Les logiciels libres dans le secteur public

Cours destiné aux cadres du secteur public

**Version 1.0**

Nord Ouvert

Le 1er décembre 2022

**Préparé pour :**  
  
L’Institut des services axés sur les citoyens

Groupe de travail sur les logiciels libres

# Avant-propos

## À propos de ce cours

Ce matériel de formation a été conçu par Nord Ouvert, à la demande de l’Institut des services axés sur les citoyens, au nom du Conseil de la prestation des services du secteur public et du Conseil des dirigeants principaux de l’information du secteur public (conseils conjoints).

**Principaux auteurs :** Matthew Claudel (Field States), Steven Coutts (Nord Ouvert).

**Contributeurs :** Ayesha Zamudio-Vazquez (Nord Ouvert), Merlin Chatwin (Nord Ouvert), John Griffin (Nord Ouvert), Elena Findley-de Regt (Foundation for Public Code) Ben Cerveny (Foundation for Public Code).

**Nous tenons également à remercier la communauté de pratique des logiciels libres pour ses contributions :** Guillaume Charest (Service numérique canadien), Alexandre Cloutier (Agence du revenu du Canada), Olivia Courney (Services numériques de l’Ontario), Kevan Hannah (Services numériques de l’Ontario), Alan Harnum (Services numériques de l’Ontario), Katy Lalonde (Services numériques de l’Ontario), Patrick McCabe (Services numériques de la Nouvelle-Écosse), et Bianca Tomazeli (Ville de Montréal).

**Citation :** Nord Ouvert. Logiciels libres au sein de l’administration fédérale : Un cours pour les cadres du secteur public – v1.0 – Institut des services axés sur les citoyens, 2022.

Ce cours est protégé par une licence de [Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)](https://creativecommons.org/licenses/by-sa/4.0/legalcode).

Si vous créez une adaptation de cette œuvre, veuillez ajouter la clause de non-responsabilité suivante avec l’attribution :

*Ceci est une adaptation d’un document conçu par l’Institut des services axés sur les citoyens et Nord Ouvert. Les points de vue et opinions exprimés dans cette adaptation relèvent de la seule responsabilité de l’auteur ou des auteurs de l’adaptation et ne sont pas approuvés par l’Institut des services axés sur les citoyens ou Nord Ouvert.*



# Table des matières

[Avant-propos ii](#_Toc120030484)

[À propos de ce cours ii](#_Toc120030485)

[Aperçu du cours 1](#_Toc120030487)

Objectifs d'apprentissage [1](#_Toc120030488)

Aperçu et structure des modules [de cours 1](#_Toc120030489)

[Module 1 : Introduction aux logiciels libres 3](#_Toc120030490)

[Objectifs d’apprentissage 3](#_Toc120030491)

Éléments clés  [3](#_Toc120030492)

[Section 1 : De quelle façon les gouvernements utilisent-ils les logiciels? 5](#_Toc120030493)

[Section 2 : Les fondements des logiciels 7](#_Toc120030494)

[Section 3 : Avantages et défis en ce qui concerne l’utilisation des logiciels libres dans le secteur public 13](#_Toc120030495)

[Ressources supplémentaires 17](#_Toc120030496)

[Module 2 : Création d’un écosystème du logiciel libre 18](#_Toc120030497)

Objectifs d'apprentissage [18](#_Toc120030498)

Éléments clés [18](#_Toc120030499)

[Section 1 : Paysage politique et législatif 20](#_Toc120030500)

[Section 2 : Enjeux et concepts clés pour la création d’un écosystème du logiciel libre 23](#_Toc120030501)

[Section 3 : Stratégies pour la création d’un écosystème du logiciel libre 26](#_Toc120030502)

[Ressources suppplémentaires 30](#_Toc120030503)

[Module 3 : Approvisionnement, budgétisation et passation de marchés 31](#_Toc120030504)

Objectifs d'apprentissage [31](#_Toc120030506)

[Section 1 : Modèles d’octroi de licences de logiciels, de revenus et de développement 32](#_Toc120030507)

[Section 2 : Enjeux et concepts clés en matière d’approvisionnement 37](#_Toc120030508)

[Section 3 : Budgétisation et passation de marchés 44](#_Toc120030509)

Ressources supplémentaires [48](#_Toc120030510)

[Module 4 : Sécurité et mise à niveau des logiciels libres 50](#_Toc120030511)

Objectifs d'apprentissage  [50](#_Toc120030512)

Éléments clés [50](#_Toc120030513)

[Section 1 : Introduction à la sécurité et à la mise à niveau des logiciels libres 52](#_Toc120030514)

[Section 2 : Enjeux et concepts clés pour la mise à niveau et la sécurisation des logiciels libres 54](#_Toc120030515)

[Section 3 : Stratégies pour la mise à niveau et la sécurisation des logiciels libres 58](#_Toc120030516)

Ressources supplémentaires [60](#_Toc120030517)

[Module 5 : Collaboration en ce qui concerne les logiciels libres 61](#_Toc120030518)

Objectifs d'apprentissage [61](#_Toc120030519)

[Éléments clés 61](#_Toc120030520)

[Section 1 : Collaboration entre administrations 62](#_Toc120030521)

[Section 2 : Stratégies de collaboration en ce qui concerne les logiciels libres 66](#_Toc120030522)

Ressources supplémentaires [69](#_Toc120030523)

[Glossaire 70](#_Toc120030524)

[Annexe A : Documents d’une page à l’intention de la haute direction 75](#_Toc120030525)

Document no [1 : Pourquoi un logiciel libre? 76](#_Toc120030526)

Document no [2 : Création d’un écosystème du logiciel libre 78](#_Toc120030529)

Document no [3 : Approvisionnement, budgétisation et passation de marchés 80](#_Toc120030532)

Document no [4 : Sécurité et mise à niveau des logiciels libres 81](#_Toc120030535)

Document no [5 : Collaboration en ce qui concerne les logiciels libres 82](#_Toc120030538)

[Annexe B : Mise en œuvre du cours 83](#_Toc120030541)

# 

# Aperçu du cours

Les dirigeants et les cadres supérieurs du secteur public ont la possibilité de mener la transformation numérique de la prestation de services essentiels et des opérations au sein de leur organisation en adoptant des logiciels libres. Pour ce faire, ils doivent :

* Réorienter la culture organisationnelle de façon à ce qu’elle devienne favorable aux logiciels libres.
* Investir dans le renforcement des capacités du personnel.
* Promouvoir des approches numériques en matière d’approvisionnement, de budgétisation et de passation de marchés qui favorisent l’adoption de logiciels libres.
* Comprendre les différences fondamentales entre les logiciels propriétaires conventionnels et les logiciels libres.
* Tirer parti du potentiel de collaboration interjuridictionnelle pour créer et appuyer des applications logicielles à code source ouvert.

Ce cours vise à fournir aux cadres du secteur public et à leurs collègues les idées, les outils et les ressources dont ils ont besoin pour surmonter les difficultés souvent associées à l’utilisation de logiciels libres – et pour profiter pleinement de leurs avantages. Ce cours a été conçu principalement du point de vue des cadres, directeurs et dirigeants ministériels œuvrant dans un domaine non technique, mais il renferme également des renseignements importants pour le personnel technique.

## Objectifs d’apprentissage du cours

Ce cours permettra aux participants :

1. D’acquérir une connaissance de base des logiciels libres, de la différence par rapport aux logiciels propriétaires conventionnels, et de la valeur que les logiciels libres apportent dans un contexte public.
2. D’apprendre à reconnaître les obstacles courants à l’utilisation des logiciels libres dans le secteur public et d’élaborer des stratégies pour les surmonter.
3. De comprendre comment un cadre opérationnel appliqué pour les décisions de politique et de gestion peut permettre une approche équilibrée en ce qui concerne les logiciels – qu’il s’agisse de logiciels ouverts ou de logiciels propriétaires conventionnels.

## Aperçu et structure des modules de cours

Le contenu a été divisé en cinq modules qui traitent chacun d’aspects spécifiques de la mise en œuvre, de l’utilisation et de l’exploitation des logiciels libres dans le secteur public :

* Les modules 1, 2 et 5 donnent une vue d’ensemble des logiciels libres et de la différence entre ces logiciels et les logiciels propriétaires, des facteurs organisationnels susceptibles de favoriser ou d’entraver leur adoption et leur utilisation dans le secteur public, et de ce que signifie la participation à l’écosystème plus vaste des logiciels libres.
* Le module 3 fournit des renseignements détaillés sur les considérations relatives à l’approvisionnement, à la budgétisation et à la passation de marchés en ce qui concerne les logiciels libres. Le contenu de ce module devrait intéresser le personnel de la fonction publique œuvrant dans les domaines de l’**approvisionnement,** des **finances** ou du **droit**.
* Le module 4 fournit des renseignements détaillés sur les différents aspects de la sécurité et de la mise à niveau des logiciels libres.

Chaque module contient des points clés et des sous-thèmes et fait état des défis connexes, de stratégies et de ressources supplémentaires pour ceux qui souhaitent en apprendre davantage.

Chaque module est également accompagné d’un **résumé** général d’une page et des points clés de chaque module. Les employés peuvent utiliser ces documents pour communiquer des messages clés à leurs cadres supérieurs ou aux membres de la haute direction et les aider à défendre – et à mettre en œuvre – les logiciels libres. Le didacticiel permettra de prendre des décisions sur l’intégration et l’harmonisation de l’utilisation des logiciels libres avec les priorités et les engagements de l’organisation.

# Module 1 : Introduction aux logiciels libres

## Objectifs d’apprentissage

Ce module vous permettra de vous familiariser avec :

* Les cas d’utilisation générale des logiciels dans le secteur public et les raisons pour lesquelles les logiciels gouvernementaux sont uniques.
* Les principales différences entre les logiciels libres et les logiciels propriétaires en explorant les origines et les motivations du mouvement des logiciels libres, ainsi que les principes fondamentaux du fonctionnement actuel des logiciels libres.
* Les principes fondamentaux de l’obtention et de la mise à niveau des logiciels libres, les avantages de l’utilisation des logiciels libres dans le secteur public, les défis uniques qui y sont associés et les stratégies permettant de surmonter ces défis – stratégies qui seront décrites plus en détail tout au long du cours.

## Éléments clés

* **Les gouvernements et les organismes du secteur public du Canada et du monde entier utilisent déjà les logiciels libres dans une large mesure.** Alors que tous les ordres de gouvernements intègrent des logiciels dans une grande variété de processus et de services pour atteindre leurs objectifs – de la prestation efficace de services publics aux objectifs sociaux et environnementaux – les logiciels libres constituent une excellente option pour garantir la qualité, le contrôle permanent et l’efficacité des ressources.
* **Les logiciels libres sont foncièrement différents des logiciels propriétaires conventionnels** à plusieurs égards : La manière dont ils sont concédés sous licence, détenus et gérés, la manière dont ils sont conçus et mis à niveau, et la manière dont ils sont financés et commercialisés. Un logiciel libre est distribué gratuitement en vertu d’une licence libre. De nombreux projets de logiciels libres sont réalisés par une collectivité mondiale d’utilisateurs et de concepteurs, alors que d’autres ont une base de contributeurs plus restreinte. Le logiciel proprement dit est gratuit, mais il se peut que votre organisation doive investir dans les capacités du personnel interne pour assurer l’intégration et la mise à niveau du logiciel, ou encore passer un contrat avec un fournisseur pour ajouter des fonctionnalités spécifiques ou faire une intégration personnalisée en fonction de votre environnement numérique existant.
* **Les logiciels libres ne doivent pas nécessairement être conçus à partir de rien.** Des centaines de projets complets de logiciels libres et encore plus de petits modules de composantes de logiciels libres offrent autant d’options et de fonctionnalités que les logiciels propriétaires conventionnels. Ces projets et modules servent de base à la création de nouveaux logiciels libres.
* Les **logiciels libres peuvent présenter une variété d’avantages uniques pour le secteur public –** notamment la personnalisation, l’interopérabilité et la transparence. Bien qu’il existe des obstacles à l’adoption des logiciels libres dans le secteur public – notamment l’absence de capacités techniques internes et l’application de pratiques d’approvisionnement rigides – il existe heureusement une variété de stratégies et de pratiques exemplaires pouvant vous aider à surmonter ces obstacles.

## Section 1 : De quelle façon les gouvernements utilisent-ils les logiciels?

### Les logiciels au sein du gouvernement : Cas d’utilisation

Les gouvernements utilisent des logiciels à des fins administratives, financières et opérationnelles très variées, qu’il s’agisse de services destinés au public, de la gestion des marchés de l’État ou du courrier électronique sécurisé. Ces cas d’utilisation incluent, sans toutefois s’y limiter, les exemples ci-après :

* Prestation de services directement aux citoyens (délivrance de permis et de licences, suivi des questions citoyennes).
* Traitement des droits et paiements (impôts fonciers, amendes, etc.).
* Renseignements opérationnels (suivi du rendement des unités opérationnelles, des programmes et des services).
* Gestion des données (Portail de données ouvertes).
* Gestion des ressources humaines (administration de la paie et des avantages sociaux, acquisition de talents).
* Gestion des actifs (suivi de l’équipement et de l’infrastructure tout au long de leur cycle de vie, modélisation des scénarios de risques et de coûts).
* Gestion des achats (lancement d’appels d’offres, gestion des relations avec les fournisseurs).
* Communications (gestion du contenu, gestion de l’ordre du jour et des réunions, services d’alerte d’urgence).

### Unicité des logiciels gouvernementaux

Le contexte gouvernemental présente des exigences uniques en matière de logiciels. Le processus d’obtention et de mise à niveau des logiciels – et la conception des logiciels proprement dits – présente des caractéristiques particulières dans le secteur public. Le gouvernement exige des logiciels qui :

* sont très fiables;
* sont suffisamment sécuritaires pour stocker des données sensibles ou des renseignements permettant d’identifier des personnes;
* peuvent être obtenus conformément à la législation sur les marchés de l’État;
* peuvent être budgétisés en amont, généralement des années à l’avance (comme pour les achats gouvernementaux en général);
* peuvent être intégrés aux systèmes informatiques et aux environnements numériques existants;
* respectent la réglementation et les normes propres à une administration, notamment les exigences en matière de résidence des données (c’est-à-dire l’hébergement des données sur des serveurs locaux par opposition au stockage basé sur l’informatique en nuage à l’extérieur de l’administration) ou le respect des exigences en matière de langue et d’accessibilité..

## Section 2 : Les fondements des logiciels

### Les fondements des logiciels : Concepts clés et terminologie

Tout au long de ce cours, nous aborderons les différents aspects de l’achat, de l’utilisation et de la mise à niveau des logiciels, et nous accorderons une attention particulière au contexte du secteur public. Quelques concepts fondamentaux sous-tendent cette discussion :

* Le **développement** désigne la *manière dont les logiciels sont mis au point.* Les logiciels sont constitués de lignes de code. Ils peuvent être conçus à partir de rien, mais le plus souvent, ils sont composés de modules préexistants (qui remplissent des fonctions spécifiques, comme l’authentification de la connexion) et ces modules peuvent eux‑mêmes être libres. Dans le cadre de ce cours, le développement englobe également la mise à jour et la mise à niveau continues des logiciels.
* L’**intégration** désigne la mise en œuvre dans un contexte particulier. Il est rare que les logiciels fonctionnent correctement dès le début. Selon les besoins de votre organisation, la mise en œuvre nécessitera des fonctionnalités personnalisées et devra toujours être intégrée à votre environnement numérique existant.
* La **distribution** désigne la *manière dont le logiciel parvient à l’utilisateur final*. Cela inclut la licence légale d’utilisation du logiciel (logiciel propriétaire ou logiciel libre), ainsi que la méthode de distribution (informatique en nuage, hybride ou sur place).
* Les **revenus** désignent *qui paie qui et pourquoi*. Il peut s’agir d’un logiciel en tant que service (SaaS), d’un déploiement sur place d’un logiciel propriétaire, ou d’une rémunération à l’acte pour créer une mise en œuvre personnalisée d’un logiciel libre.
* Les **clients,** les **vendeurs,** les **entrepreneurs et** les **utilisateurs** sont les *principaux intervenants d’un écosystème logiciel*. Les liens entre les intervenants découlent des modèles de développement, de distribution et de revenus.

