

**Mémoire déposé à la Commission d'enquête
sur la gestion de la modernisation des
systèmes informatiques de la Société de
l'assurance automobile du Québec**

Louis Martin Martin Cloutier

Université du Québec à Montréal

Version du 2 octobre 2025

Table des matières

Liste des abréviations, sigles et acronymes	ii
1. Introduction	1
2. Carence de connaissances numériques des administrateurs	3
3. Obsolescence des systèmes patrimoniaux	4
4. Estimation initiale	6
5. Mirage des <i>PGI</i>	8
6. Cout total de possession	10
7. Agile	11
8. Productivité des développeurs	14
9. Conclusion	17
A. Rompre avec l'illettrisme numérique — Le Devoir	19
Références et bibliographie	22

Liste des abréviations, sigles et acronymes

CASA Carrefour des services d'affaires

CESIS Commission d'enquête sur la gestion de la modernisation des systèmes informatiques de la Société de l'assurance automobile du Québec

COBOL COmmon Business-Oriented Language

CTP Cout total de possession — en anglais **TCO** *Total Cost of ownership*

DB2 IBM DB2 database Software

IPA Interface de Programmation d'Application — en anglais **API** *Application Programming Interface*

IDMS Integrated Database Management System

LMETA Loi fédérale sur l'utilisation de moyens électroniques pour l'exécution des tâches des autorités (Confédération suisse) — en allemand **EMBAG** *Bundesgesetz über den Einsatz elektronischer Mittel zur Erfüllung von Behördenaufgaben*

PGI Progiciel de gestion intégré

RICEFW Reports, Interface, Conversion, Enhancement, Form, and Workflow — les *RICEFWs* permettent d'adapter le *PGI* de *SAP* aux besoins spécifiques d'une organisation

SAAQ Société de l'assurance automobile du Québec

SAAQcllic Services en ligne aux particuliers et aux entreprises (plateforme transactionnelle)

SI Système d'information

TIC Technologies de l'information et de la communication

XP eXtreme Programming

1. Introduction

Le présent mémoire se concentre, d'un point de vue strictement informatique, sur les principaux problèmes ayant contribué au résultat actuel observé et de ce qui a été mis au jour à propos du projet *CASA* à la *SAAQ*. Plusieurs considérations, comme la gouvernance, les processus de prise de décision, etc., dépassent le cadre du présent mémoire qui focalise plutôt, pour l'essentiel, sur quelques réflexions au sujet de l'aspect informatique du projet *CASA*.

Le cas du projet *CASA* est loin d'être unique. Plusieurs projets informatiques entrepris par des organismes gouvernementaux et paragouvernementaux ailleurs au pays et dans le monde ont connu leur lot de délais et de dépassements importants de couts. Pour éviter la répétition de dérives de projets informatiques dans le secteur public, certains travaux de commissions d'enquête nationales ont été entrepris dans plusieurs pays, dont les résultats méritent d'être étudiés. Certains de ces projets ont fait l'objet de rapports de la part de différents types de commissions, quelques-uns de ces rapports sont mentionnés dans la section *Références* du présent document. À cette fin, nous recommandons vivement la lecture du document portant sur les *Conclusions and recommendations of the Dutch temporary committee on government ICT projects* (Representatives of the Netherlands, 2014) résultat d'un comité d'enquête parlementaire sur les échecs des projets informatiques aux Pays-Bas. Dans son rapport, ce comité aborde les principaux facteurs affectant les projets du domaine des *TIC* au niveau gouvernemental : la complexité politique, la complexité organisationnelle et la complexité technologique.

Basé sur les informations obtenues lors de la diffusion des séances de la *CESIS*, ce mémoire expose nos commentaires sur le projet *CASA*. Nous avons puisé dans les ouvrages de plusieurs praticiens reconnus dans le domaine des *TIC* et dans notre expérience de plus de cinquante ans pour apporter un point de vue qui, nous l'espérons, pourra aider.

Dans ce mémoire, nous nous concentrons brièvement sur les points suivants :

- la carence de connaissances numériques des administrateurs ;
- l'obsolescence appréhendée des systèmes patrimoniaux ;
- la difficulté d'établir une estimation initiale ;
- la pertinence du choix d'un *PGI* ;
- le cout total de possession ;
- le choix de l'approche *Agile* dans le contexte du projet *CASA* ;
- la productivité des développeurs.

La mise en place d'un nouveau *SI* est une initiative risquée. Si elle réussit, il y a peu à gagner pour ceux responsables de le réaliser, si elle connaît des problèmes, les conséquences peuvent être sans commune mesure avec les bénéfices escomptés. La traduction de ce principe est ce que Nassim Nicholas Taleb nomme un *cygne noir* (Taleb, 2020). En effet, dans une matrice de comparaison des bénéfices et des risques, ces initiatives *SI* se situent dans une zone de « risques élevés » et « bénéfices nuls ou hautement incertains ». En outre, ces initiatives sont à la confluence de nombreux risques rares et imprévisibles, et dont les effets dominos engendrés sont majeurs. Ces situations exigent donc un examen minutieux et hautement approfondi des dimensions informatiques, car les circonstances qui engendrent ces risques sont de surcroît imprévisibles dans leur évolution et doivent faire l'objet de contrôles et de mitigations. Il faut aussi retenir que les *SI* modernes sont constitués de plusieurs millions de lignes de code, devant satisfaire aux besoins de nombreuses parties prenantes ayant des exigences souvent contradictoires. Le domaine de l'informatique est jeune de seulement quelques décennies. Il connaît des changements technologiques extrêmement rapides bousculant les manières de faire dans les organisations. Tous ces éléments apportent leur lot de défis méritant une réflexion sur la situation actuelle mise en évidence par le projet *CASA* pour en tirer les meilleures leçons possibles.

En conclusion du présent mémoire, certaines pistes à explorer, susceptibles d'aider pour les projets de remplacement ou de développement de *SI* dans les organismes publics et parapublics, seront mentionnées.

2. Carence de connaissances numériques des administrateurs

Au royaume des aveugles, les borgnes
sont rois.

Proverbe populaire

Les témoignages de plusieurs administrateurs entendus lors des audiences de la *Commission* sont éloquentes à ce propos. Les administrateurs ne se considéraient pas compétents pour remettre en question les recommandations — par exemple, sur le choix d'un système ou d'un fournisseur — qui leur étaient présentées. Cet état de fait laisse présager un pouvoir excessif aux personnes perçues comme spécialistes du domaine avec les conséquences dont nous sommes témoins.

