

# Document d'Ingénieurs Canada sur l'exercice dans le domaine du génie logiciel

## Avertissement

Les guides nationaux et les documents d'Ingénieurs Canada sont élaborés par des ingénieurs, en collaboration avec les organismes de réglementation du génie provinciaux et territoriaux. Ces guides sont destinés à favoriser des pratiques uniformes à l'échelle du pays. Ce ne sont pas des règlements ni des règles. Ils visent à définir et à expliquer certains aspects de l'exercice et de la réglementation du génie au Canada.

**Les guides nationaux et documents d'Ingénieurs Canada n'établissent pas de norme légale de diligence ou de conduite et ne comprennent ni ne constituent d'avis juridique ou professionnel.**

Au Canada, le génie est réglementé par les organismes de réglementation du génie en vertu des lois provinciales et territoriales. Ces organismes sont libres d'adopter, entièrement ou en partie, les recommandations contenues dans les guides nationaux et les documents d'Ingénieurs Canada ou de ne pas les adopter. Il revient à l'organisme de réglementation de la province ou du territoire où exerce ou envisage d'exercer l'ingénieur de décider du bien-fondé d'une pratique ou d'une ligne de conduite.

## À propos de ce document d'Ingénieurs Canada

Ce document d'Ingénieurs Canada national a été préparé par le Bureau canadien des conditions d'admission en génie (BCCAG) en concertation avec les organismes de réglementation et il est destiné à fournir des orientations à ces organismes. Le lecteur est invité à consulter en même temps les lois et règlements pertinents de l'organisme de réglementation dont il dépend.

## À propos d'Ingénieurs Canada

Ingénieurs Canada est l'organisme national constitué des ordres provinciaux et territoriaux qui sont chargés de réglementer l'exercice du génie au Canada et de délivrer les permis d'exercice aux 295 000 membres de la profession.

## À propos du Bureau canadien des conditions d'admission en génie

Le Bureau canadien des conditions d'admission en génie est un comité du conseil d'Ingénieurs Canada composé de bénévoles. Il a pour rôle d'offrir du leadership national et des recommandations aux organismes de réglementation en ce qui concerne l'exercice du génie au Canada. À cet égard, il élabore à l'intention des organismes de réglementation et du public des guides et des documents d'Ingénieurs Canada qui permettent d'évaluer les compétences en génie, facilitent la mobilité des ingénieurs et favorisent l'excellence en matière d'exercice et de réglementation du génie.

# 1 Introduction

Ce document a pour objet de fournir aux organismes de réglementation du génie de l'information et des conseils concernant la discipline du génie logiciel. Il présente un exposé introductif qui traite de la nature de l'exercice dans le domaine du génie logiciel, en comparaison avec le développement de logiciels. Il vise à aider les responsables de la conformité et de l'application de la loi à cerner l'exercice du génie logiciel qui devrait être réglementé, mais il ne devrait pas être interprété comme limitant la validité du travail de génie logiciel qui ne relève pas de ce cadre.

À noter que ce document ne tente pas de décrire toute la portée et la profondeur de la discipline du génie logiciel. Pour plus de détails sur la portée du génie logiciel, le lecteur peut se reporter au Programme d'examens – Génie logiciel du Bureau canadien des conditions d'admission en génie [1] et au *Guide to the Software Engineering Body of Knowledge* de l'Institute of Electrical and Electronics Engineers [2].

De nombreux ingénieurs qui exercent dans les disciplines traditionnelles du génie ont suivi des programmes de formation agréés par le Bureau canadien d'agrément des programmes de génie, et la portée, les pratiques et les normes de ces disciplines sont bien définies. Cependant, les praticiens du génie logiciel qui souhaitent obtenir un permis d'exercice sont moins susceptibles d'être des diplômés d'un programme agréé en génie logiciel; ils viennent plutôt de divers milieux de l'industrie. Il y a des milliers de développeurs de logiciels qui travaillent au sein de l'industrie, dont bon nombre ne possèdent pas les connaissances, les compétences ni l'expérience nécessaires pour exercer la profession d'ingénieur. Dans certaines provinces ou territoires, ces personnes peuvent être admissibles à un permis restreint (ou restrictif), qui leur donne le droit de pratiquer le génie logiciel dans un champ d'exercice circonscrit.

Afin de protéger le public et d'empêcher des praticiens du développement logiciel non qualifiés d'assumer les responsabilités ou le titre d'ingénieur en génie logiciel, les organismes de réglementation doivent comprendre la portée de l'exercice réglementé dans le domaine du génie logiciel.