### Modèles de licences de logiciels

N’importe qui peut mettre au point un code logiciel et appliquer la licence de son choix. Une licence détermine qui peut utiliser le logiciel, comment il peut l’utiliser, s’il peut le modifier, etc. Il existe deux principaux modèles de licence pour les logiciels :

* Les licences **de logiciels propriétaires** font référence aux logiciels dont le code source appartient légalement à la personne ou à la société qui les a créés. Cette entité a le droit unique et exclusif de modifier et de commercialiser le code comme bon lui semble.
  + Exemples : Microsoft Windows et Apple OS X (systèmes d’exploitation), Microsoft Office (logiciels de productivité), Adobe Creative Suite (logiciel de conception), SAP ERP (progiciel de gestion intégrée), ArcGIS (système d’information géographique).
* Les licences de **logiciels ouvertes** font référence à des logiciels auxquels tout le monde peut utiliser ou modifier le code source. Le code source est mis à disposition en ligne et les contributions sont gérées à l’aide d’une plateforme comme GitHub.
  + Exemples : Linux (système d’exploitation), Firefox (navigateur web), Python (langage de programmation), Open Office (logiciels de productivité), Thunderbird (client de messagerie), Apache (serveur web), QGIS (système d’information géographique).

### Fonctionnement des logiciels propriétaires

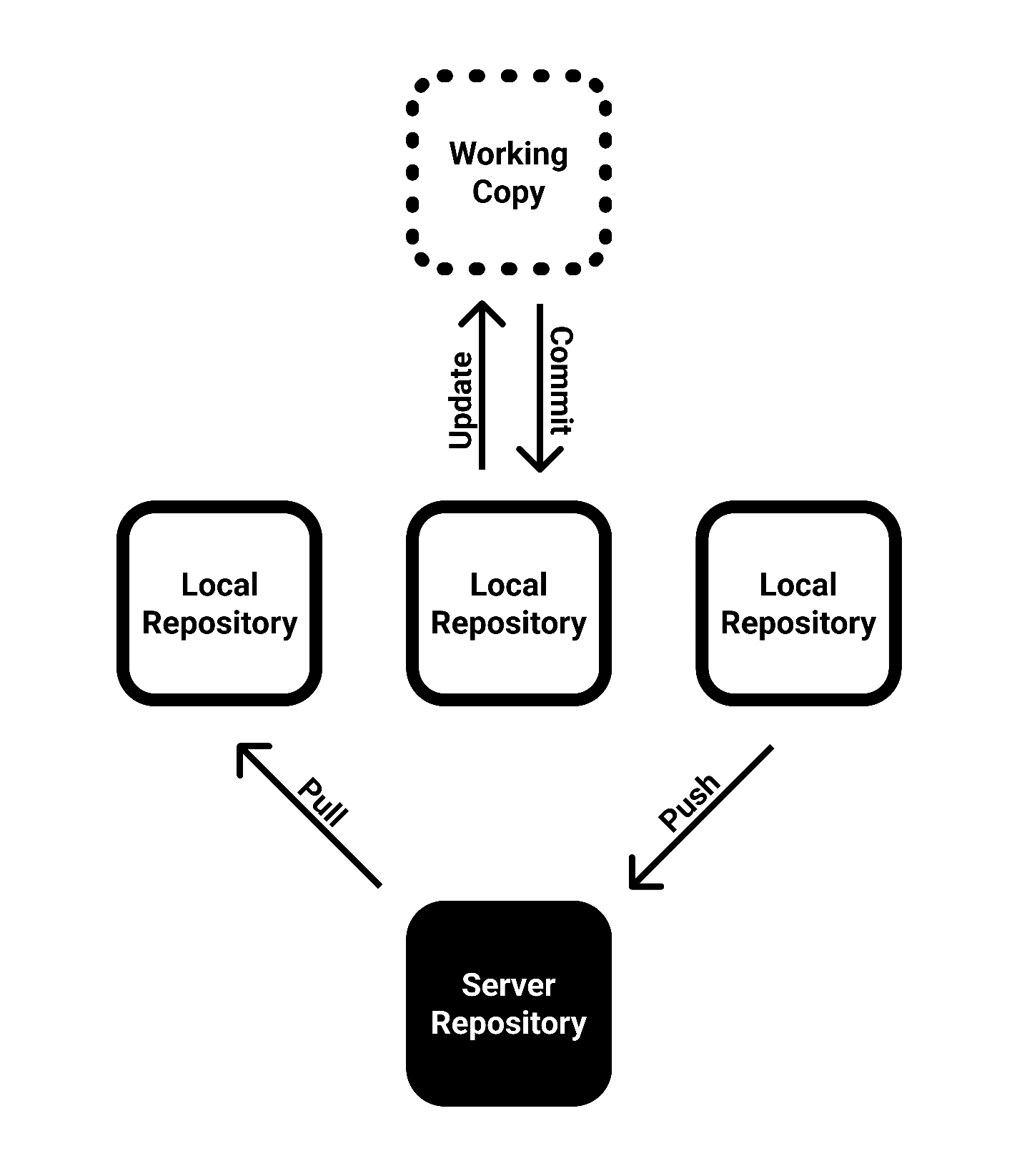
Il est plus que probable que vous utilisez déjà des logiciels propriétaires au quotidien et que vous avez une connaissance de base de leur fonctionnement. Voici un résumé des principales étapes de la chaîne de valeur des logiciels propriétaires.

* **Développement**: Les vendeurs de logiciels créent et commercialisent des logiciels. Ils fournissent généralement des mises à jour de base (ou des « rustines ») des logiciels afin qu’ils restent exempts de bogues et qu’ils soient compatibles avec des environnements numériques en évolution (les mises à jour peuvent être gratuites ou payantes).
* **Distribution**: L’organisation qui a créé le logiciel conserve la propriété des droits d’accès, d’utilisation et de commercialisation du code source par l’entremise d’une licence juridiquement contraignante. Les logiciels peuvent être hébergés dans l’informatique en nuage (en ligne), les utilisateurs payant pour y accéder, ou hébergés sur place (sur des serveurs locaux).
* **Intégration :** Les éditeurs de logiciels proposent généralement une assistance, mais ils peuvent facturer des frais. Certains fournisseurs de logiciels proposent des intégrations personnalisées en fonction de l’environnement numérique existant de votre organisation; dans le cas contraire, vous devrez faire appel à un autre fournisseur de services.
* **Revenus**: Les utilisateurs finaux paient pour utiliser le logiciel, selon un mode de paiement (montant forfaitaire, montant basé sur l’utilisation, paiement à plusieurs niveaux, freemium, etc.). Les éditeurs de logiciels peuvent publier des rustines ou offrir de nouvelles fonctionnalités, des mises à niveau et des intégrations qui sont inclus dans la structure tarifaire ou pour lesquels les utilisateurs finaux paient des frais supplémentaires.

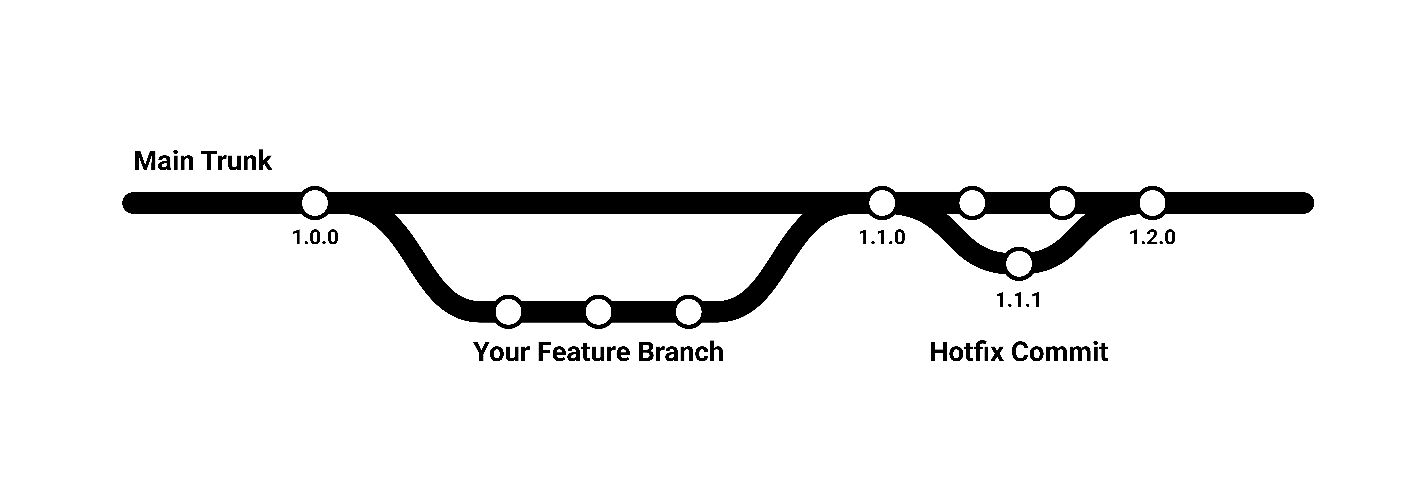
### Fonctionnement des logiciels libres

Les logiciels libres fonctionnent d’une manière fondamentalement différente des logiciels propriétaires.

* **Développement :** Un concepteur crée la première version d’un logiciel libre et la télécharge sur un dépôt public en ligne sous une licence libre. Le code est ensuite maintenu à l’aide d’un **système de contrôle de la version distribuée** (Git étant le plus populaire) dans lequel chaque utilisateur dispose d’une copie de travail et de l’historique complet des modifications d’une base de code donnée (voir la figure 1.1). GitHub, le plus grand service d’hébergement de ces dépôts, fournit un environnement fiable dans lequel une collectivité mondiale peut gérer, harmoniser et concilier des versions simultanées (possiblement contradictoires) du logiciel, identifier et corriger les bogues, et rationaliser le processus d’édition.
  + Il existe plusieurs modèles de développement de logiciels libres, chacun ayant une approche légèrement différente de la gestion des fusions avec la base de code principale et de leur fréquence. Dans la plupart des cas, ces modèles suivent une structure ramifiée, dans laquelle la base de code principale fait office de « tronc » et les « branches » sont créées lorsque des particuliers ou des organisations créent des copies – ou « bifurcations » - et ajoutent leurs propres fonctionnalités (voir la figure 1.2). De cette façon, les logiciels libres s’améliorent continuellement grâce aux contributions d’une collectivité ouverte.
  + Certains logiciels libres disposent d’une structure de gouvernance pour l’examen et la validation de toute modification du code source. Cela garantit l’utilité et l’intégrité des contributions. Les modifications validées sont réintégrées dans le tronc principal de l’arbre.



**Figure 1.1 :** Ce diagramme montre de manière simplifiée comment fonctionne un système de contrôle de la version distribuée. Le dépôt d’un serveur principal stocke tous les fichiers associés à un projet de code source. Chaque organisation collaborant au projet maintient un référentiel local qui est une copie du référentiel principal. Chaque programmeur conserve une copie de travail de son référentiel local respectif (qui est lui-même une copie du référentiel principal) sur son disque dur. Les programmeurs peuvent apporter des modifications et mettre à jour leur référentiel local sans aucune interférence. Ils peuvent mettre à jour leurs référentiels locaux avec de nouvelles données provenant du référentiel du serveur principal par une opération appelée « pull » (extraction) et apporter des modifications au référentiel principal par une opération appelée « push » (insertion), à partir de leur référentiel local.



**Figure 1.2**: Un modèle de développement simple et ramifié pour les logiciels libres. Le tronc principal représente le code source « original ». Toute personne ou organisation peut créer une copie – ou « bifurcation » - et ajouter ses propres fonctionnalités. Les branches « Hotfix » sont utilisées pour corriger rapidement les versions de production.

* **Distribution :** Une licence de logiciel libre accorde à quiconque le droit d’utiliser, d’étudier, de modifier et de distribuer le logiciel et son code source à quiconque et dans n’importe quel but (souvent avec attribution).
  + Quel que soit le mode de développement du logiciel, il peut être publié sous une licence de logiciel libre. Les défis et les opportunités des logiciels libres sont plus spécifiques et uniques une fois qu’ils ont été publiés sous une licence libre, entrant dans la phase de mise à niveau et d’adaptation.
* **Intégration :** À tout moment, un utilisateur peut décider que ses besoins sont suffisamment uniques pour mériter une version entièrement personnalisée du logiciel. Il peut alors créer une copie du code source et la modifier comme bon lui semble, sans réintégrer les modifications dans le code source principal. Cette opération est connue sous le nom de « hard fork » (changement de protocole), car elle ouvre une nouvelle voie, déconnectée du code source original.
* **Revenu :** Les utilisateurs finaux ne paient pas pour avoir accès au code source. Toutefois, cela ne signifie pas que les logiciels libres sont gratuits. L’adaptation, l’hébergement et la mise à niveau des logiciels entraînent des coûts. Ces coûts peuvent être consacrés au personnel interne ou à l’embauche d’un prestataire de services techniques externe.

#### Encadré 1.1 : Origines du mouvement des logiciels libres

Le mouvement des logiciels libres a été inspiré par la possibilité de rassembler une collectivité mondiale pour créer, développer et adapter des logiciels librement et en collaboration.

* Les logiciels « libres » ne sont pas des logiciels « gratuits ». En anglais, le mot « free » peut signifier « gratuit » et/ou « assorti de peu ou pas de restriction » (libre).
* Dans les années 1990, les concepteurs de logiciels ont commencé à offrir leurs logiciels gratuitement parce que le coût marginal de chaque unité était nul. Dans un tel contexte, le logiciel est offert gratuitement, mais le concepteur conserve les droits afférents au logiciel.
* Le mouvement des logiciels libres est apparu peu après. Ce mouvement a été important parce qu’il misait sur la possibilité d’accès au code et de modification de celui-ci, et pas simplement sur la gratuité de l’utilisation des logiciels. Les logiciels seraient libres, de sorte que d’autres concepteurs pourraient non seulement les utiliser, mais également les modifier et les améliorer.
* Dans un essai emblématique, l’un des pères du mouvement des logiciels libres, Lawrence Lessig, a lancé l’expression « pensez à la liberté d’expression par opposition à la bière gratuite » 2: les logiciels gratuits sont comme la bière gratuite, alors tandis que les logiciels libres se comparent à la liberté d’expression (accès et reproduction sans restriction).

1 Eric S. Raymond, [*The Cathedral and the Bazaar*](file:///C:\Users\Admin\Documents\R&R%20textes\INTLCMD284498087), Knowledge Technology & Policy 12, nº 3 (septembre 1999) : 23-49.