L'excellent éditorial *Rompre avec l'illettrisme numérique* paru dans *Le Devoir*¹ du 17 mai dernier décrit très bien la situation (Rioux Souci, 2025) :

... Si nous en sommes encore à piétiner sur ce front, c'est aussi parce que l'État québécois accuse un vrai retard en matière de littératie numérique. Le commissaire Gallant a remué le couteau dans la plaie cette semaine en comparant le travail des administrateurs de la SAAQ à un « acte de foi » envers les personnes chargées de présenter le projet. ...

... L'illettrisme numérique généralisé de ses décideurs et de ses administrateurs nuit grandement au Québec. Son fort penchant pour la sous-traitance aussi, car ce dernier freine le développement des expertises locales en plus de lui coûter fort cher. ...

Deux principaux constats peuvent être dégagés en matière de carence de connaissances numériques de l'État québécois :

Constat 1 — La majorité des administrateurs ayant à prendre des décisions relatives aux *SI* n'ont pas les connaissances requises.

Constat 2 — Les *SI* dans les organismes publics et parapublics jouent un rôle de plus en plus essentiel pour accomplir leur mission.

1. Une copie de l'éditorial est incluse en annexe.

3. Obsolescence des systèmes patrimoniaux

Une raison souvent évoquée pour justifier un projet *SI* d'envergure est d'alerter la haute direction d'une organisation sur l'obsolescence technologique des systèmes critiques existants. Les partisans du changement proposé ont souvent déjà une solution en tête. Il est fréquent que les problèmes pressentis soient exprimés sous la forme de solutions déjà identifiées, ce qui semble être le cas dans le projet *CASA*.

Comme noté dans un écrit portant sur les projets d'envergure dans le secteur public :

The paradox of the unexplored opportunity space : the choice of conceptual solution is made without systematically scrutinising the opportunity space up front. There is much evidence to suggest that in many cases the chosen concept is not necessarily the most effective solution to the initiating problem. In many cases, the process started out with a predetermined solution, without exploring other options. This is referred to as path dependency.

— Tiré du livre *The Front-end of Large Public Projects*¹ (Volden et Samset, 2023, p. 3)

Cette tendance à explorer de façon privilégiée une seule solution se fait souvent au détriment d'autres approches moins couteuses, moins risquées, etc.

Douze ans se sont écoulés depuis que l'alarme a été lancée concernant l'obsolescence des systèmes patrimoniaux à la *SAAQ*. Or, il est utile de noter que plusieurs de ces systèmes sont encore en fonction et le seront encore pour plusieurs années à venir.

Concernant le langage *COBOL*, la base de données *IDMS*, les progiciels *DB2*, ces technologies sont encore utilisées. Un point délicat à considérer est de s'assurer qu'un nombre suffisant de ressources connaissent les technologies en question. Des plans de formation peuvent être requis pour s'en assurer. Dans l'index *TIOBE*² classifiant les langages de programmation, *COBOL* se situe encore dans les 25 langages les plus utilisés sur les 100 langages répertoriés. De nombreuses organisations utilisent encore *COBOL* pour gérer des applications critiques, principalement dans les secteurs bancaires, gouvernementaux et de l'assurance.

1. Le livre est disponible en sous-licence libre sur le site <https://library.oapen.org/handle/20.500.12657/55789>.

2. <https://www.tiobe.com/tiobe-index/>

L'obsolescence des systèmes patrimoniaux est donc souvent plus un prétexte que le résultat d'une analyse rigoureuse fondée sur des faits. L'argument du manque de personnel ayant des connaissances sur ces technologies peut être mitigé par des formations. De plus, l'argument est fallacieux. D'une part, lors des témoignages de la *Commission*, on a cité le manque de ressources connaissant le langage de programmation *COBOL* pour justifier le changement de système, et d'autre part, on a noté la difficulté d'accès aux ressources requises pour réaliser la nouvelle plateforme du projet *CASA*. Le problème de l'accès à des ressources qualifiées semble demeurer entier.

Deux constats émergent en ce qui concerne l'obsolescence des systèmes patrimoniaux et leur remplacement :

Constat 3 — L'utilisation d'un *PGI* semble avoir été la seule alternative étudiée de façon plus poussée.

Constat 4 — Le langage *COBOL* est encore utilisé par de nombreuses organisations pour des applications critiques à leur mission.

4. Estimation initiale

The requirements will not be completely known until after the users have used the finished product.

Watts Humphrey

La compréhension de ce qu'est la conception et la mise en place d'un *SI* est un point important. Il est impossible de déterminer un budget et un échéancier précis au début d'un projet pour un *SI* d'une certaine envergure. Souvent, les besoins exprimés sont incomplets et varieront au cours du projet. La productivité de l'équipe du projet n'est pas connue, l'équipe n'étant pas constituée et n'ayant pas atteint sa vitesse de croisière. La courbe en entonnoir de Boehm (1984, p. 8), souvent utilisée pour identifier les sources d'incertitude dans le développement de logiciels, et subséquemment renommée et diffusée sous l'expression *cône d'incertitude* par McConnell (1998, p. 31), illustre ce point (voir la figure 4.1). Les règles actuelles dictées par le gouvernement exigent un échéancier et un budget précis avant d'entreprendre un tel projet. Cette exigence conduit à des dépassements conséquents. Les personnes dédiées au projet sont très au fait que l'estimation initiale repose sur des hypothèses fragmentaires et ne se sentent pas contraintes par celui-ci.

For every 25 percent increase in problem complexity, there is a 100 percent increase in complexity of the software solution. That's not a condition to try to change (even though reducing complexity is always a desirable thing to do); that's just the way it is.

Robert L. Glass

Les besoins fonctionnels — la logique d'affaires — ne représentent qu'une fraction des requis devant être réalisés pour un système informatique robuste. Les aspects de performance, d'utilisabilité, de sécurité, de maintenabilité, de disponibilité, de fiabilité, de compatibilité, de portabilité et d'extensibilité représentent une part importante du travail à réaliser et font partie des normes du domaine.

En matière d'estimation initiale, on note le constat suivant :

Constat 5 — L'estimation du projet *CASA* était irréaliste et ne prenait pas en compte le degré d'incertitude inhérent au projet dans sa phase initiale.