Ce document présente un outil simplifié et des conseils visant à aider les organismes de réglementation et le personnel responsable de l'application de la loi à reconnaître l'exercice du génie logiciel. Cela comprend une application de la définition de l'exercice du génie au domaine du logiciel, ainsi que des caractéristiques indiquant qu'une activité peut consister en l'exercice du génie logiciel qui ne peut être pratiqué que par un ingénieur en génie logiciel (c.-à-d. la portée exclusive du génie logiciel) [3], ainsi que des aspects qui peuvent être pratiqués par d'autres, en plus des ingénieurs en

logiciel. Le présent document n'est pas exhaustif; il vise uniquement à compléter les autres renseignements et outils actuellement utilisés par les organismes de réglementation pour réglementer ce domaine. Le présent document ne traite pas du permis restreint ou restrictif qui, dans certaines provinces ou territoires, est un moyen pour les candidats qui ne remplissent pas les conditions d'obtention du permis d'ingénieur de pratiquer le génie logiciel dans un champ d'exercice circonscrit.

## 2 Champ d'exercice dans le domaine du génie logiciel

Le génie logiciel est une discipline qui s'est révélée, pour la profession d'ingénieur, difficile à reconnaître et, par conséquent, à réglementer. Pour les besoins de la réglementation et de l'application de la loi, la portée du génie logiciel correspond[e] à la définition existante du génie énoncée dans le Guide national sur l'exercice de la profession d'ingénieur au Canada, soit :

L'« exercice de la profession d'ingénieur » consiste à préparer des plans, des études, des synthèses, des évaluations et des rapports, à donner des consultations, et à diriger, surveiller et administrer les travaux précités,

lorsque cela exige l'application des principes d'ingénierie

et

est associé à la protection de la vie, de la santé, de la propriété, des intérêts économiques, de l'environnement et du bien-être public.

Dans le cas du génie logiciel, un logiciel (ou un système à fort contenu logiciel) peut par conséquent être considéré comme étant un *travail d'ingénierie* si les **deux** conditions suivantes s'avèrent :

Le développement [4] du logiciel a exigé « l'application d'une approche systématique, disciplinée et quantifiable au développement, à l'exploitation et à la maintenance du logiciel » [5]

et

l'on peut raisonnablement s'attendre à ce qu'une défaillance ou le mauvais fonctionnement du système porte préjudice à la vie, la santé, la propriété, les intérêts économiques, l'environnement et le bien-être public.

En règle générale, lors de l'évaluation d'un système à fort contenu logiciel, ces conditions sont abordées dans l'ordre inverse : i) Y a-t-il un impact sur l'intérêt du public? puis ii) Les principes d'ingénierie logicielle ont-ils été appliqués? La section suivante présente l'évaluation de ces questions et une explication des principes d'ingénierie logicielle.

## 3 Indicateurs du génie logiciel

La réglementation du génie logiciel et l'application de la loi dans ce domaine se sont révélées difficiles, du fait que des activités comme la programmation logicielle semblent souvent présenter des recoupements avec le génie logiciel. Pour revenir à la définition présentée ci-dessus, il y a deux aspects clés d'une activité qui indiquent que cette activité relève de la portée exclusive du génie logiciel :

- » elle est associée à l'intérêt du public (la vie, la santé, la propriété, les intérêts économiques, le bien-être public ou l'environnement) et
- » elle exige l'application des principes d'ingénierie.

L'application de ces deux aspects à l'exercice dans le domaine du génie logiciel est examinée ci-après.

### 3.1 Protection de l'intérêt du public

Première question à se poser : « Est-ce que ce travail/cette activité a une incidence matérielle sur l'intérêt du public, à savoir la vie, la santé, la propriété, les intérêts économiques, l'environnement ou le bien-être public? »

Un indicateur clé permettant de déterminer qu'un logiciel ou un système relève de la portée exclusive du génie logiciel est le fait que la défaillance ou le mauvais fonctionnement du logiciel ou du système peut présenter un danger pour « la vie, la santé, la propriété, les intérêts économiques, l'environnement ou le bien-être public ». On pourrait invoquer le fait que le logiciel lui-même ne peut pas présenter de danger, mais un logiciel qui fait partie d'un système le peut certainement, et la nature du risque qui en découle dépend de la nature et de l'application du système. À l'évidence, de tels dangers peuvent présenter un large éventail de niveaux de risque, et cela peut être un facteur pour déterminer s'il s'agit de génie logiciel. Par exemple, un logiciel qui contrôle directement un matériel dangereux serait catégorisé comme étant *essentiel à la sécurité*[3] et ainsi le développement d'un tel logiciel serait presque certainement considéré comme relevant de l'exercice du génie logiciel. En revanche, la défaillance d'un logiciel de jeu ne présenterait un risque que pour les intérêts économiques de l'entreprise de développement (et non pour le public), et cette activité ne serait probablement donc pas considérée comme relevant du génie logiciel en fonction de cet indicateur.