2 Lawrence Lessig, [*Free, as in Beer*](https://www.wired.com/2006/09/free-as-in-beer/), WIRED, le 1er septembre 2006.

## Section 3 : Avantages et défis en ce qui concerne l’utilisation de logiciels libres dans le secteur public

### Avantages

Neuf facteurs font en sorte que l’utilisation des logiciels libres est idéale dans un contexte gouvernemental :

1. **Ouverture et transparence**. Les logiciels libres sont justement ouverts. Ce n’est pas une boîte noire. Les collectivités mondiales et locales sont en mesure de vérifier le code, d’évaluer les implications de sa conception et de se tenir mutuellement responsables.
2. **Interopérabilité accrue**. Les logiciels libres sont généralement conçus à partir de modules de sources ouvertes, ce qui signifie qu’un logiciel libre est susceptible d’utiliser les mêmes sous-modules (ou des sous-modules compatibles) qu’un autre. Dans le cas des logiciels propriétaires, les vendeurs sont incités financièrement à créer des dépendances avec leurs propres logiciels propriétaires ou ceux de leurs partenaires. Dans le cas des logiciels libres, il y a une incitation à l’interopérabilité, car celle-ci augmente l’efficacité et la réplication.
3. **Soutien de produits décentralisé, dirigé par la collectivité**. Les logiciels libres reposent sur des mécanismes différents pour la mise à jour et la mise à niveau des logiciels, la recherche de bogues et leur correction, mais ils sont en définitive tout aussi sûrs que les logiciels propriétaires. Comme le code est ouvert, les contrôleurs de la sécurité et les contributeurs peuvent inspecter le code pour déterminer s’il présente des risques ou des failles de sécurité. Les projets de logiciels libres appuyés par des collectivités actives sont constamment mis à jour, ce qui garantit le rendement et la cybersécurité dans des conditions changeantes.
4. **Coûts contrôlés**. Un logiciel libre est gratuit, et vous avez un contrôle direct sur tous les coûts associés à ce logiciel – que vous embauchiez du personnel en interne ou que vous passiez un contrat avec des prestataires de services externes. Il n’y a pas de frais cachés, de mise à niveau coûteuse, de fonctionnalités payantes ou de renégociations de contrat.
5. **Renforcement de l’économie technologique locale**. Les fournisseurs locaux de logiciels sont bien placés pour créer des fonctionnalités ou des intégrations personnalisées pour les organisations gouvernementales, car ils établissent des relations à long terme et comprennent les priorités locales. En revanche, les logiciels propriétaires sont généralement créés et commercialisés par des fournisseurs de logiciels monolithiques qui n’ont aucun lien avec le contexte local. En embauchant des entreprises locales, les gouvernements peuvent renforcer l’économie locale, en particulier son secteur technologique.
6. **Pas de dépendance vis-à-vis du fournisseur**. Les logiciels propriétaires peuvent être ancrés dans une organisation de manière opérationnelle ou contractuelle, ce qui désavantage le gouvernement lorsqu’il prend des décisions par la suite. Les fournisseurs de logiciels propriétaires sont incités à créer des dépendances, ce qui rend difficile le choix libre et équitable de la meilleure solution. Les logiciels libres créent des marchés ouverts pour les fournisseurs de tous les types de support.[[1]](#footnote-1)
7. **Adaptabilité en fonction des besoins**. La politique et la réglementation changent constamment (par exemple, les normes d’accessibilité ou les protocoles relatifs aux renseignements personnels identifiables). Les fournisseurs de logiciels propriétaires peuvent ou non publier des mises à jour pour répondre à l’évolution des besoins. Les mises à jour sont coûteuses et peuvent nécessiter la renégociation des contrats. Les logiciels libres, en revanche, peuvent être rapidement adaptés et partagés entre administrations.
8. **Personnalisation en fonction du contexte local et des cas d’utilisation**. Si de nombreuses organisations gouvernementales remplissent des fonctions similaires, la manière dont ces fonctions sont exercées peut varier d’un endroit à l’autre. Les gouvernements peuvent répondre à des cas d’utilisation locale très spécialisés en développant leur propre logiciel libre ou en se regroupant avec d’autres gouvernements pour partager un ensemble de modules de logiciels libres et en payant chacun une petite somme pour qu’un entrepreneur local crée des fonctionnalités personnalisées pour leur environnement unique.
9. **Réduction de la responsabilité**. Les organisations gouvernementales peuvent utiliser les logiciels libres comme bon leur semble (sous réserve des lois et politiques applicables), sans obligation ni restriction supplémentaire imposée par un fournisseur. En outre, les logiciels libres ne sont pas soumis aux risques juridiques associés aux solutions propriétaires, notamment ceux qui découlent des contrats.

### Défis

L’utilisation des logiciels libres dans un contexte gouvernemental présente cinq grands défis. Par contre, ces défis peuvent être surmontés – et ce cours vous proposera des stratégies et des outils pour ce faire.

1. **Approvisionnement et budgétisation**. Les logiciels libres ont une structure de coûts très différente. Le logiciel proprement dit est gratuit, mais il y a des coûts associés à la personnalisation et à l’intégration. Cela signifie que l’on passe de l’achat d’un produit à la conception d’une solution. Il y a souvent un manque de financement pour le développement de nouveaux logiciels, et une résistance à l’idée de les offrir ensuite gratuitement à d’autres. Les normes de passation de marchés, les modèles de contrats et les listes de fournisseurs agréés créent des obstacles à l’obtention et à l’utilisation de logiciels libres ou à la conclusion de contrats avec des fournisseurs de services en ce qui concerne les logiciels libres.
2. **Expertise interne**. L’enseignement et la formation que reçoivent la plupart des professionnels du secteur public privilégient toujours les logiciels propriétaires, et les systèmes informatiques sont souvent préchargés de logiciels propriétaires. Pour ces raisons, et bien d’autres encore, les organismes du secteur public s’appuient traditionnellement sur des logiciels prêts à l’emploi proposés par les principaux fournisseurs, au lieu de recruter et de conserver des compétences en interne. Le personnel est souvent traditionnellement formé à l’utilisation de logiciels propriétaires. L’introduction de nouvelles solutions logicielles libres peut donc nécessiter un recyclage.
3. **Disponibilité des fournisseurs**. Les petites municipalités et les zones rurales ne possèdent pas nécessairement un écosystème local mature d’entreprises technologiques capables de fournir des services. Et, bien que plusieurs organismes travaillent à travers le Canada et servent un large éventail de clients, il peut s’avérer difficile d’en identifier un qui puisse fournir des services répondant aux besoins de votre organisation.
4. **Taux de changement**. Comme les logiciels libres sont souvent développés et maintenus par une collectivité mondiale, ils peuvent avoir des taux de mise à jour et de changement plus rapides ou moins rapides. Si les mises à jour dont vous avez besoin n’émergent pas de la collectivité, il se peut que votre organisation doive procéder à ces changements de manière proactive, ou faire appel à une entreprise qui peut le faire. Lorsque vous envisagez d’utiliser un logiciel libre, il est important d’évaluer le niveau d’activité de la collectivité de soutien et de mettre au point un plan d’urgence au cas où le logiciel en question ne bénéficierait plus d’un soutien adéquat et où vous devriez trouver une solution de rechange.
5. **Gouvernance**. Les logiciels libres nécessitent une participation active de la part des contributeurs et des utilisateurs de différentes organisations, ce qui peut créer des difficultés pour gérer la responsabilité de la mise à niveau.

### Résumé

Ce premier module vous a permis de vous familiariser avec les fondements des logiciels libres. Vous devriez avoir une meilleure compréhension de la fonctionnalité et de l’historique des logiciels libres, ainsi que des avantages et défis en ce qui concerne l’utilisation des logiciels libres dans un contexte gouvernemental.

Les modules suivants vous permettront d’examiner de plus près plusieurs aspects différents de l’obtention, de l’utilisation et de la mise à niveau des logiciels libres dans un contexte gouvernemental. Ils vous fourniront des outils et des stratégies pour bien faire ce travail.

[Module 2 : Création d’un écosystème du logiciel libre](#_Module_2:_Building).

[Module 3 : Approvisionnement, budgétisation et passation de marchés](#_Module_3:_Procurement,).

[Module 4 : Sécurité et mise à niveau des logiciels libres](#_Module_4:_Open).

[Module 5 : Collaboration en ce qui concerne les logiciels libres](#_Module_5:_Collaborating).

## Ressources supplémentaires

[**Qui utilise GitHub?**](https://government.github.com/community/) **:** Une liste de centaines d’organismes gouvernementaux aux niveaux national, infranational et municipal qui utilisent GitHub pour l’échange et la collaboration en ce qui concerne des projets de logiciels libres.

Le [**Linux Professional Institute (LPI)**](https://www.lpi.org/) est un organisme à but non lucratif qui propose des formations et des accréditations, et qui appuie les collectivités de logiciels libres.

[**Les Normes relatives au numérique du gouvernement du Canada : Directives**](https://www.canada.ca/fr/gouvernement/systeme/gouvernement-numerique/normes-numeriques-gouvernement-canada.html) « constituent le fondement du virage du gouvernement vers une plus grande souplesse, une plus grande ouverture et une plus grande attention sur l’utilisateur. Elles guideront les équipes dans la conception de services numériques, d’une façon qui servira le mieux les Canadiens. » Il s’agit de l’outil idéal pour vous familiariser avec les logiciels libres.

[**L’Échange de ressources ouvertes**](https://code.ouvert.canada.ca/fr/index.html) présente un catalogue consultable de logiciels libres, ainsi que de projets source ouverts, de normes ouvertes, de partenariats et de design libres utilisés par différents ordres de gouvernements au Canada.

[**Open Source Basics (Intel Software)**](https://www.youtube.com/watch?v=Tyd0FO0tko8): Une vidéo d’introduction expliquant les logiciels libres.

[**What is Open Source explained in LEGO (Socialsquare)**](https://www.youtube.com/watch?v=a8fHgx9mE5U): Une autre vidéo d’introduction expliquant les logiciels libres (cette fois-ci en utilisant LEGO).

[**The mind behind Linux:**](https://www.youtube.com/watch?v=o8NPllzkFhE)[**Linus Torvalds (TED)**](https://www.youtube.com/watch?v=o8NPllzkFhE): Un entretien avec Linus Torvalds, qui a transformé la technologie à deux reprises – initialement avec le noyau Linux, qui contribue à alimenter l’internet, puis avec Git, le système de gestion de code source utilisé par les développeurs du monde entier.

# Module 2 : Création d’un écosystème du logiciel libre

## Objectifs d’apprentissage

Ce module vous permettra de vous familiariser avec :

* Les principaux facteurs favorables et les obstacles à l’adoption des logiciels libres, liés à la dotation en personnel et à la structure organisationnelle.
* La grande variété de fonctions, de profils professionnels et d’ensembles de compétences nécessaires pour le développement, la mise en œuvre et la mise à niveau de projets de logiciels libres au sein d’une organisation gouvernementale.
* Les pratiques exemplaires pour constituer et diriger une équipe technique de taille adéquate.

## Éléments clés

* **Le paysage politique et législatif appuie de plus en plus l’adoption des logiciels libres.** Cependant, les structures, les pratiques et les mentalités actuelles au sein de nombreux organismes gouvernementaux et du secteur public en général peuvent donner lieu à des obstacles lorsqu’il s’agit d’adopter des logiciels libres.
* **La dotation en personnel technique est l’un des principaux obstacles perçus en ce qui concerne l’adoption de logiciels libres.** Un manque de capacités internes et de culture de collaboration peut conduire à une dépendance excessive à l’égard des fournisseurs : Peu d’organisations gouvernementales disposent d’un personnel à temps plein (aux niveaux de la direction et de la mise en œuvre) ayant les compétences et les connaissances nécessaires pour acquérir, développer et/ou mettre à niveau des logiciels libres. Par conséquent, les gouvernements s’en remettent généralement à des fournisseurs.[[2]](#footnote-2) Cela peut entraîner l’enfermement du fournisseur, l’explosion des coûts et des frais cachés, ainsi que la perte de contrôle (sur les fonctionnalités, les mises à jour et la conformité).[[3]](#footnote-3)
* Les **dirigeants d’organisations de toutes tailles peuvent relever ces défis en réalisant des investissements stratégiques, en réduisant les risques réels et perçus et en encourageant un changement de culture organisationnelle**. Cela comprend :
  + Apprendre comment fonctionnent l’acquisition, le développement et la mise à niveau des logiciels libres et appuyer les groupes de travail et les champions internes.
  + Favoriser une culture organisationnelle qui encourage la transparence et l’engagement continu des utilisateurs.
  + Mobiliser une variété de partenaires, d’organisations homologues et de prestataires de services pour compléter l’équipe interne.

## Section 1 : Paysage politique et législatif

Le paysage politique et législatif appuie de plus en plus l’adoption des logiciels libres aux niveaux fédéral, provincial, territorial et municipal. Certaines lois et politiques prévoient directement l’utilisation des logiciels libres, alors que d’autres prévoient indirectement l’utilisation de tels logiciels.

### Les logiciels libres au niveau fédéral

De nombreuses politiques fédérales au niveau fédéral qui définissent l’approche du gouvernement du Canada en matière de logiciels libres.

* Le gouvernement du Canada a comme mandat de publier ses propres codes sources sous des licences Open Source tant que cela est compatible avec les principes fondamentaux du droit administratif tels que la transparence, la responsabilité, la légalité et l’équité procédurale.[[4]](#footnote-4)
* En 2018, le gouvernement du Canada a adopté la [Charte internationale sur les données ouvertes](https://opendatacharter.net/principles-fr/) pour démontrer son engagement en ce qui concerne les données ouvertes. La Charte n’est pas un instrument juridiquement contraignant, mais son principe de données « ouvertes par défaut » a permis de jeter bases de l’utilisation et du développement de logiciels libres.
* La [Directive sur les services et le numérique](https://www.tbs-sct.canada.ca/pol/doc-fra.aspx?id=32601) et les [Normes relatives au numérique du gouvernement du Canada](https://www.canada.ca/fr/gouvernement/systeme/gouvernement-numerique/normes-numeriques-gouvernement-canada.html) : Directives constituent des documents d’orientation supplémentaires pour les fonctionnaires fédéraux.

### Les logiciels libres au niveau provincial

Les provinces ont un degré de contrôle important sur les logiciels et services spécifiques qu’elles utilisent, ainsi que sur l’interprétation et la mise en œuvre des politiques fédérales, sous la forme d’une politique provinciale en matière de données.

* Les [Normes des services numériques](https://www.ontario.ca/fr/page/norme-des-services-numeriques) de l’Ontario demande aux ministères et organismes provinciaux d’utiliser « des normes ouvertes, des logiciels ouverts et des plateformes gouvernementales communes, dans la mesure du possible, », et d’accorder « la préférence à des outils ouverts qui sont accessibles et appuyés par une forte collectivité de développeurs ».[[5]](#footnote-5)
* La province de la Colombie-Britannique a publié des [directives sur l’utilisation des logiciels libres](https://www2.gov.bc.ca/assets/gov/government/services-for-government-and-broader-public-sector/information-technology-services/standards-files/1_02-v1_0-guidelines_on_the_use_of_open_source_software.pdf), ainsi qu’un [cadre politique](https://github.com/bcgov/BC-Policy-Framework-For-GitHub) guidant l’utilisation de GitHub par les employés.
* La province de Québec « préconise l’utilisation du logiciel libre lorsqu’il s’avère le meilleur choix et fera en sorte que les dirigeants aient le soutien nécessaire à cet effet. » Ses ressources sur les logiciels libres comprennent un Guide de référence, un Guide d’analyse de maturité et un Guide d’analyse du coût total de propriété.[[6]](#footnote-6) Le Québec a également développé la [Licence Libre du Québec](https://forge.gouv.qc.ca/licence/en/) – une licence de logiciel libre validée par Open Source Initiative et spécifiquement conçue pour être utilisée dans le secteur public, avec trois degrés de restriction quant à l’utilisation, l’application et la modification.[[7]](#footnote-7)

Même lorsqu’il n’existe pas de politique provinciale explicite encourageant l’utilisation de logiciels libres, plusieurs provinces ont mis au point des projets de logiciels libres pour répondre à des besoins spécifiques. Ainsi, le gouvernement de l’Alberta a publié une [boîte à outils sur les codes sources ouverts](https://github.com/abgov/ab-compensation-transparency-toolkit) afin d’appuyer les divulgations en vertu de la loi dite *Public Sector Compensation Transparency Act* de l’Alberta. Au plus fort de la pandémie de COVID-19, le gouvernement de la Nouvelle-Écosse a publié un [outil open source pour aider les citoyens à déterminer eux-mêmes quand appeler la ligne d’urgence 8-1-1 pour les services de santé non urgents](https://github.com/Nova-Scotia-Digital-Service/when-to-call-811).

### Les logiciels libres au niveau municipal

Les municipalités exercent également beaucoup de contrôle sur les logiciels et services spécifiques qu’elles utilisent, et le principal levier est l’approvisionnement. Le personnel doit se familiariser avec l’éventail des logiciels libres existants – et un bon point de départ pour ce faire est de savoir comment d’autres administrations ont procédé pour répondre à des besoins similaires.

* [Plusieurs municipalités canadiennes](https://cityssm.github.io/municipal-github-rankings/) participent activement au développement et au partage de projets de logiciels libres sur GitHub.
* Certaines municipalités ont adopté des politiques qui encouragent explicitement l’utilisation de logiciels libres. C’est le cas, notamment, de [Montréal](https://github.com/VilledeMontreal/politique-libre/blob/master/Politique/PolitiqueDuLibre.md#english-version), [Vancouver](https://council.vancouver.ca/20090521/documents/csb5.pdf) et [Toronto](https://www.toronto.ca/wp-content/uploads/2022/03/9728-DISFAcc2.pdf)).
* La Ville de Montréal tient également un [inventaire des solutions en logiciels libres](https://donnees.montreal.ca/ville-de-montreal/solutions-en-logiciels-libres) qu’elle utilise.