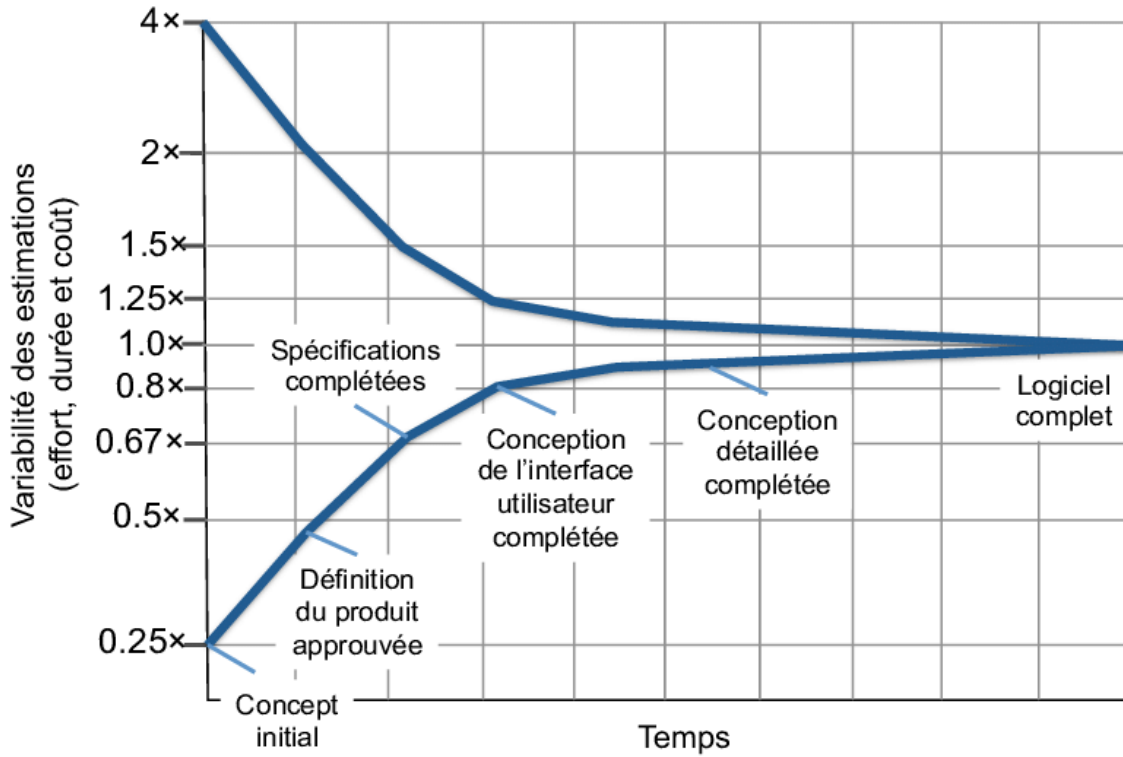


FIGURE 4.1. – Diagramme stylisé et adapté pour illustrer le cône d'incertitude (d'après Boehm (1984, p. 8) et McConnell (1998, p. 31))

5. Mirage des *PGI*

La démonstration qu'un *PGI* était la solution optimale pour la *SAAQ* est loin d'avoir été clairement établie. Il faut reconnaître que les éditeurs de *PGI* — *SAP* et *Oracle* pour les *PGI* de cette taille — peuvent présenter des arguments forts efficaces pouvant s'avérer convaincants pour les personnes ayant des carences au niveau des connaissances numériques portant sur les savoir-faire techniques, sur la conduite des projets en SI et les changements organisationnels.

Pour résumer, un *PGI* représente une architecture monolithique, en une seule application, ce qui rend difficile la cohabitation avec des logiciels dits “les meilleurs” — *best of breed*.

Un *PGI* rend extrêmement difficile la migration en fin de vie utile du produit vers une autre solution. Une des difficultés vient du fait qu'un *PGI* cohabite mal avec des logiciels externes. Ce point fondamental, la difficulté de migration, est très important, les *PGI* n'étant plus les meilleures solutions modernes, leur remplacement est ardu. Comment s'effectuera le transfert vers une autre plateforme, à quels coûts ? Les éditeurs de *PGI* savent très bien que leurs clients sont pratiquement captifs. Cela se traduit par des tarifs exorbitants. Leurs intégrateurs en profitent également. L'organisation hôte devient un otage face à l'éditeur et à l'intégrateur.

Un des effets pervers d'un *PGI*, à cause de son caractère monolithique, est la quasi-obligation d'avoir une approche *big bang* pour la mise en production. Cette approche, pour un système couvrant plusieurs fonctions critiques de l'organisation, est fort risquée et demande des tests intensifs et complets avant son exécution. Voici un endroit où le projet *CASA* a échoué à remplir ses attentes.

La *SAAQ* doit régulièrement prendre en charge des modifications à des lois et à des règlements dans la conduite de ses activités. Avec *SAP*, à titre d'exemple pour chaque changement, il faut faire affaire avec des ressources externes spécialisées. Leurs honoraires sont conséquents, le code résultant devient la propriété de *SAP*, et des coûts de licence et support annuels additionnels seront facturés, si les renseignements transmis à ce propos lors des audiences de la *Commission* sont justes.

Pour accommoder plusieurs clients différents, un progiciel offre une certaine flexibilité de paramétrage, ce qui augmente sa complexité. Même avec cette flexibilité, il est fréquent que le client doive modifier ses façons de faire pour s'adapter au progiciel ou, alors, effectuer un développement spécifique sur le progiciel. La règle du pouce dans l'industrie est que, si

vous devez apporter des modifications pour plus de 15 % des fonctionnalités, il vaut mieux y aller avec un logiciel sur mesure. La raison est que la complexité du progiciel est plus grande que celle d'un logiciel sur mesure, car le progiciel doit répondre à une gamme plus grande de besoins. Dans le cas de *SAP*, on aura dû modifier entre 60 % et 70 % du progiciel pour la livraison 2, selon le témoignage du vérificateur interne de l'époque.

Pour des fonctions auxiliaires et non stratégiques à la mission de l'organisation, les progiciels — et non les *PGI* — peuvent représenter une solution intéressante pourvu qu'une mobilité entre fournisseurs reste possible. Dans le cas d'une nouvelle organisation, l'utilisation d'un progiciel pour les fonctions auxiliaires peut également avoir un effet structurant. Un *PGI* représente des risques importants par rapport à un dosage optimum entre des progiciels *best of breed* et des logiciels sur mesure. L'argument de la complexité de cohabitation entre différents logiciels nous semble faible avec les architectures informatiques modernes. Par exemple, les *Interface de Programmation d'Application (IPA)* permettent aux logiciels de communiquer entre eux, facilitant l'échange de données et de fonctionnalités. En d'autres termes, une *IPA* agit comme un intermédiaire, permettant à différentes applications de travailler ensemble sans que les développeurs aient besoin de comprendre les détails internes de chaque système.