Le développement d'un logiciel qui constitue une partie essentielle d'un système, ou qui interagit directement avec un système qui est lui-même un travail d'ingénierie et qui est donc associé à la sécurité serait aussi normalement considéré comme relevant de l'exercice du génie logiciel. Il se pourrait qu'un ingénieur (exerçant une autre discipline) assume la responsabilité de l'ensemble du système. Dans ce cas, l'ingénieur responsable doit accorder toute l'attention voulue aux lignes directrices relatives à la supervision adéquate et à l'exercice correspondant aux compétences, comme cela est exigé dans toutes les situations où des personnes non titulaires de permis participent à des travaux d'ingénierie sous la

supervision d'un ingénieur.

La définition d'un logiciel essentiel à la sécurité englobe aussi généralement tout logiciel qui est conçu pour atténuer les conséquences d'un accident ou pour assurer la reprise subséquente (p. ex. : système d'arrêt d'urgence), de sorte que le développement d'un tel logiciel constituerait aussi normalement l'exercice du génie logiciel. En outre, un logiciel peut devenir essentiel à la sécurité du fait de l'application du système dans lequel il est utilisé. Par exemple, un logiciel faisant partie d'un commutateur de réseau de télécommunications pourrait être essentiel à la sécurité si le commutateur est utilisé pour gérer des systèmes de sécurité ou de maintien des fonctions vitales.

La défaillance d'un réseau de télécommunications pourrait aussi avoir une incidence matérielle sur les intérêts économiques du grand public ou sur l'économie d'une région, de sorte que le développement d'un tel réseau serait considéré comme relevant du génie logiciel en fonction de cet indicateur également.

Un aspect de sécurité unique aux logiciels est que ceux-ci ont souvent pour fonction d'assurer la sécurité des données privées. Des données personnelles et d'entreprise d'importance cruciale peuvent être produites, traitées et/ou protégées par un logiciel. Par conséquent, la défaillance d'un tel logiciel ou système pourrait être préjudiciable au bien-être public et pourrait menacer la vie dans des circonstances particulières (p. ex. : en échouant à protéger l'identité des personnes et en les exposant ainsi au risque d'être victimes d'actes criminels). Le développement de ces systèmes peut donc être considéré comme relevant du génie logiciel.

Pour obtenir une illustration des très nombreuses façons dont les systèmes informatiques peuvent être préjudiciables au bien-être public, voir : *Forum on Risks to the Public in Computers and Related Systems* de l'Association of Computing Machinery (<http://catless.ncl.ac.uk/Risks/>).

### 3.2 Principes d'ingénierie

Deuxième question à se poser : « Est-ce que des connaissances et des compétences en génie ont été nécessaires et est-ce que des principes d'ingénierie ont été appliqués pour développer ce produit? »

Alors que l'impact sur l'intérêt du public est une caractéristique de **l'application** du logiciel, l'application des principes d'ingénierie est une caractéristique du **processus de développement** d'un logiciel. En plus des principes d'ingénierie générale, il y a aussi des principes d'ingénierie **logicielle** qui caractérisent le travail dans ce domaine. Sur le plan le plus général, ces principes d'ingénierie logicielle englobent « l'application d'une approche systématique, disciplinée et quantifiable au développement, à l'exploitation et à la maintenance d'un logiciel. » [8]. Pour déterminer si les principes du génie logiciel sont nécessaires, il faut évaluer le processus de développement et le produit lui-même.

Pour examiner le processus de développement d'un logiciel, il faut considérer ce *quiaurait dû être* entrepris au cours du cycle de vie du produit. Il faudrait rechercher des principes d'ingénierie générale et des principes d'ingénierie logicielle. Ce sont les principes d'ingénierie logicielle qui font que le travail va au-delà de la « simple programmation ». Ils comprennent :

» Aptitude à l'usage prévu

Le logiciel ou système a un usage ou une portée prévus, et les exigences sont définies en fonction de cette portée. On s'attend à ce que le logiciel ou système soit conforme à l'usage prévu.

» Utilisation de conceptions et de pratiques connues

La conception du logiciel ou système et les diverses pratiques utilisées durant le cycle de vie sont définies et sélectionnées avec soin, de façon appropriée au risque et aux autres aspects du problème d'ingénierie.

» Gestion du risque

Les risques de natures diverses sont analysés et gérés conformément au problème d'ingénierie. Les risques ayant une incidence sur la santé, la sécurité et le bien-être publics sont gérés de façon prioritaire, mais les risques techniques et financiers sont également pris en compte.