Tout comme au niveau provincial, l’absence de politiques en matière de logiciels libres n’a pas empêché plusieurs municipalités de développer des logiciels libres pour répondre à des cas d’utilisation locale. Ainsi, la ville de Sault Ste. Marie, qui partage son code via GitHub depuis 2017,[[8]](#footnote-8) a développé [plusieurs outils en logiciels libres](https://cityssm.github.io/) afin d’appuyer les opérations municipales quotidiennes dans tous les services, notamment un suivi de l’expiration des contrats, un gestionnaire de licences générales et un système de billets de stationnement.

### Lois sur la protection des renseignements personnels

Au Canada, les lois fédérales et provinciales sur la protection des renseignements personnels peuvent avoir une incidence sur la façon dont les gouvernements adoptent les logiciels libres s’ils risquent de traiter des renseignements personnels sur les citoyens. La législation qui s’applique à un logiciel libre donné dépend de l’utilisateur du logiciel et de l’usage qu’il en fait.[[9]](#footnote-9)

La législation promulguée à l’extérieur du Canada peut n’avoir qu’une incidence directe limitée sur les activités du gouvernement en matière de logiciels libres au Canada, mais il est néanmoins très important d’en tenir compte pour comprendre les normes relatives à la façon dont les logiciels sont développés pour les marchés mondiaux et la façon dont les renseignements personnels sont protégés dans les logiciels. Ainsi, le [*Règlement général sur la protection des données (RGPD*](https://gdpr-info.eu/)) de l’Union européenne (UE) impose des responsabilités supplémentaires aux responsables du contrôle et du traitement des données, et ces responsabilités peuvent s’appliquer en dehors de l’UE dans les cas où des données appartenant à des citoyens de l’UE sont visées.

## Section 2 : Enjeux et concepts clés pour la création d’un écosystème du logiciel libre

### Une organisation habilitante

Les logiciels libres exigent que les intervenants de différents ministères, organismes gouvernementaux et unités opérationnelles acquièrent de nouvelles compétences et adaptent leur rôle quotidien de façon à appuyer efficacement les logiciels libres. Il s’agit notamment des personnes affectées aux services numériques, à l’informatique, à un certain nombre de fonctions de prestation de services directs (p. ex., la circulation et le stationnement), aux achats, à la budgétisation ou à la comptabilité, aux services juridiques et à d’autres fonctions au sein des organisations et entre elles.

Toutes ces personnes doivent être capables de poser les bonnes questions, de déterminer les bons résultats et d’avoir une connaissance de base des principes fondamentaux de la conception moderne de logiciels.[[10]](#footnote-10)

### Compétences et expertise internes

La dotation en personnel technique est l’un des principaux obstacles perçus en ce qui concerne l’adoption des logiciels libres. Étant donné que les logiciels libres nécessitent une participation active de l’utilisateur (par opposition aux logiciels propriétaires, dont le service et la mise à niveau sont assurés par le vendeur), nombreux sont ceux qui supposent qu’ils sont inaccessibles sans capacités de développement de logiciels.

À l’inverse, nombreux sont ceux qui pensent que l’obtention d’un logiciel propriétaire auprès d’un fournisseur signifie qu’ils n’ont pas besoin d’experts techniques au sein de leur effectif. Or, en l’absence d’expertise interne, les processus de passation de marchés, d’acquisition et de mise en œuvre de logiciels propriétaires peuvent avoir des conséquences néfastes. Sans une connaissance adéquate de la propriété et de la monétisation des données, un contrat portant sur un logiciel propriétaire peut finir par priver le secteur public de tout pouvoir à long terme (voir le [module 3](#_heading=h.o24g9bwi3uvb)).

Les clients – le secteur public en particulier – comptent sur les fournisseurs de logiciels pour obtenir un soutien continu. Dans le cas des logiciels propriétaires, les clients n’ont pas de choix. Le code source étant caché, le fournisseur est la seule partie à y avoir accès. Dans le cas des logiciels libres, tout le monde peut contribuer, mais il n’y a pas de fournisseur unique. Les gouvernements doivent disposer de capacités internes, collaborer avec leurs pairs ou faire appel à un prestataire capable d’offrir du soutien.

### Rôles du personnel afin d’appuyer les logiciels libres

Il est possible d’entreprendre des projets de logiciels libres sans rôles officiels. Toutefois, cette approche peut ne pas être viable à long terme si les « champions » de tels logiciels au sein de votre organisation ne sont pas en mesure d’encourager les efforts en matière de logiciels libres en plus de leurs obligations professionnelles officielles.

Idéalement, vous pourrez faire appel aux compétences et à l’expertise de plusieurs intervenants différents lorsque vous vous lancerez dans une initiative de logiciel libre (voir le tableau 2.1). Ces rôles passent au premier plan ou à l’arrière-plan, selon la phase et, une fois de plus, il n’est pas nécessaire que tous ces rôles soient internes ou à temps plein.

Tableau 2.1 : Rôles associés à chaque phase d’une initiative de logiciel libre

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Participation par phase** | | | | |
| **Rôle** | **Renforcement des capacités** | **Découverte** | **Approvisionnement** | **Intégration** | **Mise à niveau** |
| **Responsable des politiques/membre de la haute direction** | Oui | Oui |  |  |  |
| **Responsable technique\*** | Oui |  | Oui |  | Oui |
| **Responsable de produits** | Oui | Oui | Oui | Oui |  |
| **Concepteur de logiciels\*** |  |  | Oui |  | Oui |
| **Spécialiste de l’approvisionnement** |  |  | Oui |  |  |
| **Spécialiste des finances** |  |  | Oui |  |  |
| **Spécialiste des contrats/du droit** |  |  | Oui |  |  |
| **Spécialiste en informatique** |  | Oui | Oui | Oui | Oui |
| **Chef du Service des utilisateurs finaux** |  |  | Oui | Oui |  |
| **Utilisateur final/Spécialiste ministériel** |  | Oui |  | Oui | Oui |

\* Indique un rôle qui peut exister en dehors de votre organisation (p. ex., au sein d’une administration semblable ou sous contrat avec un fournisseur).

### Élimination du cloisonnement ministériel

Un manque systémique de collaboration entre les ministères et les unités opérationnelles peut avoir une incidence sur l’adoption des logiciels libres et ce, pour plusieurs raisons :

* Les ministères peuvent avoir des objectifs divergents ou conflictuels, ce qui peut entraîner une désharmonisation des priorités et des frictions avant et pendant le processus d’acquisition et de développement des logiciels.
* Le cloisonnement fréquent entre les unités opérationnelles existantes (p. ex., le service juridique, l’informatique, la prestation de services, la budgétisation et l’approvisionnement) peut faire en sorte qu’il est difficile de déterminer qui est réellement *responsable* d’un processus. Cela est particulièrement vrai pour les processus nouveaux ou non conventionnels.

L’adoption de normes et de protocoles interministériels nécessite une gestion du changement de processus indépendante des services concernés.

De nombreuses grandes entreprises du secteur privé ont mis en place des bureaux de programme responsables des logiciels libres (OSPO), qui servent de centre d’échange pour tout ce qui concerne les logiciels libres au sein de l’organisation. Cela peut inclure notamment le travail en collaboration avec des fondations/organisations et des collectivités OSS, assurer la conformité légale, élaborer et mettre en œuvre des stratégies OSS, lancer de nouveaux projets, fournir des formations, fournir des conseils aux employés sur la façon dont ils pourraient s’engager dans des activités OSS et autres.[[11]](#footnote-11)

Certains gouvernements et institutions du secteur public adoptent désormais la même approche. Ainsi, la Commission européenne a créé son bureau Open Source en octobre 2020. Une structure organisationnelle telle qu’un OSPO vise à amener les gouvernements non seulement à adopter des politiques générales, mais également à garantir leur application et leur maintien d’une manière plus horizontale et stratégique, tout en assurant leur mise en œuvre dans une pratique quotidienne.[[12]](#footnote-12)

## Section 3 : Stratégies pour la création d’un écosystème du logiciel libre

### Faire preuve de leadership en matière de logiciels libres

Un leadership fort est essentiel dans toute organisation – en particulier dans le contexte gouvernemental – pour définir les politiques et renforcer les capacités qui garantiront le succès de l’adoption des logiciels libres. Si vous suivez ce cours, vous êtes en excellente position pour faire preuve d’un tel leadership.

Le renforcement des capacités afin d’appuyer l’adoption des logiciels libres peut commencer bien avant le début d’un processus d’acquisition ou de développement. Envisagez la possibilité d’organiser régulièrement, à l’échelle de l’organisation, des ateliers sur les logiciels libres, la stratégie informatique ou l’innovation en matière d’approvisionnement afin de démystifier ces sujets pour tous. Mettez l’accent sur les possibilités d’acquérir de meilleurs produits et services, la simplicité, l’efficacité et la créativité. Vous serez surpris du nombre de personnes qui feront preuve d’enthousiasme et qui voudront participer.

Réduisez les obstacles à l’adoption de logiciels libres au sein de votre organisation en donnant aux équipes qui souhaiter utiliser des logiciels libres les autorisations dont elles ont besoin pour le faire, et en simplifiant les processus pour leur permettre de miser sur la recherche et l’adaptation des logiciels en fonction des besoins de leurs utilisateurs.[[13]](#footnote-13)

### Constituer une équipe de taille adéquate

Les gouvernements – en particulier dans les petites et moyennes administrations – n’ont pas nécessairement besoin de disposer de développeurs de logiciels au sein de leur effectif. Cependant, pour obtenir et gérer efficacement les actifs numériques, ils devraient :

* Veiller à ce qu’un large éventail de membres du personnel possède une compréhension de base du fonctionnement de l’acquisition, du développement et de la mise à niveau des logiciels.
* Appuyer les groupes de travail internes et les champions.
* Se joindre à des réseaux avec des administrations comparables et des collectivités de logiciels libres pour partager les défis, les occasions, les capacités, les ressources et les processus.

Dans le cas des petites administrations ou des petits organismes, la personne responsable de la stratégie technologique de l’organisation est probablement la mieux placée pour superviser l’adoption des logiciels libres.[[14]](#footnote-14) Il est également important de noter qu’un membre du personnel peut jouer plusieurs rôles tout au long du processus. Ainsi, un spécialiste en informatique peut également être à la fois responsable du produit et utilisateur final.

Il faut toutefois éviter de compter sur des « champions » passionnés qui se chargeraient d’une grande part du travail sur les logiciels libres en plus de leur travail habituel. Les logiciels libres nécessitent un engagement actif. Or, cet engagement peut disparaître si un champion clé s’épuise, se heurte à un trop grand nombre d’obstacles ou quitte l’organisation. Si vous êtes encore en train de suivre ce cours, ce champion pourrait être *vous.* On doit préconiser un soutien systématique et à long terme des logiciels libres. Le retour sur investissement est important, d’autant plus que de plus en plus de produits logiciels sont adoptés et que les processus d’acquisition, de mise en œuvre et de mise à niveau sont solidifiés.

Dans le cas des grandes administrations ou des grands organismes disposant d’une capacité interne importante, la mise en place d’un bureau ou d’une équipe centralisée est un bon pas vers l’institutionnalisation des logiciels libres.[[15]](#footnote-15)

### Attirer les talents en instaurant une culture favorable aux logiciels libres

Il existe plusieurs raisons pour lesquelles il est plus facile de recruter et de retenir des professionnels de l’informatique talentueux dans une organisation qui privilégie l’utilisation de logiciels libres.[[16]](#footnote-16)

* Les logiciels libres sont largement utilisés par les entreprises, du développement à la production. Cela signifie que les talents misent sur le développement de leur capacité à les utiliser, et qu’ils cherchent des occasions d’améliorer leurs connaissances professionnelles pour progresser dans leur carrière.
* En contribuant à des projets de logiciels libres, les employés font partie d’un « réseau de confiance » public et étoffent leur curriculum vitae.
* Certains déclarent que le fait de travailler avec des logiciels libres améliore leur satisfaction au travail parce qu’ils contribuent à un projet communautaire de grande envergure (corrections du code source, rapports sur les bogues, mises à jour de la documentation, etc.).
* Les logiciels libres offrent des possibilités d’apprentissage, car les contributeurs ont accès à « tout ce qui se trouve sous le capot » et peuvent apprendre des développeurs plus expérimentés).

### Créer un rôle de responsable du produit

Le responsable du produit est la figure centrale d’un projet de logiciel libre, qu’il s’agisse d’identifier le besoin en logiciel, de gérer un marché ou de développer en interne. Cette personne possède des compétences en matière d’analyse, de recherche sur l’utilisateur et de réflexion sur la conception, et guide le processus logiciel, en fournissant un point de contact unique et clair pour les fournisseurs – ou pour les collègues de l’ensemble de l’organisation. Le responsable du produit établit des liens avec toutes les unités opérationnelles existantes, et s’intègre à un Service (utilisateurs finaux) au besoin, pour effectuer une recherche axée sur la découverte.[[17]](#footnote-17)

Lorsque vous créez un nouveau logiciel, le responsable du produit est une personne qui comprend votre organisation et le problème que vous essayez de résoudre, et qui pourra défendre le produit que vous allez finir par créer ensemble. Le responsable sera chargé d’établir et de respecter la vision à long terme du produit, de mettre en œuvre une stratégie et d’orienter son évolution, en fonction des recherches menées auprès des utilisateurs.[[18]](#footnote-18)

Si votre organisation n’a pas encore de rôle de responsable ou de gestionnaire de produit, envisagez d’en créer un (si vous êtes en mesure d’appuyer correctement un tel rôle) ou de plaider pour un élargissement (rémunéré) de vos responsabilités afin d’inclure la responsabilité du produit.

### Faire preuve de transparence et rester en contact avec les utilisateurs

Le fait de travailler dans un environnement ouvert centré sur l’utilisateur final constitue l’un des principes clés et l’une des bonnes pratiques reconnues dans le cas des logiciels contemporains. C’est particulièrement vrai dans le cas des logiciels libres développés selon la méthode agile (ce qui constitue, en soi, une bonne pratique). Le fait de travailler de manière ouverte signifie publier publiquement son travail sur des projets logiciels, y compris les projets existants, les progrès futurs et les autres produits du travail.[[19]](#footnote-19) Pour bien concevoir et mettre en œuvre un logiciel, vous devez connaître votre utilisateur final et le solliciter en permanence de trois manières principales :

* **Les premières recherches sur les utilisateurs.** Si vous concevez un nouveau logiciel ou rédigez un appel d’offres pour acquérir un logiciel nouveau, adapté ou existant, le point de départ est l’utilisateur. Dans chacun de ces scénarios, la première étape de votre processus consistera en une recherche axée sur la découverte auprès de l’utilisateur final, afin d’identifier ses besoins et ses difficultés. C’est également le moment de prendre en compte les besoins d’accessibilité des différents groupes d’utilisateurs et de les intégrer dès le départ.[[20]](#footnote-20) Les « scénarios utilisateurs », comme on les appelle, deviennent des descriptions des fonctionnalités du logiciel. Ils guident le développement du produit et l’évaluation du résultat final.
* **Évaluation du produit.** Au cours d’un processus de développement ou d’acquisition, l’utilisateur final doit être sollicité pour l’évaluation des maquettes et des premières ébauches. Ainsi, votre logiciel restera proche des utilisations et des besoins réels. Le personnel de première ligne peut donner son avis sur la meilleure façon d’intégrer le logiciel à son travail, et devenir autonome dans l’utilisation et la mise à niveau du logiciel au fil du temps.
* **Adaptation et mise à niveau permanentes.** En particulier dans le cas d’un logiciel libre, le logiciel sera adapté et maintenu au fil du temps. De nouvelles fonctionnalités et la conformité peuvent être ajoutées au fur et à mesure de l’évolution des normes, des politiques et des besoins des utilisateurs finaux. Les utilisateurs doivent être habilités à faire des suggestions, à examiner les fonctionnalités potentielles, à identifier les bogues et, s’ils ont la compétence requise pour ce faire, à contribuer au code proprement dit.