En ce qui concerne les *PGI*, les constats suivants sont à souligner :

Constat 6 — Une analyse rigoureuse de solutions alternatives à un *PGI*, réalisées selon les bonnes pratiques, n'a pas été documentée.

Constat 7 — Un *PGI* crée une dépendance forte de la *SAAQ* envers son éditeur et son intégrateur, impactant ainsi la mission de l'organisation.

Constat 8 — L'ampleur des modifications requises pour prendre en charge les requis de la livraison 2 du projet *CASA* tend à démontrer que le *PGI* n'était pas la solution optimale pour ce faire.

Constat 9 — La dépendance envers un seul système sous le contrôle d'un seul fournisseur et d'un seul intégrateur engendre de nombreux désavantages pour l'organisation hôte, notamment la difficulté de faire cohabiter différents logiciels pour des fonctions non stratégiques de l'organisation.

Constat 10 — L'approche *big bang* pour la mise en production de la livraison 2 du projet *CASA* représentant un risque élevé pouvant affecter la mission même de l'organisation n'était pas recommandée.

6. Cout total de possession

Le cout total de possession *CTP* est aussi connu sous le sigle anglais *TCO* — Total Cost of Ownership.

*Gartner defines total cost of ownership (TCO) a comprehensive assessment of information technology (IT) or other costs across enterprise boundaries over time. For IT, TCO includes hardware and software acquisition, management and support, communications, end-user expenses and the opportunity cost of downtime, training and other productivity losses.*¹

Plusieurs tableaux financiers ont été présentés à la *Commission*. Certains de ces tableaux incluait les couts de récurrence, d'autres, non, d'autres encore seulement les couts de récurrence de l'*Alliance*. À l'occasion, les tableaux présentés couvraient des périodes différentes : cinq, dix ou quinze ans. Pour pouvoir s'y retrouver, le *CTP* est recommandé. Cette approche permet également de comparer plusieurs solutions différentes en utilisant une base commune. Le *CTP* est indépendant des méthodes utilisées pour comptabiliser l'amortissement d'un système.

Le *CTP* comprend, à titre d'exemple :

- les couts d'infrastructure — matériels et logiciels dédiés ;
- les couts des logiciels d'application ;
- les couts associés à la paramétrisation et au développement ;
- les couts de conversion des données ;
- les couts de formation du personnel ;
- les couts associés à la sécurité et aux sites de relève ;
- les couts d'opération ;
- les couts de support et d'entretien ;
- les couts récurrents de licence ;
- les couts de maintenance et de mise à jour des versions ;
- les couts de retrait du système.

Constat 11 — Une méthode uniforme pour calculer de manière complète le *CTP* des différents scénarios des solutions étudiées et leurs variances dans le temps n'a pas été utilisée dans le projet *CASA*.

1. <https://www.gartner.com/en/information-technology/glossary/total-cost-of-ownership-tco>

7. Agile

En ce qui concerne l'emploi d'une approche dite *Agile*, la question qui se pose est : peut-on être *Agile* avec 600 personnes y incluant des développeurs externalisés à l'étranger ?

Pour évaluer si l'approche *Agile* est pertinente pour un projet, Boehm et Turner (2003) présentent un diagramme où l'on inscrit l'évaluation de cinq dimensions (voir le diagramme 7.1 extrait de Boehm et Turner (2003, p. 59), voir également Boehm et Turner (2008)). Plus l'évaluation des cinq dimensions tend vers le centre, plus le projet est un candidat intéressant pour l'approche *Agile*. Plus l'évaluation des cinq dimensions tend à s'éloigner du centre, plus une approche dite *disciplinée* est de mise.

La première dimension est la criticité des impacts en cas d'erreurs ou d'échec du projet. Par exemple, si des vies humaines sont en jeu, une approche très rigoureuse est nécessaire. Dans certains de ces cas, il est fréquent que les tests puissent représenter près de 90 % de l'effort total d'un tel projet. Si le système envisagé, en cas d'échec, n'a pas de conséquences importantes, l'approche *Agile* y est possible.

La deuxième dimension est le nombre de personnes travaillant activement au projet. Une petite équipe favorise des échanges informels alors que, si plusieurs centaines de personnes sont impliquées, les échanges sont obligatoirement plus formalisés. Ce point milite pour des projets plus petits permettant à l'approche *Agile* d'être optimale.

La troisième dimension a trait au taux de changement dans les requis. Si le nombre de requis appelé à changer est élevé, cela demande une plus grande agilité.

La quatrième dimension évalue la tolérance de l'organisation à une certaine flexibilité en cours de projet. Par exemple, une organisation sujette à une reddition de compte stricte par son cadre réglementaire ne favorise pas l'adoption d'une approche *Agile*.

La cinquième dimension évalue la capacité de l'équipe à évaluer ses manières de faire, donc à faire preuve de réflexivité, et à les modifier dans une optique d'amélioration, le cas échéant. Plus la proportion de membres expérimentés et en mesure de faire preuve de réflexivité est élevée, plus l'approche *Agile* est efficace.

Selon les dimensions du modèle de Boehm et Turner (2003), le projet *CASA*, avec plus de 600 personnes, une importance capitale pour l'organisation, une équipe ayant une performance inconnue, une obligation de reddition de compte stricte et un faible taux de

