastrophe est au rendez-vous. Il appartient aux pouvoirs publics de les faire respecter, voir de donner l'exemple dans leur respect, ce qui est malheureusement loin d'être le cas.

Des normes comme la DO-178 ou d'autres plus basiques comme ISO/CEI-15288¹⁷ sur l'ingénierie ou ISO/CEI-15408 sur la sécurité, garantissent que le processus projet qui conduit à l'élaboration des systèmes est contrôlable et auditable. La maturité des équipes projets, tant au plan individuel qu'au plan collectif, en charge de ces systèmes a également fait l'objet de

¹³ Ces mécanismes ont été présentés dans mon livre *Survivrons nous à la technologie ?*, déjà cité. Le style de programmation sous-jacent est dit « transactionnel » dont Jim Gray fut un ardent promoteur, dès les années 1980, avec l'émergence des systèmes distribués.

¹⁴ Voir <https://www.sei.cmu.edu/>

¹⁵ Voir https://fr.wikipedia.org/wiki/Logiciel_unique_à_vocation_interarmées_de_la_solde

¹⁶ Voir <https://www.faa.gov/>

¹⁷ Voir https://en.wikipedia.org/wiki/ISO/IEC_15288 ; une synthèse de plus de 50 ans de savoir-faire en ingénierie système.

nombreuses études orchestrées par le SEI¹⁸, un processus lent car il concerne les organisations de développement dans leur globalité qui ne peut s'acquérir qu'en réalisant des systèmes.

Les systèmes dits « intelligents » dont il est surtout question dans ce séminaire n'échappent pas à ces règles, et on pourrait même dire a fortiori car imitant d'une certaine façon l'intelligence humaine sans en avoir les capacités, ils peuvent induire chez les usagers naïfs une impression trompeuse de sécurité. Comme toute l'informatique, ce ne sont que des outils au service de ceux qui les ont conçus et non des substituts à notre libre arbitre, mais dans tous les cas de figures ils doivent créer de la confiance et non de la défiance, ce qui reste un défi pédagogique permanent, comme on le voit avec la voiture autonome, et comme on l'a vu avec le nucléaire qui est passé du statut d'énergie bénéfique à l'humanité à celui d'énergie maudite et dangereuse, les bombes atomiques y étant pour quelque chose.

Ce qui au départ de cette transition ne concernait que quelques milliers de programmeurs systèmes, concerne aujourd'hui des millions de programmeurs répartis sur toute la planète. Là résulte la difficulté principale, et peut-être même une limitation qui reste à découvrir, car il s'agit d'un phénomène collectif où la défaillance d'un seul peut ruiner le travail de tous. Les processus humains sous-jacents, le sens des responsabilités tant au niveau individuel qu'au niveau collectif, ce sens du « tout est lié », imposent une éthique collective qui reste difficile à acquérir. Une discipline librement acceptée est indispensable comme le reflètent certains ouvrages qui ont structuré la profession comme *A discipline of programming*, 1976, de E. Dijkstra, ou encore *A discipline for software engineering*, 1994, de W. Humphrey, pour n'en citer que deux. Le « *taedium vitae* », cette fatigue de vivre et de faire des efforts, dont on commence à sentir les effets délétères en Occident qui semble noyé sous la complexité qu'il a lui-même engendré avec une mondialisation qui apparaît comme incontrôlable, sur un fond de défiance généralisée. A défaut de tout comprendre, l'utilisateur doit faire confiance aux experts qui en retour ont l'impérieux devoir d'expliquer ce qu'ils font en toute transparence. Cette boucle systémique vertueuse usagers utilisateurs ↔ ingénieurs concepteurs est un bien commun dans un monde où « tout est lié ».

Une pédagogie de la complexité comme la réclame la juriste Mireille Delmas-Marty¹⁹ devient une tâche urgente si l'on veut survivre, une pédagogie dans laquelle les erreurs joueront sans nul doute un rôle de premier plan dans un référentiel inspiré de celui de l'ingénierie système.

Voir les planches N°15, 16 et 17 dans l'annexe

¹⁸ Voir https://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration

¹⁹ Voir son livre *Sortir du pot au noir – L'humanisme juridique comme boussole*.