## Ressources supplémentaires

[**Outil d’auto-évaluation DevOps**](https://sara-sabr.github.io/ITStrategy/auto-evaluation-devops.html): Conçu pour l’équipe de stratégie en matière de technologie de l’information d’Emploi et Développement social Canada (EDSC), ce questionnaire est un bon point de départ pour d’autres organisations gouvernementales afin d’évaluer les capacités techniques, les processus, les mesures et la culture de leur équipe en ce qui concerne la livraison de logiciels et le rendement organisationnel, de comprendre leurs forces et leurs défis, et de formuler des recommandations sur des domaines d’amélioration spécifiques. Il permettra de déterminer les domaines dans lesquels les compétences de votre équipe n’atteignent pas le niveau requis pour atteindre les objectifs de l’équipe.

[**OSPO++**](https://ospoplusplus.com/): OSPO++ est un réseau et une collectivité de bureaux de programmes collaboratifs à code source ouvert dans les universités, les gouvernements et les institutions civiques. Il met en place des ressources pour aider à créer des OSPO et favorise les discussions sur la meilleure façon de gérer et de développer les programmes de sources ouvertes.

[**Exemples et modèles de politique de source ouverte**](https://github.com/todogroup/policies)**:** Une liste des politiques existantes en matière de sources ouvertes qui peuvent être adaptées au contexte de votre organisation.

# Module 3 : Approvisionnement, budgétisation et passation de marchés.

## Objectifs d’apprentissage

Ce module vous permettra de vous familiariser avec :

* Les principales différences entre les approches en matière de licence et de développement et la raison pour laquelle les systèmes gouvernementaux ont tendance à privilégier les logiciels propriétaires et le développement en cascade.
* Les différents modèles de revenus des logiciels et de la manière dont ils affectent les structures de coûts des gouvernements.
* Les nouvelles approches en matière d’approvisionnement, de passation de marchés, d’octroi de licences et de dotation en personnel.

## Éléments clés

* **Les structures et les normes en matière d’approvisionnement, de budgétisation et de passation de marchés sont conçues pour des actifs antérieurs à l’ère numérique – ce qui complique l’obtention de logiciels libres.** Cependant, à mesure que les organisations gouvernementales adoptent des approches axées sur le numérique, les logiciels libres et la méthode agile deviennent la voie la plus facile.
* **Les logiciels libres nécessitent une approche différente de celle des logiciels propriétaires en matière de budgétisation.** Alors que les logiciels propriétaires sont tarifés selon l’un des nombreux modèles de revenus standard, les logiciels libres sont distribués gratuitement, mais nécessitent des investissements en personnel et/ou en fournisseurs de services pour déployer et maintenir les logiciels.
* Les **logiciels libres s’harmonisent bien avec les méthodes « agiles »**, qui se révèlent également être les pratiques exemplaires en matière de développement de logiciels. Lors de l’acquisition de services logiciels personnalisés, le gouvernement devrait utiliser des approches contractuelles et budgétaires modulaires, qui permettent un développement agile.
* **La passation de marchés est un processus créatif et exploratoire.** En utilisant correctement les marchés publics, vous pouvez non seulement trouver la meilleure solution au moindre coût, mais aussi générer des retombées positives.
* L**’achat groupé est une stratégie efficace** permettant à un groupe d’organisations gouvernementales de partager le coût de l’obtention et de la mise à niveau de logiciels libres répondant à un besoin commun.

## Section 1 : Modèles d’octroi de licences de logiciels, de revenus et de développement.

### Approvisionnement dans le contexte gouvernemental

Le gouvernement dispose de normes, de processus et de modèles bien établis pour l’approvisionnement, la budgétisation et la passation de marchés. Ces normes, processus et modèles ont été conçus à l’origine pour des actifs non numériques et des services de fournisseurs (comme le matériel de bureau ou le déneigement). Ces actifs et services sont fixes et prévisibles. Lorsque les gouvernements utilisent ces processus traditionnels pour obtenir des logiciels, ils ont tendance à privilégier les actifs logiciels (logiciels propriétaires) et les services (développement en cascade) fixes et prévisibles. Ils partent du principe que l’expertise technique est externe, et ne tiennent pas compte du partage entre pairs et des communautés de pratique. Mais, comme l’a fait remarquer une journaliste, il est de plus en plus fréquent que les longs cycles budgétaires et les mentalités des gouvernements ne correspondent pas au rythme de la technologie et à son besoin d’améliorations et de mise à niveau constantes.[[21]](#footnote-21)

Les dirigeants du secteur public ont la possibilité d’adopter une approche axée sur le numérique pour l’approvisionnement, la budgétisation et la passation de marchés, ce qui serait conforme aux pratiques exemplaires en matière de développement de logiciels, tout en préservant l’intégrité d’un processus d’approvisionnement public légitime, équitable, transparent et objectif.

Les gouvernements ont des ressources limitées, surtout lorsqu’il s’agit d’ajouter des étapes numériques aux unités opérationnelles et aux services existants. Des processus d’approvisionnement efficaces et des approches intelligentes en matière de logiciels libres peuvent les aider à atteindre les objectifs de transformation numérique malgré des budgets limités.

Les processus numériques d’approvisionnement, de budgétisation et de passation de marchés permettent aux employés du secteur public de contrôler la manière dont leurs outils numériques sont conçus, gérés et mis en œuvre, en particulier lorsqu’ils utilisent des logiciels libres.

### Octroi de licences de logiciels

N’importe qui peut développer un code logiciel et appliquer la protection qu’il souhaite. Une **licence de logiciel** est un document juridique qui définit comment un logiciel peut être utilisé, commercialisé, modifié, concédé en sous-licence et redistribué. Les licences de logiciels sont juridiquement exécutoires en vertu de la *Loi sur le droit d’auteur*.

Il existe deux grandes catégories de licences de logiciels : les licences propriétaires et les licences libres.

* Une **licence propriétaire** accorde à un utilisateur final le droit d’utiliser un logiciel, généralement contre rémunération. Il est interdit aux utilisateurs de modifier, copier, concéder en sous-licence ou distribuer le logiciel. Les logiciels propriétaires sont conçus de façon à empêcher les utilisateurs d’accéder au code source.
* Une **licence libre** confère à quiconque le droit d’utiliser, d’étudier, de modifier et de distribuer le logiciel et son code source à d’autres, dans n’importe quel but (souvent avec attribution). Le code sous-jacent (code source) est accessible aux utilisateurs finaux.

Les exemples les plus courants de licences de logiciels libres sont [Creative Commons](https://creativecommons.org/), la [licence publique générale GNU](https://www.gnu.org/licenses/gpl-3.0.en.html) et la [licence MIT](https://opensource.org/licenses/MIT). Ces licences permettent au créateur d’un logiciel de préciser les conditions de son utilisation, de sa reproduction et de sa commercialisation.

Des organisations comme la [Free Software Foundation](https://www.fsf.org/) et le [Software Freedom Law Center](https://softwarefreedom.org/) occupent une place importante dans la collectivité des logiciels libres. Ils sont les responsables de fait de l’application des licences de logiciels libres. Toutefois, les actions en justice sont rares, car les licences de logiciels libres vont directement de l’auteur/propriétaire au titulaire de la licence, [de sorte que] la seule partie en position légale pour faire respecter [les protections de la licence] est l’auteur ou le propriétaire du droit d’auteur. Les distributeurs intermédiaires n’ont pas le pouvoir d’exécution, sauf en ce qui concerne leur contribution éventuelle.[[22]](#footnote-22)

### Aperçu des modèles de revenus des logiciels

Les sociétés de développement de logiciels ont adopté divers modèles pour générer des revenus à partir de logiciels protégés par une licence propriétaire. Voici quelques-uns des modèles de revenus les plus courants, ainsi que des exemples de logiciels qui les utilisent.

* **Licence/achat unique.** Il s’agit de vendre un logiciel sous forme de licence qui peut être utilisé par un seul utilisateur ou un groupe d’utilisateurs. L’idée générale est de proposer un produit qui ne nécessite qu’un seul paiement.
  + Microsoft Windows, serveur Apache.
* **Abonnement (paiement récurrent).** Contrairement à la licence, l’utilisateur a accès au logiciel en payant un abonnement mensuel ou annuel.
  + Netflix, Spotify, Adobe Creative Suite.
* **Logiciel payable à l’utilisation.** Cette méthode de tarification est principalement utilisée dans le cas de différents produits et services basés sur l’informatique en nuage qui vous facturent les puissances de calcul/la mémoire/les ressources/le temps utilisés.
  + Amazon Web Services et Google Cloud Platform.
* **Freemium (vente incitative).** Le freemium est un type de monétisation en vertu duquel un utilisateur peut accéder gratuitement à un produit de base, mais pour lequel il sera facturé pour des fonctions, services, suppléments ou modules d’extension supplémentaires.
  + Skype, Evernote, Gitbook.
* **Tarification hybride.** Les entreprises de logiciels combinent parfois des éléments de plusieurs modèles de revenus différents dans leurs plans de tarification. Ainsi, un plan freemium pourrait se transformer en une forme de plan à plusieurs niveaux avec paiement à l’utilisation. Après avoir dépassé une certaine limite en termes de calcul ou de ressources, un utilisateur peut être contraint d’utiliser un autre type de tarification, ou invité à le faire.
  + Mailchimp, Amazon Web Services, SalesForce.[[23]](#footnote-23)
* **Prestation de services.** Certaines sociétés de développement de logiciels facturent des services spécifiques, comme la création d’une instance personnalisée d’un logiciel libre existant. En pareil cas, le client paie le temps et l’expertise nécessaires pour la personnalisation et la mise en œuvre du logiciel, et non le logiciel proprement dit.
  + GNU/Linux, SugarCRM, LibreOffice.

### Modèles de développement de logiciels

Il existe deux principaux modèles de développement de logiciels.

* Le développement **en cascade** exige un plan détaillé, linéaire et séquentiel – faisant état de chaque détail, chaque fonctionnalité, chaque réunion, chaque repère – avant d’entreprendre un projet. L’inconvénient du développement en cascade est qu’il ne se prête pas bien à la plupart des développements de logiciels, car il sera presque toujours nécessaire de s’adapter à de nouvelles contraintes, de nouvelles exigences ou de nouveaux problèmes. Et, comme les développeurs ne sont pas en mesure de prévoir tous les problèmes potentiels, il est très probable que le produit final soit défectueux.
* Le développement **agile** est axé sur un résultat final prévu, et peut-être même un échéancier, mais il n’entre dans les détails qu’en cas de besoin, ce qui permet des changements en réponse à de nouvelles informations. Le développement agile commence par la définition d’histoires d’utilisateurs qui représentent des cas d’utilisation et des fonctionnalités réels. Le codage se fait par « sprints » d’une durée d’une à quatre semaines, chacun étant consacré au développement d’une fonctionnalité logicielle basée sur un récit utilisateur. Grâce à une méthode de développement agile, des modules distincts mais interdépendants peuvent être développés, testés, déployés et révisés facilement.

### Le décalage entre l’acquisition et le développement de logiciels

Le processus d’octroi de marchés publics est conçu de façon à obtenir le meilleur rapport qualité‑prix et éviter les risques. L’objectif est d’identifier un besoin et de trouver le produit le moins cher qui répond à ce besoin dans un délai prévisible.[[24]](#footnote-24)

Il s’agit d’une bonne approche pour l’achat de produits analogues (p. ex., matériel de bureau, lampadaires, service de déneigement). Définir les spécifications, comparer les options en fonction des réponses à un appel d’offres, prévoir les coûts futurs et rédiger un contrat à prix ferme sont des tâches assez simples.

Toutefois, en ce qui concerne les logiciels, les structures d’approvisionnement traditionnelles vous inciteront à acheter des logiciels prêts à l’emploi ou à contracter des logiciels personnalisés développés en cascade, en fonction de feuilles de route pluriannuelles surspécifiées. Mais cette approche est de moins en moins conforme aux pratiques exemplaires du secteur de développement des logiciels, qui privilégie les pratiques agiles telles que l’itération rapide et les tests continus auprès des utilisateurs pour créer des logiciels qui répondent à leurs besoins. Comme l’a fait remarquer un expert, cela a laissé une déconnexion fondamentale entre les pratiques de passation de marchés (il est plus prudent de tout spécifier) et les pratiques de développement de logiciels (il est dangereux de tout spécifier).[[25]](#footnote-25)

L’approvisionnement conventionnel – l’achat d’un logiciel offert sur le marché ou le développement d’un produit personnalisé selon la méthode en cascade – peut *sembler être* l’option la moins risquée. Cependant, cela conduit continuellement à une explosion des coûts et à des produits finis qui ne servent pas un objectif clair ou, pire encore, qui se terminent en catastrophe.

### Faire correspondre les pratiques exemplaires en matière de marchés publics et de logiciels

Il est important de garder à l’esprit que lorsque vous achetez un logiciel personnalisé, vous n’achetez pas un produit. Vous achetez un service, le service de développeurs qui créent des logiciels, avec des fonctionnalités classées par ordre de priorité par un propriétaire de produit du gouvernement.[[26]](#footnote-26)

Les logiciels doivent être évalués non seulement en fonction de leur coût, mais également en fonction de leur qualité, des coûts de propriété et de mise à niveau sur le cycle de vie à long terme, et de problèmes potentiels imprévus tels que le verrouillage du fournisseur, l’augmentation des frais de licence, le coût des fonctions supplémentaires, les frais de mise à niveau et la propriété des données.[[27]](#footnote-27)

Nous examinerons, dans les prochaines sections, ce que cela signifie pour l’obtention de services de développement afin de créer une instance personnalisée de logiciel libre. Nous explorerons également les pratiques exemplaires et les outils d’approvisionnement et de budgétisation modulaire qui permettent un développement logiciel agile efficace.

## Section 2 : Enjeux et concepts clés en matière d’approvisionnement

### Saisir l’occasion

L’approvisionnement est traditionnellement considéré comme une procédure de conformité qui implique des spécifications détaillées a priori, identifie les solutions les moins coûteuses et empêche la corruption. Or, l’approvisionnement est également un outil puissant de création de valeur stratégique.

Lorsque les administrations saisissent les occasions qu’offre l’approvisionnement, elles peuvent atteindre de manière créative les grands objectifs du secteur public, promouvoir l’innovation (à l’intérieur et à l’extérieur du gouvernement) et stimuler le développement économique. En bref, elles peuvent façonner la *valeur publique future*.[[28]](#footnote-28)

Tous les ordres de gouvernements investissent énormément dans leurs économies locales. Des fonds importants ont été affectés à la transformation numérique, aux services techniques partagés et à la mise à niveau des systèmes informatiques au niveau fédéral.[[29]](#footnote-29)

Selon la Banque de développement du Canada (BDC), il existe 41 800 entreprises de TI au Canada, et 90 % d’entre elles comptent 10 employés ou moins. Ensemble, ces entreprises apportent 57 milliards de dollars à l’économie canadienne.[[30]](#footnote-30) Toutefois, les marchés publics classiques favorisent généralement les grands fournisseurs bien établis, ce qui signifie que les petites entreprises et les entreprises en démarrage sont systématiquement exclues du processus.[[31]](#footnote-31) Les gouvernements peuvent mieux appuyer les collectivités technologiques nationales et locales en s’engageant auprès des fournisseurs locaux et en incluant des exigences en matière d’approvisionnement local dans leurs appels d’offres.

### Créer de la valeur avec un processus d’approvisionnement

L’approvisionnement est essentiellement un processus qui commence par l’identification d’un besoin fondamental ou d’une occasion, suivie de l’exploration et de l’obtention de solutions. Si ce processus est bien conçu, il peut générer des retombées positives (voir encadré 3.1). En utilisant des outils spécifiques tels qu’un appel d’offres, le processus d’approvisionnement peut préciser ce qui suit :

* **Critères du vendeur :** Petits fournisseurs locaux, appartenant à des membres de minorités visibles ou à des organismes sans but lucratif.
* **Critères de solution :** Propriété ou contrôle public, modèles de coûts équitables et durables, fonctionnalités, résultats (plutôt que de miser des solutions existantes).

Dans le cas des logiciels libres, les retombées positives pourraient être les suivantes : favoriser un écosystème local plus diversifié de fournisseurs de services technologiques,[[32]](#footnote-32) partager des logiciels avec des administrations comparables, et maintenir le contrôle public sur la manière dont les données sont utilisées et commercialisées.