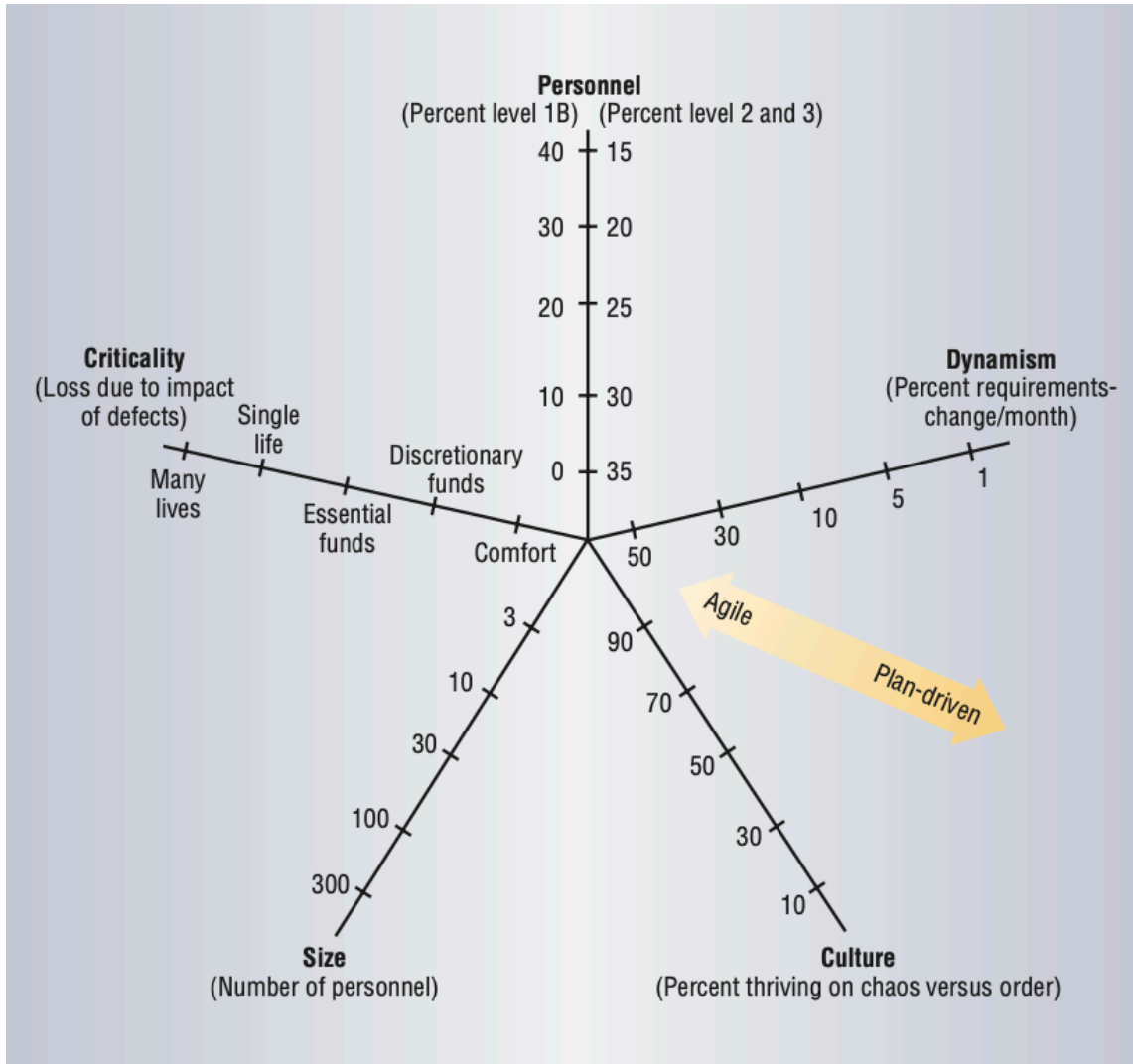


FIGURE 7.1. – Diagramme extrait de Boehm et Turner (2003, p. 59)

changements pour les requis, ne favorisait pas l'adoption et l'utilisation d'une approche *Agile*.

L'approche *Agile* connaît des dérives. Cette approche ayant connu une grande diffusion, de très nombreux cabinets-conseils du domaine et plusieurs *gourous* se sont souvent proclamés *Agile*. Et ce, en dépit du fait que très peu ont apporté des modifications à leurs pratiques.

Des processus sont apparus pour appliquer l'approche¹ *Agile*. Les plus connus sont *Scrum*, *Kanban* et *eXtreme Programming XP*. Une industrie s'est greffée au concept *Agile* pour vendre des cours, du conseil, etc. Les résultats sont souvent loin des prétentions annoncées. Là, comme ailleurs, une certaine prudence est de mise lorsque l'on promet des résultats rapides et presque « magiques ». L'important est de comprendre l'esprit du *Manifeste Agile* (*Manifeste pour le développement Agile de logiciels*, 2001) et de le mettre en pratique avec des petites équipes pour des projets rencontrant les critères établis. Le diagramme de Boehm et Turner (2003) est un outil d'évaluation qui s'est avéré fort utile pour identifier les projets candidats ou pour segmenter un projet d'une grande taille en projets plus petits.

Le point principal de l'approche *Agile* est que les progrès se mesurent avec des logiciels fonctionnels, tout autre chose étant accessoire.

Concernant l'approche *Agile*, le principal point à retenir est :

Constat 12 — Les principaux facteurs favorisant l'utilisation de l'approche *Agile* n'étaient pas présents dans le projet *CASA*.

1. https://fr.wikipedia.org/wiki/M%C3%A9thode_agile

8. Productivité des développeurs

The most important factor in software work is not the tools and techniques used by the programmers, but rather the quality of the programmers themselves.

Robert L. Glass

The best programmers are up to 28 times better than the worst programmers, according to "individual differences" research. Given that their pay is never commensurate, they are the biggest bargains in the software field.

Robert L. Glass

Il est reconnu que certains hauts dirigeants sont plus performants que d'autres. C'est d'ailleurs leur principal argument pour exiger des conditions de rémunération conséquentes. De même, il est reconnu que certaines équipes sportives sont d'un calibre supérieur à d'autres équipes.

Qu'en est-il en informatique ?

Un des éléments les plus méconnus du domaine informatique est la grande variance dans la productivité des développeurs. Du point de vue des personnes extérieures au domaine, tous les développeurs se valent. De plus, selon toute apparence, ils semblent rapidement productifs et forment des équipes cohésives. La littérature du domaine a documenté depuis longtemps le contraire, en particulier un livre classique sur le sujet (DeMarco et Lister, 1999). Selon les études réalisées, une équipe prend quatre à six mois avant d'être cohésive. Certains développeurs ont même une productivité nette négative, et donc, leur retrait d'une équipe améliore la productivité de celle-ci (Schulmeyer, 1992). Les écarts de productivité entre développeurs peuvent dépasser un facteur de dix.

To the economic question "Why is software so expensive?" the equally economic answer could be "Because it is tried with cheap labour."