» Validation et vérification du travail

À chaque étape du cycle de vie, la portée du génie logiciel englobe la validation et la vérification du logiciel ou du système, y compris ses composantes et les produits de travail intermédiaires.

» Fiabilité et répétabilité du processus utilisé

Les processus spécifiques définis et sélectionnés pour utilisation durant le cycle de vie sont appropriés au problème d'ingénierie et au niveau de risque.

» Prise de responsabilité et traçabilité

Le logiciel ou système peut être constitué de nombreuses composantes et avoir des interfaces avec d'autres dispositifs ou systèmes, qui doivent tous fonctionner de façon fiable et correcte. La prise de responsabilité d'un système (travail d'ingénierie) englobe la responsabilité des composantes et des systèmes connexes.

Etant donné que le processus de développement qui a été utilisé n'est pas toujours connu, l'on peut aussi tenir compte des caractéristiques du logiciel. Si le logiciel reflète certains facteurs, il est alors probable que des principes d'ingénierie ont été nécessaires pour le produire. N'importe lequel des facteurs suivants peut nécessiter l'application de principes de génie logiciel :

» Caractère unique

Le produit est-il particulièrement novateur? Le développement de produits pour lesquels la conception logicielle suit des schémas familiers et peut vraisemblablement être implémentée à l'aide d'outils de haut niveau bien connus ne relèvera probablement pas du génie logiciel. Le développement de produits dont la conception n'est pas aussi évidente ou exige l'intégration de technologies de façons nouvelles constituera plus probablement du génie logiciel.

» Complexité du projet

La complexité globale du projet, qui pourrait être estimée en fonction du nombre de personnes participant à l'effort de développement ou de la taille des artefacts de conception (p. ex. : le code source), ou du nombre d'éléments connectés (à la fois physiques et virtuels), qui donne une indication de la nécessité d'appliquer des principes d'ingénierie.

» Apparence pour l'utilisateur final

Les applications où le logiciel fait partie d'un plus grand système (logiciel *intégré*) et est donc normalement transparent pour l'utilisateur font l'objet de plus grandes attentes quant à leur bon fonctionnement. De ce fait, leur processus de développement doit être plus soigné. Par exemple, nous nous attendons tous à ce que les pédales d'accélérateur et de freins de nos voitures fonctionnent comme prévu en tout temps, et nous ne considérons probablement pas souvent qu'il y a un nombre important de logiciels intégrés à ces systèmes.

» Nature interdisciplinaire

Le développement logiciel où la compréhension du domaine de problème exige la connaissance d'autres disciplines peut aussi constituer du génie logiciel, en raison des vastes bases en sciences physiques qui sont essentielles à la réussite de tels systèmes. Cela s'applique aussi à ce qu'on appelle souvent *logiciels d'ingénierie* – c'est-à-dire des logiciels dont les algorithmes internes intègrent certaines connaissances d'autres disciplines du génie, comme ceux qui sont utilisés pour exécuter des calculs ou des simulations de conception. Bien qu'il soit clair, dans ce cas, que l'ingénieur qui assume la responsabilité du produit final (c.-à-d. que le logiciel est utilisé pour la conception) est responsable des sorties du logiciel, il est également clair que des principes d'ingénierie sont appliqués et donc que le développement de tels logiciels peut s'inscrire dans l'exercice du génie.

» Cadre réglementaire

Si le logiciel a été développé à des fins industrielles et est exploité dans une industrie hautement réglementée (p. ex. : aviation, nucléaire, etc.), alors il est tout à fait probable que son développement relève du génie logiciel et que des principes d'ingénierie ont été nécessaires.

Le fait qu'un produit présente l'une de ces caractéristiques ne signifie pas qu'il *doit* s'agir de génie logiciel, mais la présence d'au moins une de ces caractéristiques est certainement indicatrice que ce *pourrait* être le cas.

Enfin, il faut tenir compte des activités de travail. Le *Guide to the Software Engineering Body of Knowledge*[9] de l'Institute for Electrical and Electronics Engineers (IEEE) a énuméré les activités qui font partie du processus de développement logiciel, à savoir :

- » Définition des exigences/besoins du logiciel
- » Conception du logiciel
- » Construction du logiciel
- » Mise à l'essai du logiciel
- » Maintenance du logiciel
- » Gestion de la configuration du logiciel
- » Gestion de l'ingénierie du logiciel (c.-à-d. gestion du projet logiciel)
- » Utilisation de processus d'ingénierie de logiciel (c.-à-d. cycle de vie du logiciel, évaluation et mesure des processus et des produits)
- » Utilisation d'outils et de méthodes de génie logiciel
- » Gestion de la qualité du logiciel

La section 4 présente un arbre décisionnel permettant un processus d'évaluation basé sur tous ces indicateurs.