#### Encadré 3.1 : Approvisionnement selon la méthode agile au sein de l’administration fédérale

Les marchés publics se transforment au niveau fédéral. En 2017, le Secrétariat du Conseil du Trésor du Canada (SCT) a testé un nouveau modèle d’approvisionnement agile conjointement avec le projet pilote « Ouvert par défaut » du gouvernement du Canada. Le SCT a déployé un « un processus d’approvisionnement innovateur et rigoureux sous forme de défi, dans le cadre duquel des entreprises qualifiées proposaient des idées à un panel d’évaluateurs ». 2 Après les présentations, le contrat a été attribué le même jour. De plus, les soumissionnaires ont pu faire part de leurs commentaires sur le processus au personnel responsable des achats du SCT et aux évaluateurs.3

En 2022, Services partagés Canada (SSC) a lancé le processus d’approvisionnement agile 3.0 – une nouvelle approche simplifiée d’approvisionnement en TI qui met l’accent sur l’acquisition d’une petite démonstration ou d’un prototype comme stratégie d’évaluation des solutions potentielles. « Les fonctionnaires peuvent ainsi évaluer des exemples concrets des solutions proposées et choisir la solution qui offre le meilleur rapport qualité-prix au Canada et à la population canadienne ». 4

1 Alex Benay, [*L’approvisionnement agile pour de meilleures solutions numériques*](https://ouvert.canada.ca/fr/blogue/lapprovisionnement-agile-meilleures-solutions-numeriques), Blogue sur le gouvernement ouvert, le 27 juillet 2017.

2 Secrétariat du Conseil du Trésor du Canada, [*Le gouvernement du Canada octroie un contrat de services numériques à l’aide d’une méthode simplifiée d’approvisionnement*](https://www.canada.ca/fr/secretariat-conseil-tresor/nouvelles/2017/09/le_gouvernement_ducanadaoctroieuncontratdeservicesnumeriquesalai.html), le 14 septembre 2017.

3 Michael Grant, [*An Open, Honest Talk About Procurement Improvements*](https://blog.echidna.ca/open-honest-talk-about-procurement-improvements), Digital Echidna Blog, le 14 septembre 2017,

4 Services partagés Canada, [*Services partagés Canada lance le processus d’approvisionnement agile 3.0*](https://www.canada.ca/fr/services-partages/nouvelles/2022/03/services-partages-canada-lance-le-processus-dapprovisionnement-agile30.html), le 23 mars 2022.

### Gestion du processus d’approvisionnement : Le responsable de produit

L’approvisionnement relève généralement d’un responsable de l’approvisionnement et repose sur un budget détaillé et des spécifications fonctionnelles. Ces spécifications proviennent souvent de sources diverses, comme un spécialiste en informatique et un cadre supérieur chargé des directives politiques. Les spécifications peuvent être contradictoires ou manquer de clarté. De façon plus générale, un flux d’information à sens unique (des spécialistes aux responsables de l’approvisionnement, puis aux fournisseurs) compromet les possibilités d’élaboration de solutions créatives.

Dans le cadre d’un processus d’approvisionnement numérique, il existe une figure centrale chargée d’assurer la participation de toutes les unités opérationnelles et de faire progresser l’ensemble du processus d’approvisionnement : le responsable du produit. Cette personne n’est pas nécessairement un expert technique, mais elle doit avoir une connaissance pratique des processus de développement de logiciels, de l’approvisionnement et des méthodes de conception centrées sur l’utilisateur (voir le [module 2](#_heading=h.4w0x83lql6tt) pour obtenir de plus amples détails).

### Découverte : Établir une base solide grâce à la compréhension du problème

La phase de « découverte » est le point d’entrée de l’acquisition de logiciels. Elle comporte trois éléments principaux :

1. Définition du problème à résoudre, c’est-à-dire recherche sur les utilisateurs avec les bénéficiaires et les utilisateurs finaux du logiciel (au moyen de séances de conception et de planification, de séances de remue-méninges, etc.).[[33]](#footnote-33)

* Sondages et entretiens
* Démonstrations
* Cartographie du parcours de l’utilisateur
* Accompagnements

1. Examen des produits qui existent déjà, approche adoptée par d’autres administrations pour aborder le même problème, consultation des référentiels Github du gouvernement, lecture des blogues et participation à des conférences.

* Étude de marché et analyse du marché neutre en ce qui concerne les solutions commerciales existantes[[34]](#footnote-34).
* Examen de l’approche adoptée par d’autres administrations pour aborder des problèmes similaires.
* Consultation de l’[Échange de ressources ouvertes du GC](https://code.ouvert.canada.ca/fr/index.html).

1. Rencontre des fournisseurs, des experts et des passionnés des technologies. Discussion au sujet des problèmes à résoudre, et réflexions des intervenants sur les solutions existantes ou potentielles. Approches à adopter :

* Solliciter des idées de façon informelle (auprès de la collectivité des technologies).
* Publier une demande de renseignements formelle (DR).
* Organiser un événement lié au problème et inviter des groupes communautaires et des fournisseurs[[35]](#footnote-35).
* Consulter le milieu universitaire.

Selon une idée fausse très répandue, les fonctionnaires ne peuvent pas contacter directement les fournisseurs potentiels. En réalité, la compréhension du marché est un point de départ crucial pour trouver la meilleure solution. Il est utile d’avoir une bonne compréhension de ce que ces entreprises peuvent faire lorsque vous envisagez de lancer une demande de services. Vous pouvez également partager les besoins de votre organisation afin qu’ils se familiarisent avec votre feuille de route à long terme pour l’intégration des logiciels dans la prestation de services publics. Des contrats répétés permettent aux fournisseurs de services de se familiariser avec votre environnement technique et vos besoins spécifiques. Afin de maintenir l’intégrité et d’éviter toute inquiétude, assurez-vous que tous les échanges avec les fournisseurs sont publics, bien documentés et accessibles à la population sous forme de synthèse.

### Stratégies d’approvisionnement

Si vous suivez un processus de découverte rigoureux, vous découvrirez un certain nombre de stratégies d’approvisionnement – achat de logiciels existants, utilisation et adaptation de logiciels libres ou contrat avec un fournisseur pour la création de nouveaux logiciels. Vous devez les comparer par l’entremise d’une analyse équitable et objective du coût réel de possession sur la durée de vie. Selon les résultats, vous serez en mesure de choisir en toute confiance la bonne stratégie d’approvisionnement pour le défi à relever.[[36]](#footnote-36) Il existe un certain nombre de scénarios possibles, et chacun comporte ses propres coûts :

* **S’il existe un logiciel propriétaire *et* un logiciel libre qui répondent à tous les besoins des utilisateurs,** vous devrez choisir entre l’acquisition du logiciel propriétaire existant ou la recherche d’une solution logicielle libre.
* **S’il existe un logiciel libre qui répond à tous les besoins des utilisateurs**, une équipe interne ou un sous-traitant peut déployer une instance personnalisée qui s’intègre à votre environnement technique existant.
* **Si un logiciel libre existant doit être modifié** pour répondre à tous les besoins des utilisateurs, une équipe interne ou un sous-traitant peut développer les fonctionnalités personnalisées dont vous avez besoin.
* **Si aucun logiciel existant ne répond aux besoins des utilisateurs,** une équipe interne ou un entrepreneur embauché à contrat peut développer un logiciel à partir de rien et le diffuser sous une licence libre.

A decision tree for choosing a software sourcing strategy.
If proprietary and open source software exists and meets all user needs, you will choose between procuring existing proprietary software or pursuing an open source software solution.
If there is existing open source software that meets all user needs, an in-house team or hired contractor can deploy a custom instance that integrates with your existing technical environment.
If there is existing open source software that needs to be altered in order to meet all user needs, an in-house team or hired contractor can develop the custom features you need.
If there is no existing software that fills the need, an in-house team or hired contractor can develop software from scratch and release it under an open source license.

Figure 3.1 : Un arbre de décision pour le choix d’une stratégie d’approvisionnement en logiciels

### Analyse du coût réel du cycle de vie et de la valeur de la propriété

Afin d’évaluer correctement les options d’approvisionnement, vous devez effectuer une analyse du coût réel du cycle de vie et de la valeur de la propriété, qui comprend les coûts et avantages directs et indirects, immédiats et à long terme.[[37]](#footnote-37) Une telle analyse devrait, plus précisément, inclure :

* Le coût direct de la solution (licence annuelle ou propriété).
* La capacité technique du personnel pour ce qui est de développer, d’adapter, d’intégrer et/ou de mettre à niveau les logiciels.[[38]](#footnote-38)
* Les coûts d’acquisition, y compris tous les coûts de la procédure d’achat et de la gestion du changement (le cas échéant).
* Les coûts d’exploitation, y compris les frais annuels, le coût par utilisateur et l’assurance.
* Le coût de l’intégration à l’infrastructure numérique existante de la ville (en termes de temps du personnel et de coûts des logiciels ou des bases de données).
* Les coûts des versions ultérieures et de l’élimination, ce qui pourrait inclure le retrait de l’installation, les coûts de nettoyage, la diffusion du code sous une licence libre et les frais juridiques connexes.
* Le coût de la personnalisation future du logiciel (bons de travail) si les besoins, les processus ou l’infrastructure numérique sous-jacente de la ville changent à l’avenir.
* La valeur du contrôle des coûts, des modèles de revenus, des mises à jour et des ensembles de fonctionnalités supplémentaires (notamment pour la conformité aux normes de rendement des logiciels gouvernementaux en évolution).
* La valeur du renforcement des capacités et de l’habilitation du personnel.
* L’intérêt de publier le code sous une licence libre et de s’allier à une collectivité de pairs.

### Évaluation des options

Lors de l’évaluation d’un logiciel propriétaire existant, adoptez une approche critique et prudente quant aux problèmes potentiels de contrats en aval. Ces problèmes peuvent découler du modèle économique du fournisseur, de son approche de la propriété ou de la monétisation, ou encore du rendement du logiciel et des mises à jour des fonctionnalités. Cette évaluation devrait influencer votre choix de stratégie.

* **Propriété :** Quelles sont les conditions exactes d’octroi de la licence? Le logiciel dépend‑il d’autres systèmes propriétaires (comme les bases de données) pour lesquels il y a un coût?
* **Sécurité :** Y a-t-il des vulnérabilités connues? Quelle partie est responsable de la correction des vulnérabilités?
* **Gouvernance des données :** En quoi consistent les données recueillies? Qui en est le propriétaire? Où sont-elles stockées? Qui y a accès? Comment peuvent-elles être utilisées?
* **Interopérabilité :** Quelle est sa compatibilité avec l’infrastructure technique existante?
* **Modèle de revenus :** Comment le fournisseur génère-t-il ses revenus (par exemple, frais par utilisateur, frais d’utilisation, fonctions supplémentaires, etc.)
* **Mise à niveau :** À quelle fréquence le logiciel est-il mis à jour? Quel type de soutien continu est offert aux utilisateurs?

## Section 3 : Budgétisation et passation de marchés

### Passer des dépenses d’investissement aux dépenses d’exploitation

Les logiciels libres sont gratuits, mais leur personnalisation et leur mise à niveau demandent du travail. Cela signifie que la structure des coûts sera différente. Par conséquent, la budgétisation des logiciels libres doit tenir compte du passage des *dépenses d’investissement* – achat d’un produit (logiciel propriétaire) – aux *dépenses de fonctionnement* – achat d’un service (travail de développement et d’intégration).

Les organisations gouvernementales devront embaucher du personnel capable de mettre en œuvre les logiciels et d’en assurer la mise à niveau, ou passer des contrats avec des fournisseurs de services technologiques (en rédigeant des contrats ouverts pour les services, par opposition aux coûts fixes pour les produits numériques et les licences).

### Adopter une approche contractuelle modulaire pour le développement agile

Le développement agile est structuré en fonction de courts « sprints », chacun étant axé sur la création d’un module autonome. Le contrat modulaire est une approche de la planification financière qui appuie cette cadence de travail.[[39]](#footnote-39)

Dans un scénario de contrat modulaire, vous devez d’abord établir un contrat-cadre pour le projet en suivant une procédure régulière et en obtenant les approbations appropriées. Ensuite, vous modularisez le travail et rédigez des contrats de sous-traitance pour des composants spécifiques, en veillant à ce que le montant du contrat soit un reflet légitime du travail exigé (plutôt que de viser un montant juste en dessous du seuil d’approvisionnement).[[40]](#footnote-40)

De cette manière, vous maîtriserez les coûts et vous pourrez veiller à ce que le prestataire de services respecte les délais et les attentes en matière de qualité. Si ce n’est pas le cas, vous pouvez rompre le contrat-cadre et attribuer un contrat pour la suite à un autre fournisseur –qui sera facile à intégrer au travail déjà accompli (parce qu’il est modulaire et basé sur des composants libres). Les contrats modulaires vous donnent le contrôle, ce qui vous permet de réduire le risque global (voir encadré 3.2).

Toutefois, il faut se méfier du *fractionnement des offres* ou de la perception de procédés frauduleux. C’est le cas lorsqu’un employé du gouvernement divise les gros contrats en petits contrats pour éviter l’examen minutieux requis dans le cas des gros contrats.[[41]](#footnote-41) Bien que les responsables de l’audit et de la conformité s’habituent de plus en plus aux contrats modulaires pour un développement agile, il est néanmoins important d’anticiper cette préoccupation. Ainsi, la US General Services Administration propose des « accords d’achat généraux » qui permettent de gérer et de justifier des groupes de contrats modulaires dans le cadre d’un contrat plus important.[[42]](#footnote-42)

#### Encadré 3.2 : Créer de la valeur en utilisant l’approvisionnement modulaire pour le nouveau système de services de protection de l’enfance de la Californie

En 2015, l’État de Californie a décidé d’adopter une approche différente pour l’acquisition d’un nouveau système de services de protection de l’enfance, après avoir connu une série d’échecs dans le cas de grands projets informatiques dont les coûts s’élevaient à des centaines de millions de dollars.1

Le maintien du statu quo aurait signifié l’émission d’une demande de propositions (DP) monolithique et pluriannuelle dont le coût était estimé à plusieurs centaines de millions de dollars et dont la mise en œuvre aurait pris de cinq à sept ans.2 Au lieu de cela, l’État de la Californie a collaboré avec 18F (l’unité de transformation technologique de l’administration fédérale des services généraux) et le ministère américain de la santé et des services sociaux pour utiliser une approche d’approvisionnement modulaire associée à des méthodologies de conception et de développement agiles afin d’obtenir plus rapidement une valeur commerciale.3 Le projet a été divisé en modules plus petits qui pouvaient faire l’objet d’appels d’offres individuels, ce qui a accéléré le développement du système tout en réduisant le risque global en cas de défaillance d’un composant.

Outre les avantages directs pour le projet proprement dit, cette nouvelle approche a permis de remporter plusieurs autres victoires. Le projet a attiré de nouveaux soumissionnaires grâce à son [bassin de fournisseurs agiles présélectionnés](https://cwscms.osi.ca.gov/CWDSADPQ), et a donné l’occasion de « travailler à découvert » et de tester les modules avec les utilisateurs finaux avant le déploiement. De plus, en affinant de manière itérative le processus de demande de propositions, ce projet constitue un modèle pour d’autres projets dans d’autres ministères.4

1 Stuart Drown et Mike Wilkening, [*Leadership and Innovation at California’s Child Welfare Services*](https://18f.gsa.gov/2016/11/17/leadership-innovation-california-child-welfare-services/), *18F* (blog), le 17 novembre 2016.

2 [*CWDS Procurement Management Plan, Child Welfare Digital Services*](https://cwds.ca.gov/CWDS_Procurement_Management_Plan), consultation le 30 novembre 2022.

3 [*CWDS Procurement Management Plan*](https://cwds.ca.gov/project_management_plan)*.*

4 Stuart Drown et Mike Wilkening, [*Leadership and Innovation at California’s Child Welfare Services*](https://18f.gsa.gov/2016/11/17/leadership-innovation-california-child-welfare-services/)*.*

### Passer d’un financement basé sur les projets à un financement basé sur les produits

Votre analyse du coût de possession sur toute la durée de vie devrait avoir permis d’identifier les coûts d’entretien permanents. Il est important de créer un plan financier à long terme pour les applications de logiciels libres. Un contrat de service classique mise sur le projet en cours, par exemple la *création et la livraison d’un logiciel*. Toutefois, cette approche ponctuelle peut entraver la capacité d’itération et d’amélioration fréquente d’une équipe informatique,[[43]](#footnote-43) notamment en ce qui concerne la mise à niveau à long terme.