Edsger Wybe Dijkstra

Ces éléments sont connus des intégrateurs. Ces derniers font face à un dilemme. D'une part, il leur est impossible de facturer un taux élevé pour les supers développeurs, ce point est difficile à vendre à quelqu'un extérieur au domaine. D'autre part, si l'intégrateur assigne les développeurs les plus productifs au projet, il en résultera des revenus moindres, car le projet sera réalisé beaucoup plus rapidement.

Les intégrateurs savent que la demande du client d'aller à cout fixe est irréaliste, car les besoins vont évoluer au fur et à mesure que l'incertitude concernant les besoins va diminuer. Ils acceptent néanmoins cette condition du donneur d'ordre, sachant qu'il souhaitera assurément modifier les requis et, en conséquence, le cout pourra être revu à la hausse. La tactique optimale pour les intégrateurs consiste donc à assigner le maximum de ressources d'une productivité moyenne rapidement, soit avant un ajout au budget ou même l'interruption du projet, maximisant ainsi leurs revenus dans tous les cas.

Selon notre compréhension des informations divulguées lors des audiences de la *Commission*, la réalisation de la livraison 2 du projet *CASA* représentait environ deux-millions d'heures de travail pour environ 1 300 *RICEFWs*. Basés sur ces estimations, nous en arrivons à une moyenne de 1 500 heures par *RICEFW*, soit environ une année-personne par *RICEFW*. Sans plus d'information, ce résultat nous paraît élevé pour adapter un *PGI* qualifié d'optimal pour répondre aux besoins de la *SAAQ*.

Constat 13 — L'ampleur des adaptations requises pour répondre aux besoins de la *SAAQ*, lors de la livraison 2, est excessive dans le cadre de l'utilisation d'un *PGI*.

En partant du principe qu'il faut au moins quatre mois pour qu'une équipe devienne vraiment productive, et en considérant le recrutement de 400 personnes en seulement six mois pour mener à bien la livraison 2 du projet *CASA*, nous arrivons à un montant de trente-millions de dollars. Ce montant représente donc le cout encouru pour rendre l'équipe opérationnelle. Cette estimation est basée sur les informations fournies par DeMarco et Lister (1999).

Concernant les couts de gestion de projet, ces derniers représentent habituellement de 10 à 15 % des couts totaux d'un projet. S'agissant d'un projet de près d'un milliard de dollars, la gestion du projet représente entre cent et cent-cinquante-millions de dollars.

La gestion des développeurs internes, surtout dans les organismes publics et parapublics, est paradoxale. À l'interne, il est souvent difficile d'obtenir des ressources permanentes additionnelles, et ce, malgré les nombreux départs à la retraite. Les développeurs restant en poste sont forcés de consacrer un temps de plus en plus important au support des systèmes patrimoniaux. Les nouveaux projets, pour moderniser les systèmes vieillissants, font alors appel à des ressources externes. Cette manière de faire éloigne les nouveaux talents, amplifiant ainsi la tâche de travail des quelques braves en soutien à un fardeau

de plus en plus lourd jusqu'à leur départ pour la retraite. Cette situation appauvrit en compétences et en qualité d'exécution les organismes de façon importante.

S'agissant de la gestion des développeurs, on note le constat suivant :

Constat 14 — Le développement et la gestion d'équipes informatiques internes aux organisations sont des éléments essentiels pour la conduite de leur mission. L'ajout de ressources externes ponctuelles est fort coûteux. Ce point semble avoir été très peu abordé lors des témoignages devant la *Commission*.

9. Conclusion

La folie, c'est de faire toujours la même chose et de s'attendre à un résultat différent.

Citation attribuée à Albert Einstein

Il est difficile de tirer des conclusions définitives et de formuler des recommandations sans avoir en main l'ensemble des facteurs à considérer pour qu'elles puissent être mises en place de façon réaliste. De plus, la réalisation de certaines recommandations exige des actions concertées de la part de plusieurs instances, et ce, sur une longue période. En particulier, c'est le cas pour effectuer un changement de culture permettant de mettre en place les conditions de réalisation d'un développement informatique par des ressources internes. C'est donc *cum grano salis* que nous proposons les recommandations qui suivent. Nous suggérons également la lecture des recommandations contenues dans le document *Conclusions and recommendations of the Dutch temporary committee on government ICT projects* (Representatives of the Netherlands, 2014) déjà cité au début du présent mémoire¹.

Il est essentiel de pallier la carence de connaissances numériques des administrateurs.

Recommandation 1 — Préparer et diffuser une formation spécifique pour les administrateurs des secteurs publics et parapublics pour préparer ces derniers à remplir leur rôle de façon plus éclairée et proactive pour le domaine des *SI*. Il existe déjà un matériel pertinent sur le sujet produit par plusieurs associations professionnelles.

Nous avons présenté le cône d'incertitude de Boehm démontrant qui est très difficile, si non impossible d'établir un échéancier et un cout précis pour un projet avant de l'entreprendre.

Recommandation 2 — Modifier le *Décret 1159-2022 du 22 juin 2022 Concernant les Règles relatives à la gestion des projets en ressources informationnelles*,

1. En guise d'élément pertinent de veille concernant les approvisionnements numériques et technologiques dans le secteur public, le *National Audit Office* du Royaume-Uni a présenté, le 9 janvier 2025, un rapport contenant des constats et des recommandations pour guider le gouvernement sur la manière de relever les défis à ce sujet (<https://www.nao.org.uk/insights/governments-approach-to-technology-suppliers-addressing-the-challenges/>).

en particulier la *SECTION II — ÉVALUATION DU COÛT D'UNE INTERVENTION*², pour tenir compte du cône d'incertitude de Boehm.

Nous avons abordé plusieurs enjeux soulevés par l'utilisation d'un *PGI*.

Recommandation 3 — Éviter, pour les activités touchant le cœur de la mission d'une organisation, d'être à la merci d'un seul système sous le contrôle d'un seul fournisseur et d'un seul intégrateur.

Plusieurs études démontrent que plus un projet est de grande envergure et plus sa durée est longue, plus les risques d'échec ou de débordement sont décuplés. De plus, l'approche *Agile* est mieux adaptée aux projets de moindre envergure.

Recommandation 4 — Fractionner les projets de grande envergure en plus petits projets d'une durée ne dépassant pas une année ou deux. Chacun de ces petits projets visant un nombre restreint de parties prenantes. Ces petits projets peuvent être réalisés par des équipes de quelques dizaines de personnes, au mieux, moins d'une dizaine de personnes.

Dans le projet *CASA*, les adaptations apportées au *PGI*, payées par la *SAAQ* semblent demeurer la propriété de *SAP*.