## 4 Exemples

Il existe deux cas importants où le génie logiciel s'applique souvent :

1. Le logiciel est une composante intégrée d'un dispositif ou d'un système qui est lui-même un travail d'ingénierie;
2. Un système à fort contenu logiciel, où le logiciel est une composante principale qui peut exécuter diverses fonctions de traitement, d'analyse, etc.

Dans le premier cas, un système ou un dispositif peut être considéré comme étant un travail d'ingénierie s'il contrôle des systèmes mécaniques physiques ou des sources d'énergie, et s'il pose des risques potentiels pour la santé et la sécurité du public. Bien que le logiciel puisse constituer une composante essentielle d'un tel système ou dispositif, le dispositif en entier est un travail d'ingénierie et l'ingénieur responsable peut avoir été formé en génie mécanique, électrique ou dans une autre discipline du génie. Dans ce cas, la responsabilité du logiciel intégré est déléguée à un ingénieur en logiciel, qui travaillerait avec d'autres ingénieurs pour ce qui concerne la conception de l'interface, l'intégration et la sécurité.

Dans le second cas, un système à fort contenu logiciel est conçu et mis en œuvre sur une variété de plateformes de systèmes informatiques. Le logiciel peut ne pas être implémenté comme un travail d'ingénierie traditionnel – un dispositif ou système physique qui lui-même contrôle des systèmes mécaniques ou des sources d'énergie.

Les sections suivantes présentent des exemples de logiciels intégrés et de systèmes à fort contenu logiciel. Certains de ces exemples sont facilement reconnaissables comme relevant de l'exercice du génie logiciel, d'autres ne relèvent pas du génie logiciel, et d'autres encore illustrent les zones grises où le niveau de réglementation peut varier.

### 4.1 Énergie nucléaire

Des logiciels sont utilisés dans de nombreux aspects de la production d'énergie nucléaire, y compris les systèmes de protection, les systèmes de contrôle, la conception et l'enregistrement chronologique de données. L'industrie a établi des lignes directrices et des normes qui caractérisent les logiciels comme étant reliés, ou non, à la sûreté, et qui définissent les processus qui devraient être suivis dans le développement et la vérification de tels systèmes [10]. Le développement d'un nouveau logiciel considéré comme étant « relié à la sûreté », qui comprend des systèmes de protection et des systèmes de contrôle, est un exemple évident de génie logiciel.

L'aspect « sécurité » de ce genre de systèmes est clair : la défaillance du logiciel pourrait provoquer une défaillance catastrophique du réacteur, le dégagement de radiations nucléaires ou un arrêt accidentel entraînant une perte d'alimentation électrique pour certains clients. La nécessité d'appliquer des principes d'ingénierie est légèrement moins évidente, mais elle est bien établie par un examen des processus décrits dans les recommandations de l'Agence internationale de l'énergie atomique [11]. En raison du niveau de risque élevé associé à la production d'énergie nucléaire et, de ce fait, de la stricte réglementation de l'industrie, il est essentiel que tout développement logiciel suive un processus rigoureux et bien documenté. En outre, ce développement logiciel exige des connaissances interdisciplinaires reliées aux capteurs et aux actionneurs, ainsi qu'aux principes de fonctionnement des systèmes de centrales nucléaires.

### 4.2 Applications et instruments biomédicaux

Des logiciels sont utilisés pour activer ou contrôler des dispositifs biomédicaux, comme des appareils à rayonnement (imagerie), des robots chirurgicaux et des instruments non invasifs, tels que des sphygmomanomètres (appareils servant à mesurer la pression artérielle) [12]. Dans certains cas, les instruments médicaux interagissent directement avec les êtres humains et doivent être correctement contrôlés. Les logiciels et les systèmes globaux doivent être conçus non seulement pour fonctionner correctement, mais aussi pour empêcher le mauvais fonctionnement et l'utilisation inappropriée ou non sécuritaire, même en cas de défaillances.

Outre les appareils médicaux qui interagissent directement avec les êtres humains, d'autres appareils médicaux ou systèmes d'information médicale peuvent effectuer l'analyse ou le traitement d'informations médicales. Un logiciel défectueux pourrait entraîner le traitement erroné des données, ce qui mènerait à des décisions ou à des interventions non valides de la part des soignants. Un exemple d'un tel risque serait un système d'information médicale dans lequel un ensemble de médicaments fait l'objet d'une vérification croisée en vue de déceler les interactions dangereuses possibles. Un traitement erroné de la part de ce système logiciel pourrait se traduire par un résultat « faux négatif » (où le système indique qu'il n'y a pas d'interaction médicamenteuse dangereuse, alors qu'il y en a) ou par un résultat « faux positif » (où le système indique qu'il y a des interactions médicamenteuses dangereuses, alors qu'il n'y en a pas).