Les gouvernements devraient plutôt s’orienter vers une planification axée sur les feuilles de route de produits, avec un financement stable pour les flux de valeur composés de membres d’équipes multidisciplinaires capables d’appuyer les demandes et de prendre des décisions opportunes reflétant les besoins des utilisateurs finaux, et une gouvernance qui mesure les progrès par l’entremise de logiciels fonctionnels plutôt que de documents de planification.[[44]](#footnote-44)

Dans un paradigme de financement basé sur le produit, une unité opérationnelle spécifique (comme le Service des parcs) peut parrainer un projet qui est réalisé par une équipe de produit multidisciplinaire incluant les utilisateurs finaux du Service des parcs, de l’informatique et autres. L’équipe multidisciplinaire identifie le défi, acquiert ou développe un produit, et établit des cadres pour une mise à niveau continue.

Le personnel est donc responsable de la mise en place et du maintien d’un portefeuille de produits dans le temps. Cela va de pair avec l’utilisation de logiciels libres. En utilisant des modules interopérables, une équipe bien budgétée et gérée peut mettre en œuvre et maintenir un large portefeuille de produits logiciels. Les ressources de l’équipe informatique doivent tenir compte des coûts directs (main-d’œuvre, abonnements et licences de logiciels) et indirects (ressources communes).

La Politique sur les services et le numérique conseille aux dirigeants de rechercher un financement suffisant dans les budgets de renouvellement des programmes afin d’appuyer le cycle de vie complet d’un produit, y compris les améliorations stratégiquement prévues. Cela peut nécessiter le recours à différentes enveloppes de financement pour le développement et la mise à niveau continue. Les budgets doivent tenir compte des coûts directs (main-d’œuvre, abonnements et licences de logiciels) et indirects (ressources communes). Ce type de financement est connu sous le nom de financement par produit, car il est axé sur le succès continu du *produit*, et non sur la réalisation d’un *projet* ponctuel.[[45]](#footnote-45)

### Participer à un accord d’achat groupé

Il n’est pas toujours possible d’acquérir ou de développer des logiciels libres par ses propres moyens, surtout dans le cas des petites et moyennes municipalités. Mais la plupart des administrations ont besoin d’un logiciel similaire, et aucune d’entre elles ne devrait en supporter seule le coût total. Les achats groupés sont courants dans d’autres domaines (comme le transport), mais ne sont pas monnaie courante dans le cas des logiciels. Zachary Spicer note que les accords de services partagés dans le domaine des technologies de l’information sont beaucoup moins courants que dans d’autres domaines municipaux, tels que les transports ou les services d’urgence, ce qui indique que les municipalités ont la possibilité d’explorer ces types d’accords.[[46]](#footnote-46) Cela est probablement dû au fait que l’achat groupé est impossible dans le cadre d’un paradigme de licence de logiciel propriétaire; un tel mode d’achat est interdit par le contrat de licence. Les logiciels libres étant libres d’utilisation, d’adaptation et de redistribution, les gouvernements sont parfaitement en droit de se procurer collectivement des services de développement ou de personnalisation auprès d’un fournisseur de services technologiques et de les partager entre eux. Les achats groupés sont importants pour les raisons suivantes :

* Chaque bénéficiaire fait un investissement initial moins important.
* De meilleurs logiciels peuvent être construits avec des ressources regroupées.
* Optimisation des ressources financières.
* Les pairs contribuent à l’amélioration du code partagé[[47]](#footnote-47).

Il existe plusieurs outils et stratégies pouvant être utilisés pour organiser des achats groupés entre un groupe d’organisations (voir le module 5).

## Ressources supplémentaires

[**Guide pour la réforme de l’approvisionnement en technologie de l’information au gouvernement du Canada**](https://govcanadacontracts.ca/it-procurement-guide/): Basé sur un projet de recherche mené par la Dre Amanda Clarke, en collaboration avec Sean Boots, ce guide offre un aperçu succinct des problèmes et des défis associés à l’approvisionnement en technologie contemporaine au sein du gouvernement du Canada et des raisons pour lesquelles une réforme du processus d’approvisionnement s’impose. À cette fin, il propose également une série de recommandations politiques et de pratiques exemplaires émergentes à l’intention des dirigeants gouvernementaux.

[**Citymart Procurement Institute**](https://medium.com/citymartinsights)**:** Un guide pour les fonctionnaires des gouvernements locaux afin de mieux comprendre comment les marchés publics peuvent faire partie de leur processus de gouvernance, en enrichissant la politique et la prise de décisions. Cette ressource contient huit unités d’apprentissage et chacune d’entre elles porte sur une étape logique du processus d’approvisionnement. Chaque unité contient une introduction au sujet, des exemples, des avis d’experts, des exemples de cas et des feuilles de travail pour vous aider à démarrer.

[**Open Contracting Partnership Resources Archive**](https://www.open-contracting.org/resources/): Toutes les stratégies d’approvisionnement, de budgétisation et de passation de marchés examinées précédemment relèvent de la méthodologie générale des « contrats ouverts ». Selon l’Open Contracting Partnership, un organisme sans but lucratif axé sur la transparence et l’efficacité du gouvernement en matière de passation de marchés, la passation de marchés ouverts consiste à publier et à utiliser des données ouvertes, accessibles et opportunes sur les marchés publics afin d’inciter les citoyens et les entreprises à résoudre les problèmes et à obtenir des résultats. L’OCP offre une bibliothèque de ressources bien garnie.

[**Free Procurement Toolkit for City & Utility Officials**](https://the-atlas.com/procurement-toolkit/): Une boîte à outils pour aider les responsables gouvernementaux à élaborer un énoncé clair du problème et à choisir l’approche d’approvisionnement la plus appropriée pour le problème qu’ils veulent résoudre.

**FVH Application Evaluator: An Open Source Software Tool for Comparing Bids**: L’agence finlandaise d’innovation publique Forum Virium Helsinki a développé un logiciel libre qui facilite les marchés publics. À la fois outil et processus, le FVH Application Evaluator est un outil clair, complet, reproductible et facile à adapter à différents types de marchés publics. L’objectif de cet outil est de rendre l’évaluation et la comparaison des propositions faciles et agiles. Il comprend une couverture graphique qui donne rapidement un aperçu visuel.[[48]](#footnote-48) Les administrations publiques peuvent utiliser le code, le répliquer et y contribuer via GitHub.

[**Exemple de demande d’information (ville de Boston)**](https://drive.google.com/file/d/1GJUo6kR9zWmxF4835kHvSq0Rq7QuMLI-/view?usp=sharing): Un exemple de demande d’information de la ville de Boston sollicitant des soumissions afin d’évaluer l’intérêt et d’obtenir des renseignements sur les solutions potentielles pour leur initiative de données ouvertes.

[**Échange de ressources ouvertes du gouvernement du Canada :**](https://canada-ca.github.io/ore-ero/en/index.html) Un référentiel de ressources pour les organismes gouvernementaux, dont une section pour les logiciels libres.

[**UnTools :**](https://untools.co/) Une collection d’outils et de cadres de réflexion pour vous aider à résoudre des problèmes, à prendre des décisions et à comprendre des systèmes.

# Module 4 : Sécurité et mise à niveau des logiciels libres

## Objectifs d’apprentissage

Ce module vous permettra de vous familiariser avec :

* Les considérations liées à la sécurité et à la mise à niveau associées aux logiciels propriétaires conventionnels et aux logiciels libres.
* Les principaux défis auxquels les organisations sont confrontées dans la mise à niveau et la sécurisation de leurs solutions logicielles libres.
* Les stratégies pour assurer une gestion et une mise à niveau réussies des logiciels libres.

## Éléments clés

* **Les logiciels à risque zéro n’existent pas : Tous les logiciels, qu’il s’agisse de logiciels propriétaires ou de logiciels libres, comportent des sous-modules et présentent des risques devant être gérés.** En fait, la plupart des entreprises de logiciels propriétaires conventionnels utilisent le code source ouvert dans leurs produits et participent à la mise à niveau des logiciels libres.
* **Les logiciels libres ne sont pas intrinsèquement moins sûrs que les logiciels propriétaires conventionnels**. Comme tout logiciel, les logiciels libres comportent certains risques inévitables, mais cela ne doit pas vous décourager d’envisager leur utilisation au sein de votre organisation. La meilleure solution est de procéder à une évaluation approfondie des risques et d’élaborer des plans appropriés pour les atténuer.
* **Plus il y a d’intervenants qui utilisent un logiciel libre, plus il y a de chances que les problèmes de mise à niveau critiques et les failles de sécurité soient résolus.** Les logiciels libres appuyés par une grande collectivité active d’utilisateurs et de développeurs sont plus susceptibles de recevoir régulièrement des mises à jour et des rustines de sécurité. Toutefois, si vous utilisez un logiciel libre qui n’est pas largement appuyé par une collectivité d’utilisateurs, vous devrez compenser en vous assurant que vous disposez des capacités internes appropriées (ou que vous êtes en mesure de faire appel à une assistance externe) pour surveiller les vulnérabilités de sécurité.
* **Ne créez pas de « bifurcation » (fork) si vous n’êtes pas en mesure d’assurer la mise à niveau; plus votre organisation crée de codes personnalisés, plus vous devrez effectuer de mises à niveau**. Si vous créez une instance locale indépendante (en bifurquant le code), les mises à jour et les rustines de sécurité deviendront votre responsabilité – et deviendront de plus en plus difficiles, car votre version s’éloignera de la version initiale au fil du temps.

## Section 1 : Introduction à la sécurité et à la mise à niveau des logiciels libres

### En quoi consistent la sécurité et la mise à niveau des logiciels libres?

De l’avis d’un expert en sécurité logicielle, contrairement au vin et au fromage, les logiciels ne s’améliorent pas avec l’âge – en fait, leur niveau de sécurité diminue avec le temps. Cela est dû à l’obsolescence des logiciels.[[49]](#footnote-49) Lorsqu’il s’agit de logiciels libres, la sécurité et la mise à niveau vont de pair.

* La **sécurité des logiciels libres** consiste à préserver la confidentialité, l’intégrité et la disponibilité des ressources d’information afin d’assurer la réussite des opérations.[[50]](#footnote-50)
* La **mise à niveau des logiciels libres** consiste à maintenir le bon fonctionnement des logiciels et à éviter qu’ils ne deviennent obsolètes.

La mise à niveau est essentielle et le fait de ne pas investir suffisamment dans cette activité entraîne des « coûts cachés » : coûts *directs* liés à l’exploitation des failles de sécurité et à la perte de fonctionnalité qui en résulte, et coûts *indirects* résultant (entre autres) de la perte de main-d’œuvre qualifiée et du ralentissement de la croissance et de l’innovation.[[51]](#footnote-51)

### Sécurité et mise à niveau des logiciels propriétaires et des logiciels libres

On croit souvent à tort que les logiciels libres sont moins sûrs que les logiciels propriétaires conventionnels. Dans le cas des logiciels propriétaires, la mise à niveau est la responsabilité du fournisseur. En outre, dans le cas des logiciels propriétaires conventionnels, les contrats de service comprennent généralement des assurances sous forme de clauses de responsabilité et de garantie. C’est ce qui donne l’impression que ces logiciels sont plus sûrs.

En revanche, dans le cas des logiciels libres, la responsabilité de la mise à niveau est plus diffuse et repose en fin de compte sur la collectivité des utilisateurs. Toutefois, cela ne signifie pas que les logiciels libres sont moins sûrs. Selon un rapport, 89 % des dirigeants de grandes entreprises informatiques estiment que les logiciels libres sont aussi sûrs que les logiciels propriétaires.[[52]](#footnote-52) En bref, *les modèles de licence n’ont rien à voir avec la sécurité.*[[53]](#footnote-53)

### Les logiciels et la vie privée à l’ère numérique

Les gouvernements, de par leur nature même, génèrent et utilisent de grandes quantités de renseignements personnels sur leurs résidents afin de gérer des programmes et des services et de prendre des décisions stratégiques éclairées.

De nos jours, la quasi-totalité des renseignements permettant d’identifier une personne se présentent sous forme numérique, ce qui rend leur gestion beaucoup plus efficace tout au long de leur cycle de vie à l’aide d’un ensemble d’applications logicielles, mais les rend également plus vulnérables aux attaques. Même les renseignements ne permettant pas d’identifier une personne peuvent être rassemblés pour établir un profil précis de celle-ci, notamment sa localisation, ses habitudes et ses préférences.[[54]](#footnote-54)

La vie privée est considérée comme un élément fondamental de la liberté dans notre société.[[55]](#footnote-55) Les renseignements personnels identifiables stockés dans un endroit mal sécurisé – que ce soit sur un serveur local ou dans le nuage – sont vulnérables aux atteintes à la vie privée. C’est pourquoi la protection des renseignements, y compris les renseignements permettant d’identifier une personne, doit être une priorité essentielle lorsqu’on envisage la sécurité des logiciels propriétaires conventionnels et des logiciels libres.

## Section 2 : Enjeux et concepts clés pour la mise à niveau et la sécurisation des logiciels libres

### Risques sur le plan de la sécurité

Plusieurs aspects des logiciels libres peuvent les rendre vulnérables aux risques de sécurité :

* **Accès déraisonnable :** Le code source ouvert est accessible à tous, ce qui permet aux auteurs de cybermenace de manipuler le code à des fins malveillantes.[[56]](#footnote-56)
* **Lacunes en matière d’assurance de la qualité :** Il n’est pas toujours certain que des experts ont mené des tests rigoureux visant à garantir la qualité du produit pendant toutes les phases de développement du logiciel libre ni que ceux qui ont examiné le code ont minutieusement vérifié toutes les fonctions de sécurité.[[57]](#footnote-57)
* **Manque de soutien :** Les logiciels libres pris en charge par une grande collectivité active d’utilisateurs et de développeurs sont plus susceptibles de recevoir régulièrement des mises à jour et des rustines de sécurité. Toutefois, si vous utilisez un logiciel libre qui n’est pas largement appuyé par une collectivité d’utilisateurs, vous devrez compenser en vous assurant que vous disposez des capacités internes appropriées (ou que vous pouvez faire appel à une assistance externe) pour surveiller les vulnérabilités de sécurité.

### Les limites du modèle de soutien de la collectivité du logiciel libre

La plupart des collectivités de logiciels libres actives mettent régulièrement à jour les logiciels et publient des rustines pour les vulnérabilités connues. Cependant, des problèmes peuvent survenir si personne ne met à jour le logiciel, soit parce que le logiciel est trop nouveau ou trop beau pour attirer une collectivité de développeurs suffisamment importante, soit parce que la collectivité existante n’a pas maintenu un projet de logiciel libre.

Une attaque de cybersécurité qui se traduit par une fuite de données contenant des données sensibles peut avoir une incidence très négative sur la confiance au sein d’une organisation gouvernementale. En pareil cas, il se pourrait que le fait d’attendre que des mises à jour et des rustines de sécurité soient développées par la collectivité de logiciels libres ne soit pas une option. Les organisations gouvernementales doivent plutôt être en mesure d’évaluer et de résoudre rapidement le problème et, pour ce faire, elles doivent disposer d’une certaine expertise interne en ce qui concerne les modules ou logiciels libres utilisés par l’organisation.[[58]](#footnote-58)

### Approches conventionnelles en matière de gestion des risques

La quasi-totalité, sinon la totalité des organisations gouvernementales ont déjà mis en place un cadre de sécurité informatique. Une politique de sécurité informatique organisationnelle établit les objectifs généraux d’une organisation en matière de sécurité. Elle décrit de quelle façon les employés travaillent et collaborent sur les réseaux de l’organisation tout en respectant la position de l’organisation en matière de gestion des risques liés à la sécurité.

Dans de nombreux cas, les mesures mises en place pour atténuer les risques de sécurité découlant des logiciels propriétaires conventionnels couvrent également les logiciels libres.[[59]](#footnote-59) En fait, les pratiques actuelles d’évaluation des risques peuvent être appliquées aux logiciels libres sécurisés en traitant les vulnérabilités et en minimisant leur incidence.

Il est plus que probable que votre organisation ait adopté une approche en matière de gestion des risques informatiques et qu’elle ait un plan en place. Même s’il existe de multiples versions de la manière d’aborder la gestion des risques, une fois que vous avez identifié un risque, il existe quatre solutions de base : éviter, atténuer, transférer ou accepter.