La Confédération suisse a adopté la loi — *LMETA*³ — obligeant tout développement de logiciel, au niveau fédéral, payé par des fonds publics à être mis sous licence libre et accessible à tous.

*La mise en œuvre d'EMBAG [selon l'acronyme en allemand, LMETA, en français], devrait servir de modèle à d'autres pays envisageant des mesures similaires. La loi vise à promouvoir la souveraineté numérique et à encourager l'innovation et la collaboration au sein du secteur public. À mesure que la Suisse adoptera cette approche, les avantages des logiciels libres, à savoir une plus grande sécurité, un meilleur rapport cout-efficacité et une confiance accrue du public, pourraient devenir plus évidents.*⁴

Recommandation 5 — Diffuser tout logiciel réalisé avec des fonds publics sous licence libre.

Le présent mémoire est basé sur les informations fragmentaires disponibles publiquement en date du document. Il est possible que certains contenus, constats et recommandations présentés dans ce mémoire doivent être reconsidérés à la suite de nouvelles informations.

2. https://www.publicationsduquebec.gouv.qc.ca/fileadmin/gazette/pdf_encrypte/lois_reglements/2022F/77766.pdf

3. <https://www.fedlex.admin.ch/eli/oc/2023/682/fr>

4. <https://interoperable-europe.ec.europa.eu/collection/open-source-observatory-osor/news/new-open-source-law-switzerland?etrans=fr>.

A. Rompre avec l'illettrisme numérique — Le Devoir

LEDEVOIR

Rompre avec l'illettrisme numérique



Louise-Maude Rioux Soucy

Publié le 17 mai **Éditorial**
Éditoriaux

Le fiasco SAAQclic est-il la pointe d'un iceberg numérique qui menace le Québec ? Chose certaine, à entendre le nouveau ministre de la Cybersécurité et du Numérique, il vaudrait mieux que les Québécois préparent leurs canots et leurs vestes de sauvetage, car son ministère « n'a pas les outils de gouvernance et de gestion » nécessaires pour éviter un autre naufrage.

Il y a déjà de la houle autour du Dossier de santé numérique (DSN), destiné à stocker les données médicales des Québécois. Sa mise en place dans deux régions « vitrines » est récemment passée au jaune (<https://www.tableaubordprojetsri.gouv.qc.ca/tableau-de-bord/projet/59362367>), pour l'échéancier comme pour le coût. Les eaux sont aussi agitées autour de la numérisation des dossiers judiciaires et des registres de tous les palais de justice. L'échéancier est passé au rouge franc (<https://www.tableaubordprojetsri.gouv.qc.ca/tableau-de-bord/projet/4526101>), avec une augmentation du temps de livraison de 83,8 %.

La tempête s'est déjà levée autour du projet SIFARH, qui vise la centralisation de la paie, de la chaîne d'approvisionnement et de la gestion des ressources humaines dans le réseau de la santé et des services sociaux. Outre des dépassements de coûts substantiels, des ratés dans les contrats et des retards importants, il faut savoir que le piètre état du chantier (<https://www.tableaubordprojetsri.gouv.qc.ca/tableau-de-bord/projet/59362252>) (réalisé à seulement 12 %) empêche le versement de primes prévues dans les nouvelles conventions collectives du personnel, et ce, plus de cinq mois après la signature de l'entente. C'est gênant !

L'aveu d'impuissance du ministre Gilles Bélanger tranche avec les manières de son prédécesseur déchu. Même au plus fort de la tempête entourant la numérisation de la Société de l'assurance automobile du Québec (SAAQ), Éric Caire n'avait jamais rien trouvé à se reprocher « hormis de ne pas avoir été assez méfiant » (https://www.ledevoir.com/politique/quebec/849220/eric-caire-demission?utm_source=recirculation&utm_medium=hyperlien&utm_campaign=corps_texte). S'il n'avait rien vu, a-t-il répété sur toutes les tribunes, c'est qu'il n'y avait rien à voir dans cette purée de pois : ni indice ni feu rouge.

Pourtant, des indices et des feux rouges, on sait désormais qu'il y en a eu et qu'il y en a encore beaucoup, grâce notamment au Tableau de bord des projets en ressources informationnelles (<https://www.tableaubordprojetsri.gouv.qc.ca/accueil/>) et à la commission Gallant, dont les travaux vont bon train.

Ce qu'on y a entendu jusqu'à présent brosse le portrait d'un équipage politique et administratif coupé de son navire, réfugié derrière d'immenses œillères qui suintent la complaisance et l'incompétence. Nouveau capitaine de ce paquebot branlant, le ministre Bélanger a commencé à faire le tour des projets de transformation numérique en cours au gouvernement. Et ce qu'il voit dans sa lunette enfin

dégagée ne lui plaît pas du tout.

L'octroi de contrats et la gestion de projets en transformation numérique sont une bête noire pour les gouvernements d'un peu partout dans le monde. Le Québec n'y échappe pas, avec retards, explosions de coûts et abandons à la clé. Entre 2006 et 2015, le Vérificateur général a publié quatre rapports frappant sur les mêmes clous : des projets mal conçus, mal préparés et mal exécutés.

Toute ressemblance avec ce qui mine le dossier CASA/SAAQclic (ou le DSN, ou le SIFARH...) ne saurait être fortuite. La même litanie guette encore, et crier « Iceberg droit devant ! » ne suffira pas.

Oui, le **gouvernement Legault** (https://www.ledevoir.com/francois-legault?utm_source=recirculation&utm_medium=hyperlien&utm_campaign=corps_texte) a terriblement manqué de vigilance et promet de se ressaisir. Mais en a-t-il même les moyens ? Si nous en sommes encore à piétiner sur ce front, c'est aussi parce que l'État québécois accuse un vrai retard en matière de littératie numérique. Le commissaire Gallant a remué le couteau dans la plaie cette semaine en comparant le travail des administrateurs de la SAAQ à un « acte de foi » envers les **personnes chargées de présenter le projet** (https://www.ledevoir.com/politique/quebec/880015/gouvernement-savait-depuis-2022-budget-saaq-clic-avait-explose?utm_source=recirculation&utm_medium=hyperlien&utm_campaign=corps_texte).

L'illettrisme numérique généralisé de ses décideurs et de ses administrateurs nuit grandement au Québec. Son fort penchant pour la sous-traitance aussi, car ce dernier freine le développement des expertises locales en plus de lui coûter fort cher. Ici, c'est la moitié des budgets de technologies de l'information (TI) qui est attribuée à des ressources externes. Ailleurs au Canada, cette proportion va plutôt de 17 % à 35 %.