L'aspect « sécurité » de tels systèmes est évident : la défaillance ou le comportement erroné peut être préjudiciable aux patients. En raison de ce risque, la conception, la production, l'exploitation et la maintenance des dispositifs médicaux sont assujetties à des réglementations propres à l'industrie. Comme dans d'autres champs/domaines, il est essentiel que le développement logiciel suive un processus rigoureux et bien documenté.

### 4.3 Avionique

L'avionique se définit comme l'application des techniques de l'électronique au domaine de l'aviation, ce qui englobe le

matériel et les logiciels utilisés pour contrôler et activer les avions modernes et leurs systèmes. En termes de génie logiciel, la tâche consiste à écrire des algorithmes logiciels qui maintiennent la stabilité de l'avion et contrôlent la navigation, l'état des systèmes, la réaction en cas d'urgence et les systèmes environnementaux. Ces algorithmes prennent les données provenant des capteurs, des commandes du pilote, des transmissions du sol et d'autres avions, des bases de données internes, ainsi que d'autres routines logicielles qui assurent l'interface avec les contrôles et d'autres capteurs. Ces routines logicielles peuvent contrôler directement le matériel de vol, les systèmes à fonctionnement automatique et les affichages.

Dans le cas où les logiciels ont des interfaces avec des équipements, des capteurs ou des contrôles physiques, cela exige une connaissance approfondie des caractéristiques de vol et des autres systèmes de l'avion. Cela, en soi, peut être interprété comme étant des connaissances en génie et l'exercice du génie.

L'aspect « sécurité » de tels systèmes est évident : les vies de l'équipage et des passagers à bord de l'avion dépendent de l'exactitude et du bon fonctionnement – dans toutes les circonstances prévues – des systèmes et des algorithmes d'aérodynamique. En raison de ce haut niveau de risque, un ingénieur en génie logiciel titulaire d'un permis devrait concevoir le ou les systèmes et assumer la responsabilité du travail réalisé.

#### 4.4 Système de freinage antiblocage

Les systèmes de freinage antiblocage (ABS) sont conçus pour surveiller les roues d'une automobile et contrôler le freinage de manière à ce que les roues ne se bloquent pas. La plupart des systèmes ABS modernes sont informatisés et utilisent un ensemble de capteurs, des contrôles hydrauliques et un système de contrôle logiciel. Lors d'un freinage violent, le système contrôle la pression de freinage pour empêcher les roues de se bloquer.

L'aspect « sécurité » de ce système est évident : la sécurité du public est directement visée, car la vie des passagers du véhicule, celle des passagers d'autres automobiles sur la route ainsi que celle des piétons sont en cause.

Il existe aussi d'autres raisons qui font que ce travail relève de la portée exclusive du génie. Il s'agit d'un système invisible pour l'utilisateur qui exige la connaissance des caractéristiques physiques de l'automobile, du système de freinage, de l'environnement de conduite et de la sécurité. Dans cet exemple particulier, un ingénieur devra connaître le volet « logiciel » du système, ainsi que le matériel, les capteurs et les contrôles physiques. Il s'agit donc d'un projet multidisciplinaire qui nécessitera du travail de développement logiciel, d'informatique et de mécanique. Par conséquent, seul un ingénieur titulaire de permis et possédant une vaste formation est légalement autorisé à concevoir et à construire le système et à assumer la responsabilité du travail réalisé.

#### 4.5 Jeux vidéo

Le développement de jeux vidéo est une entreprise commercialement lucrative. À première vue, cependant, il est évident que ces jeux ont très peu d'impact sur le bien-être, la sécurité, la santé du public ou l'environnement. Comme il s'agit du facteur le plus important et prédominant, il est clair que le développement de jeux vidéo ne relève pas de la portée exclusive du génie logiciel.

Cependant, cela n'empêche pas l'application de principes du génie logiciel (et d'autres) au développement de ce type de logiciel, et cela n'empêche pas les ingénieurs en logiciel de participer à ce travail. Comme dans d'autres disciplines du génie, exercer à l'extérieur de la portée exclusive du génie ne dégage pas un ingénieur de l'obligation de respecter les normes d'exercice attendues de tous les ingénieurs, et ainsi, un ingénieur en logiciel travaillant dans ce domaine devrait respecter ces mêmes normes. Plus précisément, cependant, cela ne signifie pas qu'un ingénieur en logiciel qui développe un jeu vidéo doit utiliser les mêmes techniques qui seraient appliquées au développement d'un système essentiel à la sécurité, mais plutôt que l'ingénieur en logiciel doit choisir les techniques qui sont adaptées à l'usage prévu du système.

#### 4.6 Logiciel d'opérations boursières

La plupart des négociateurs boursiers modernes utilisent des logiciels automatisés pour exécuter des opérations programmées. Ces systèmes sont conçus pour surveiller les conditions du marché et, en suivant des règles prédéterminées, ils exécutent automatiquement des transactions sur le marché boursier auquel ils sont connectés. Les règles sont établies par l'utilisateur et correspondent à sa connaissance du marché.

Il est peu probable qu'un seul utilisateur de ce genre de logiciel cause des dommages sur le marché avec un logiciel défectueux, mais des milliers d'utilisateurs du même logiciel pourraient causer un effondrement du marché si leurs systèmes exécutaient simultanément des transactions erronées. Ces transactions seraient normalement effectuées à haute vitesse et ne seraient pas vérifiées avant d'être traitées. Il y a déjà eu des exemples de fluctuations boursières causées par des opérations programmées erronées. Si un effondrement du marché était déclenché par un logiciel boursier, les dommages économiques seraient considérables.

L'aspect « intérêt du public » de ces systèmes est évident : advenant un krach boursier, il y aurait un impact très négatif sur les intérêts économiques du public. Les conséquences d'un tel krach, découlant d'un système incorrectement conçu sur le plan de l'ingénierie, seraient telles que le système juridique serait dans l'impossibilité de les compenser. À lui seul, ce risque caractérise ce logiciel comme un travail devant être réalisé par un ingénieur titulaire de permis.



## 4.7 Logiciel de gestion de la paye

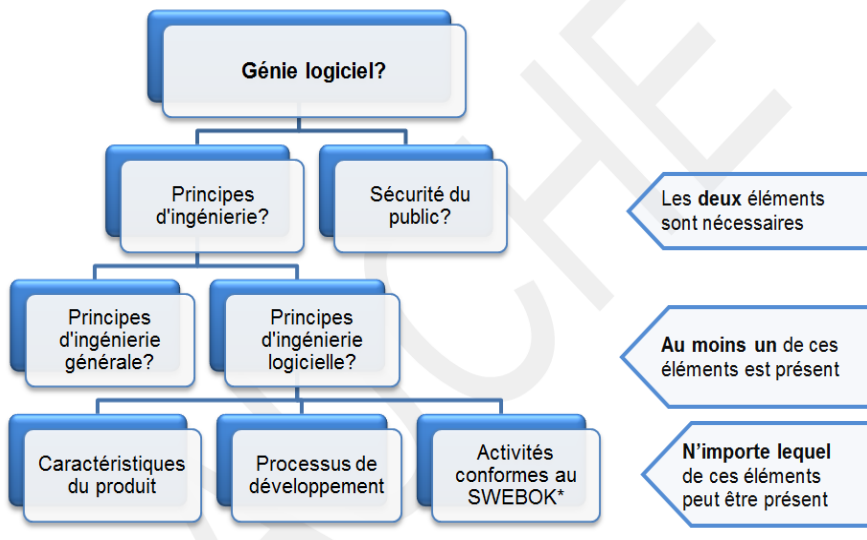
Les systèmes de paye sont utilisés par les entreprises partout dans le monde pour calculer les salaires et les impôts. Ces systèmes peuvent imprimer des chèques ou générer des transferts électroniques de fonds qui sont envoyés à la banque d'une entreprise pour exécution. Un logiciel de système de paye commercial n'est pas unique – le même logiciel peut être concédé sous licence et utilisé par des milliers d'entreprises. Une erreur dans le calcul de la paye pourrait avoir un impact négatif sur des centaines de milliers de personnes.

Ces applications logicielles sont gérées et peuvent être auditées par le client ou l'utilisateur. Avant que les chèques soient imprimés ou que le fichier de transfert électronique de fonds soit envoyé à la banque, le gestionnaire responsable ou l'opérateur du logiciel de paye peut imprimer des rapports et vérifier si les calculs effectués par l'application logicielle sont corrects. Cela transfère une partie de la responsabilité du bon fonctionnement du système du développeur du logiciel à l'opérateur du logiciel.

Cela constitue également un exemple où la protection contre un logiciel de mauvaise qualité pourrait être assurée par le système juridique. L'on peut se demander si le développement d'un logiciel destiné à un produit de comptabilité relève, ou non, de l'exercice du génie.