Il faut dire que le boom phénoménal des TI ces dernières années reste très concentré. Les six premières économies du monde accaparent **70 % de la valeur ajoutée mondiale de ce secteur** (<https://www.banquemondiale.org/fr/news/immersive-story/2024/03/05/global-digitalization-in-10-charts>), selon le Groupe de la Banque mondiale. Cela rend les innovations et les prises en charge locales encore plus difficiles. Spécialement quand les bogues et les pannes surviennent.

On l'a bien vu quand la SAAQ a été paralysée durant une quarantaine d'heures. Si la société d'État a été prompte à montrer Microsoft du doigt, elle a dû se rétracter après que des indices ont laissé entrevoir de mauvaises manipulations à l'interne. Qui a fauté ? La SAAQ est toujours incapable de le dire et a dû mandater une firme externe pour obtenir un audit neutre capable de trancher l'affaire dans le vif.

Ça ne va pas du tout. Il faut reprendre le contrôle de notre transformation numérique. Et pour naviguer dans ces eaux profondes et agitées, il faudra plus que de bonnes jumelles au ministre Bélanger. Le projet de loi 82 est censé lui dégager de la marge de manœuvre, mais il devra aller bien plus loin dans ses efforts pour que le Québec rattrape son retard numérique.

Ce texte fait partie de notre section Opinion. Il s'agit d'un éditorial et, à ce titre, il reflète les valeurs et la position du *Devoir* telles que définies par son directeur en collégialité avec l'équipe éditoriale.

Références et bibliographie

- Balka, K., Heslin, B. et Risse-Tenk, S. (2022). Unlocking the potential of public-sector IT projects. *McKinsey & Company*. <https://www.mckinsey.com/industries/public-sector/our-insights/unlocking-the-potential-of-public-sector-it-projects>
- Boehm, B. W. (1984). Software engineering economics. *IEEE Transactions on Software Engineering*, *SE-10*(1), 4-21. <https://doi.org/10.1109/TSE.1984.5010193>
- Boehm, B. W. et Turner, R. (2008). *Balancing agility and discipline : a guide for the perplexed* (6. printing). Addison-Wesley.
- Boehm, B. et Turner, R. (2003). Using risk to balance agile and plan-driven methods. *Computer*, *36*(6), 57-66. <https://doi.org/10.1109/MC.2003.1204376>
- Confédération suisse, A. fédérale de la. (2023, décembre). Loi fédérale sur l'utilisation de moyens électroniques pour l'exécution des tâches des autorités (LMETA) [EMBAG, en allemand]. <https://www.fedlex.admin.ch/eli/oc/2023/682/fr>
- DeMarco, T. et Lister, T. R. (1999). *Peopleware : productive projects and teams* (2 éd.). Dorset House Publishing Co.
- Dijkstra, E. W. (1982). « Why is software so expensive ? » An explanation to the hardware designer. Dans *Selected writings on computing : A personal perspective* (p. 338-348). Springer. https://doi.org/10.1007/978-1-4612-5695-3_61
- Glass, R. L. (2010). *Facts and fallacies of software engineering* (9 impression). Addison-Wesley.
- Laptick, S. (2025, mars). Big bang vs. gradual data migration : pros, cons, and best practices. *XB Software*. <https://xbsoftware.com/blog/big-bang-or-gradual-data-migration/>
- Manifeste pour le développement Agile de logiciels. (2001). <https://manifesteagile.fr/#valeurs> (texte original en anglais <https://agilemanifesto.org/>)
- McConnell, S. (1998). *Software project survival guide*. Microsoft Press.

- National Audit Office (Royaume-Uni), I. and T., Department for Science. (2025, janvier). *Government's approach to technology suppliers : addressing the challenges*. <https://www.nao.org.uk/insights/governments-approach-to-technology-suppliers-addressing-the-challenges/>
- Open Source Observatory (OSOR) (Thévenet, A.). (2024). *Nouvelle loi Open Source en Suisse. La Suisse exige la divulgation du code source du logiciel pour le secteur public : un jalon juridique*. <https://interoperable-europe.ec.europa.eu/collection/open-source-observatory-osor/news/new-open-source-law-switzerland?etrans=fr>
- Peppard, J. (2015). Executives get the IT they deserve. *Harvard Business Review Digital Articles*, 2-5.
- Peppard, J. et Bastien, R. (2024). The hidden causes of digital investment failures. *MIS Quarterly Executive*, 23(2), 131-144. <https://aisel.aisnet.org/misqe/vol23/iss2/2>
- Perkins, B. (2007). 12 things you know about projects but choose to ignore. *Computerworld*. <https://www.computerworld.com/article/1690019/12-things-you-know-about-projects-but-choose-to-ignore.html>
- Québec, G. du. (2022, juillet). Règles relatives à la gestion des projets en ressources informationnelles. https://www.publicationsduquebec.gouv.qc.ca/fileadmin/gazette/pdf_encrypte/lois_reglements/2022F/77766.pdf
- Representatives of the Netherlands, H. of. (2014, octobre). Temporary committee on government ICT projects. https://www.houseofrepresentatives.nl/sites/default/files/news_items/conclusions_and_recommendations_0.pdf
- Rioux Souci, L.-M. (2025). Rompre avec l'illettrisme numérique. *Le Devoir*. <https://www.ledevoir.com/opinion/editoriaux/880870/editorial-transformation-numerique-rompre-illettrisme-numerique?>
- Schulmeyer, G. G. (1992, juin). The net negative producing programmer. http://web.archive.org/web/20011023084845/http://pyxisinc.com/pyxis_artmain.html
- Taleb, N. N. (2020). *Le cygne noir : la puissance de l'imprévisible ; suivi de Force et fragilité : réflexions philosophiques et empiriques*. Les Belles Lettres.
- Volden, G. H. et Samset, K. (2023). Introduction. Dans T. M. Williams, K. Samset et G. H. Volden (dir.), *The front-end of large public projects : paradoxes and ways ahead* (p. 1-9). Routledge.

Williams, T., Klakegg, O. J., Walker, D. H. T., Andersen, B. et Magnussen, O. M. (2012). Identifying and acting on early warning signs in complex projects. *Project Management Journal*, 43(2), 37-53. <https://doi.org/10.1002/pmj.21259>