## 5 Conclusion

Pour déterminer si une activité ou un système relève de l'exercice du génie logiciel, on devrait se poser les questions suivantes :



\*SWEBOK : Software Engineering Body of Knowledge/Corpus de connaissances propres au génie logiciel

L'utilisation de ce cadre, en combinaison avec les caractéristiques et principes énoncés dans la section 2, permettra de mieux comprendre la portée exclusive de l'exercice du génie logiciel et soutiendra les activités d'application de la loi dans ce domaine.

## 6 Notes de fin

[1] Bureau canadien des conditions d'admission en génie, *Programme d'examens – Génie logiciel*, 2004. [https://www.engineerscanada.ca/srv/engineerscanada-website-drupal/src/sites/default/files/syllabus\\_4\\_19\\_0.pdf](https://www.engineerscanada.ca/srv/engineerscanada-website-drupal/src/sites/default/files/syllabus_4_19_0.pdf)

[2] Institute of Electrical and Electronics Engineers, *Guide to the Software Engineering Body of Knowledge (SWEBOK)*, 2004. <https://www.computer.org/web/swebok>

[3] Certaines lois, et la définition d'Ingénieurs Canada citée ci-dessous, utilisent le terme *exercice de la profession d'ingénieur* pour désigner une activité qui ne peut être exercée que par un ingénieur titulaire d'un permis ou sous sa supervision. Dans ce document, nous utilisons le terme *génie logiciel* pour désigner la pratique qui relève de la portée exclusive du génie logiciel, tandis que le terme *développement logiciel* désigne l'activité qui peut être exercée par des personnes non titulaires de permis d'exercice du génie

[4] En génie logiciel, le terme « développement » renvoie au cycle de vie complet d'un produit, comprenant la conception, la mise en œuvre, les tests et, parfois, l'installation et la maintenance (correction des défauts et amélioration des caractéristiques), conformément à [2]. Dans ce document, le terme « développement » englobe toutes ces activités.

[5] Institute of Electrical and Electronic Engineers, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Std 610.12-1990, New York, NY. <http://standards.ieee.org/findstds/standard/610.12-1990.html>

[6] Pour avoir une illustration des nombreuses façons dont les systèmes informatiques peuvent être préjudiciables au bien-être public, voir : *Forum on Risks to the Public in Computers and Related Systems* de l'Association of Computing Machinery (<http://catless.ncl.ac.uk/Risks/>).

[7] Institute of Electrical and Electronic Engineers, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Std 610.12-1990, New York, NY. <http://standards.ieee.org/findstds/standard/610.12-1990.html>

[8] Institute of Electrical and Electronic Engineers, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Std 610.12-1990, New York, NY. <http://standards.ieee.org/findstds/standard/610.12-1990.html>

[9] Bureau canadien des conditions d'admission en génie, *Programme d'examens – Génie logiciel*, 2004. [https://www.engineerscanada.ca/srv/engineerscanada-website-drupal/src/sites/default/files/syllabus\\_4\\_19\\_0.pdf](https://www.engineerscanada.ca/srv/engineerscanada-website-drupal/src/sites/default/files/syllabus_4_19_0.pdf)

[10] International Atomic Energy Agency, *Verification and validation of software related to nuclear power plant instrumentation and control*, Technical reports series no. 384, Vienna, 1999. <https://www-pub.iaea.org/books/iaeabooks/5718/Verification-and-Validation-of-Software-Related-to-Nuclear-Power-Plant-Instrumentation-and-Control>

International Atomic Energy Agency, *Software for computer based systems important to safety in nuclear power plants: safety guide*, Safety Standards Series No. NS-G-1.1, Vienna, 2000. <https://www-pub.iaea.org/books/iaeabooks/5718/Verification-and-Validation-of-Software-Related-to-Nuclear-Power-Plant-Instrumentation-and-Control>

[11] *Ibid.*

[12] *Biomedical Engineering Desk Reference*, Buddy D. Ratner et al, Academic Press, 2009, ISBN: 9780123746467

Santé Canada, *Avis – Logiciels réglementés comme des instruments médicaux de classe I ou de classe II/3* décembre 2010. [http://www.hc-sc.gc.ca/dhp-mps/md-im/activit/announce-annonce/md\\_notice\\_software\\_im\\_avis\\_logiciels-fra.php](http://www.hc-sc.gc.ca/dhp-mps/md-im/activit/announce-annonce/md_notice_software_im_avis_logiciels-fra.php)

Food and Drug Administration, *Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices*, May 11, 2005. <http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm089543.htm>