Le problème avec les logiciels propriétaires conventionnels est qu’il peut être difficile d’identifier les risques dès le départ, car ils peuvent être cachés derrière le discours commercial du fournisseur et les petits caractères du contrat de licence.

### Gestion des risques liés aux logiciels libres

La gestion des risques liés aux logiciels libres peut être abordée de manière similaire. Un cadre récemment publié par Google identifie trois domaines problématiques largement indépendants avec un objectif concret : connaître les vulnérabilités de votre logiciel libre, empêcher l’ajout de nouvelles vulnérabilités et corriger (ou supprimer) les vulnérabilités.[[60]](#footnote-60)

Pour connaître vos vulnérabilités, votre équipe doit être en mesure de saisir des métadonnées précises sur les vulnérabilités à partir de toutes les sources de données disponibles, de préférence en suivant les normes du secteur pour simplifier les tâches de suivi. À l’heure actuelle, il est nécessaire de disposer d’un meilleur outil pour comprendre rapidement les vulnérabilités nouvellement découvertes. Toutefois, le fait de comprendre les risques inhérents aux nouvelles dépendances est un bon début. La correction des vulnérabilités n’est pas toujours une tâche facile, car la résolution directe des vulnérabilités nécessite une expertise interne que votre équipe n’a peut-être pas. Toutefois, si votre organisation doit s’attaquer à une vulnérabilité spécifique, le meilleur plan d’action est de travailler de bas en haut dans la chaîne de dépendance pour que le logiciel soit corrigé.

À mesure que de nouvelles menaces apparaissent, de nouvelles vulnérabilités se manifestent, ce qui signifie que la gestion des risques est une activité permanente qui doit être constamment mise à jour, car les mesures existantes peuvent devenir inefficaces. Une façon de suivre les risques et la tolérance au risque d’une organisation est d’utiliser une grille de risques pouvant aider les organisations à évaluer leur probabilité de risques et à comprendre et modéliser les relations entre les menaces, les vulnérabilités et les mesures devant être appliquées.[[61]](#footnote-61)

En fait, les vulnérabilités des logiciels libres doivent être abordées comme un prolongement des objectifs généraux en matière de sécurité informatique et des processus de rapports d’une organisation plutôt que comme une tâche entièrement nouvelle. N’oubliez pas que la compréhension de vos risques contribuera à créer une culture consciente des risques au sein de l’organisation, tout en examinant comment les risques technologiques – y compris les risques liés aux logiciels libres – s’harmonisent avec les objectifs opérationnels.

### Dépendances logicielles : Un défi pour la sécurité et la mise à niveau

Tous les logiciels sont construits à partir d’autres logiciels, d’une façon ou d’une autre. Les développeurs peuvent créer de nouvelles applications beaucoup plus rapidement en pointant vers un code qui a déjà été créé par un tiers que s’ils devaient créer un nouveau code à partir de rien. En fait, une enquête menée en 2017 a révélé que, sur plus de 1 100 bases de code commerciales, 96 % des applications numérisées contenaient des composants de logiciels libres, avec une moyenne de 257 composants par application. »[[62]](#footnote-62)

Lorsque le « programme A » a besoin du « programme B » pour fonctionner, on parle de **dépendance logicielle**. Les dépendances peuvent être directes (c’est-à-dire que le programme A dépend directement du programme B pour fonctionner) ou transitives (c’est-à-dire que le programme A dépend du programme B, qui dépend à son tour du programme C pour fonctionner).[[63]](#footnote-63) Les dépendances sont en grande partie inévitables, mais celles qui échappent à votre contrôle présentent un risque.

Une défaillance de dépendance peut se produire lorsqu’une bibliothèque de codes ou un paquetage critique pour le fonctionnement d’un logiciel donné est modifié ou supprimé. L’incident de 2016 concernant le « coussinet gauche » est un exemple qui montre à quel point les conséquences d’une défaillance de la dépendance peuvent être importantes :

* En 2016, un paquet de codes connu sous le nom de « [coussinet gauche](https://www.npmjs.com/package/left-pad) »'[[64]](#footnote-64) ne comportant que 11 lignes de codes a été retiré par son développeur du registre NPM où il avait été hébergé. Sa suppression a provoqué une cascade de défaillances de dépendance, entraînant des pannes de logiciels dans le monde entier.[[65]](#footnote-65)
* Selon un blogueur, le problème plus vaste illustré par cet échec de dépendance est que tant de paquets et de projets ont pris une dépendance pour une simple fonction de chaîne de remplissage gauche, plutôt que leurs développeurs prennent deux minutes pour écrire eux-mêmes une fonction aussi basique.[[66]](#footnote-66)

En bref, plus les dépendances d’un logiciel échappent à votre contrôle, plus le risque de sécurité est élevé et plus les chances que quelque chose se brise augmentent. La première étape vers la gestion des dépendances et l’atténuation des incidences en aval sur la sécurité et la mise à niveau de votre logiciel libre est simplement d’en être conscient. À cette fin, de nombreux développeurs de logiciels incluent désormais une nomenclature logicielle (SBOM) qui répertorie tous les composants logiciels et tiers présents dans un code base.[[67]](#footnote-67)

## Section 3 : Stratégies pour la mise à niveau et la sécurisation des logiciels libres

### Exploiter le cadre de votre organisation en matière de gouvernance des données

Les organisations gouvernementales devraient avoir une compréhension approfondie des données qu’elles recueillent, utilisent et conservent et de la manière dont elles les aident à atteindre leurs objectifs. La mise en place d’un cadre de gouvernance des données à l’échelle de l’organisation formalise cette compréhension et fournit un point de référence pour toutes les activités liées aux données, qu’elles visent des logiciels propriétaires ou des logiciels libres.

Les considérations spécifiques liées à la gouvernance des données dans le cas des logiciels libres incluent des politiques et des procédures afin de veiller à ce que les renseignements personnels identifiables soient tenus à l’écart des codes bases libres.[[68]](#footnote-68) Ces considérations incluent également le fait de « conserver les données sensibles comme les justificatifs d’identité en lieu sûr et séparément du code source » et « éviter d’entreposer des clés et d’autres documents de nature délicate dans des systèmes non approuvés à cette fin.[[69]](#footnote-69)»

### Faire de la confidentialité et de la sécurité une priorité

Les dirigeants du secteur public devraient promouvoir une culture axée sur la protection des données et inciter leurs équipes à adopter des pratiques de protection de la vie privée et de sécurité dès la conception[[70]](#footnote-70) afin de veiller à ce que tous les logiciels – qu’il s’agisse de logiciels propriétaires ou de logiciels libres – soient exempts de vulnérabilités susceptibles d’entraîner une violation de la vie privée ou une perte de fonctionnalité.[[71]](#footnote-71)  Pour ce faire, vous pouvez :

* Réviser périodiquement les politiques afin de veiller à ce qu’elles s’harmonisent avec les menaces en cours.
* Organiser des séances de formation périodiques à l’intention du personnel sur des sujets liés à la sécurité informatique.

Ces pratiques devraient être revues et mises à jour régulièrement afin qu’elles reflètent les menaces en cours. Et, bien que cela ne soit pas spécifique aux logiciels libres, il est important de disposer d’un plan pour assurer la continuité des activités et évaluer comment l’organisation réagira en cas de violation de la sécurité ou de perte de données pour des raisons non intentionnelles (comme la perte d’un ordinateur).

### Établir des politiques et des procédures pour l’examen périodique du code et la divulgation des vulnérabilités

Si votre équipe compte des développeurs qui prennent en charge des applications logicielles libres utilisées par votre organisation, il est important que vous établissiez des politiques et des procédures appropriées pour examiner le code source afin de détecter les risques de sécurité potentiels. L’examen périodique des codes « augmente la probabilité de détecter les bogues, les vulnérabilités en matière de sécurité et réduit le risque d’engager des données sensibles ».[[72]](#footnote-72) L’examen des codes vous permet également d’évaluer le niveau de dépendance de votre logiciel libre et de minimiser l’exposition aux défaillances de dépendance.

Un certain nombre d’outils et de services peuvent aider votre équipe à évaluer rapidement les logiciels libres pour détecter les vulnérabilités de sécurité connues et les problèmes de dépendance potentiels, notamment l’outil [Open SSF Scorecard](https://securityscorecards.dev/), qui permet de vérifier s’il existe des vulnérabilités affectant différentes parties de la chaîne d’approvisionnement des logiciels, notamment le code source, la construction, les dépendances, les tests et la mise à niveau du projet ».[[73]](#footnote-73)

### Ne transférez pas un code si vous n’êtes pas en mesure d’en assurer la mise à niveau

La mise à niveau et la sécurisation de tout type de logiciel, qu’il s’agisse de logiciels propriétaires ou de logiciels libres, exigent la gestion d’un grand nombre de composants libres et de dépendances afin d’atténuer les risques en matière de sécurité. C’est plus facile lorsque votre logiciel est un élément actif de la collectivité de logiciels libres.

Toutefois, si votre organisation crée une instance locale indépendante (en bifurquant le code), les mises à jour et les rustines de sécurité relèvent de votre propre responsabilité – et elles deviennent de plus en plus difficiles, car votre version s’éloigne de la version initiale au fil du temps. En bref : pas de bifurcation si vous n’êtes pas en mesure d’assurer la mise à niveau.

Le Guide pour l’utilisation de logiciels libres du gouvernement du Canada recommande ce qui suit : « Dans la mesure du possible, utiliser des logiciels libres sans modification ou les contribuer au projet » et « Utiliser la configuration et personnaliser le logiciel avec des modules, des plugiciels ou des extensions et rendre ceux-ci disponibles à la collectivité ».[[74]](#footnote-74)

## Ressources supplémentaires

[**2022 Open Source Security and Analysis Report**](https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html): Ce rapport de Synopsys examine les vulnérabilités et les conflits de licence trouvés dans plus de 2 400 bases de code dans 17 secteurs d’activité. Le rapport propose des recommandations pour aider les équipes chargées de la sécurité, du droit, des risques et du développement à mieux comprendre le paysage de la sécurité et des risques qui accompagnent le développement et l’utilisation des logiciels libres.

[**OpenSSF Scorecard**](https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html): Cet outil vérifie les vulnérabilités affectant différentes parties de la chaîne d’approvisionnement des logiciels, notamment le code source, la construction, les dépendances, les tests et la mise à niveau du projet. Chaque mesure automatisée renvoie une note sur 10 et un niveau de risque. Le niveau de risque ajoute une pondération au score, et cette pondération est compilée dans un score unique et global. Ce score permet de donner une idée de la posture de sécurité globale d’un projet. Outre les scores, l’outil fournit des pistes de solution pour vous aider à résoudre les problèmes et à renforcer vos pratiques de développement.

[**Concise Guide for Evaluating Open Source Software**](https://github.com/ossf/wg-best-practices-os-developers/blob/main/docs/Concise-Guide-for-Evaluating-Open-Source-Software.md#readme): Ce guide, destiné aux développeurs de logiciels, fournit une série de questions techniques à régler avant d’adopter des dépendances ou des outils logiciels libres, afin d’évaluer leur sécurité et leur viabilité.

# Module 5 : Collaboration en ce qui concerne les logiciels libres

## Objectifs d’apprentissage

Ce module vous permettra de vous familiariser avec :

* Les questions interjuridictionnelles soulevées dans les modules précédents et les considérations clés lors de la formation d’équipes multijuridictionnelles autour d’un logiciel particulier, y compris le partage des ressources entre les différentes administrations ou ordres de gouvernements.
* La participation à une collectivité mondiale de gestion des codes bases.
* L’utilité de diffuser votre logiciel sous une licence libre.

## Éléments clés

* **La collaboration en ce qui concerne les logiciels libres présente des avantages pour tous les intervenants.** Le logiciel que vous développez, ou les adaptations que vous faites, seront également utiles à d’autres administrations. Les logiciels libres sont plus performants et plus sûrs à mesure que la collectivité qui les utilise se développe. Il est donc dans l’intérêt de chaque utilisateur de contribuer, et d’ajouter d’autres utilisateurs contributeurs.
* **Les organisations gouvernementales n’ont pas besoin de réinventer la roue.** Lorsque le code source est publié sous une licence libre, une administration peut procéder à de petites adaptations du logiciel afin de veiller à ce qu’il soit le mieux adapté aux cas d’utilisation, aux réglementations et aux normes locales. Lorsqu’un ensemble de fonctionnalités de base est partagé, le temps, les efforts et les ressources de chaque administration peuvent être consacrés à de petites adaptations en fonction du contexte local (plutôt que de créer un ensemble complet de fonctionnalités de base à partir de rien).
* **Les gouvernements de même niveau sont des collaborateurs naturels lorsqu’il s’agit de concevoir, d’obtenir et de mettre à niveau des logiciels libres.** Ils sont chargés de fournir un ensemble similaire de services et ont un ensemble similaire de besoins (internes, administratifs et externes, axés sur les citoyens). Ils sont donc des collaborateurs naturels lorsqu’il s’agit de concevoir, d’obtenir et de mettre à niveau des logiciels libres.
* **Les administrations collaborent déjà dans le cadre de différents réseaux et groupes de pairs bien établis qui peuvent être activés en tant que collectivités de régie de logiciels libres.** Les plateformes de contrôle de la version distribuées (comme GitHub) permettent aux utilisateurs de contribuer ou de suggérer des modifications de manière structurée et collaborative.

## Section 1 : Collaboration entre administrations

### Avantages de la collaboration

**La collaboration sur les logiciels libres présente des avantages pour tous les intervenants.** Le logiciel que vous concevez, ou les adaptations que vous faites, seront également utiles à d’autres administrations. Les logiciels libres sont plus performants et plus sûrs à mesure que la collectivité qui les entoure s’agrandit. Chaque utilisateur bénéficie des améliorations continues apportées aux logiciels (sous la forme de mises à jour, de nouvelles fonctionnalités ou d’améliorations de la sécurité). Il est donc dans l’intérêt de chaque utilisateur de contribuer, et d’ajouter d’autres utilisateurs contributeurs.

Un autre avantage de la collaboration pour les organisations gouvernementales réside dans le fait qu’elles n’ont pas besoin de réinventer constamment la roue. Lorsque le code source est publié sous une licence libre, une administration peut procéder à de petites adaptations du logiciel afin de veiller à ce qu’il soit le mieux adapté aux cas d’utilisation, aux réglementations et aux normes locales. Lorsqu’un ensemble de fonctionnalités de base est partagé, le temps, les efforts et les ressources de chaque administration peuvent être consacrés à de petites adaptations pour adapter le logiciel à leur contexte local (plutôt que de construire un ensemble complet de fonctionnalités de base à partir de rien). Le partage des connaissances et des ressources aide les partenaires moins expérimentés en matière de logiciels libres à renforcer leurs capacités.

### Avec qui pouvez-vous collaborer?

Les gouvernements et les organismes du secteur public qui souhaitent utiliser des logiciels libres devraient se joindre à des réseaux existants ou établir des relations axées sur la collaboration avec des administrations comparables. Ces réseaux peuvent s’intéresser à un logiciel spécifique ou miser plus largement sur l’utilisation des logiciels libres dans le secteur public. Au sein de ces collectivités, les utilisateurs peuvent partager les défis, les occasions, les capacités, les ressources et les processus (voir encadré 5.1).

Les gouvernements de même niveau – qu’il s’agisse de gouvernements municipaux ou provinciaux – sont des collaborateurs naturels lorsqu’il s’agit de concevoir, d’obtenir et de mettre à niveau des logiciels libres. Ils sont chargés de fournir un ensemble similaire de services et ont un ensemble similaire de besoins (internes, administratifs et externes, axés sur les citoyens). Et lorsqu’il s’agit d’acquérir un nouveau logiciel, une approche d’achat groupé permet à un groupe d’organismes gouvernementaux comparables de profiter d’économies d’échelle.

#### Encadré 5.1 : Plateforme de gestion des données ouvertes CKAN

La plateforme de gestion des données ouvertes CKAN est un excellent exemple de collaboration entre administrations. Conçue par l’Open Knowledge Foundation (OKFN), organisme à but non lucratif, pour le gouvernement du Royaume-Uni, cette plateforme est utilisée par les gouvernements nationaux et régionaux du monde entier comme base de leurs portails de données ouvertes, notamment le [gouvernement du Canada](https://open.canada.ca/en/open-data) , le [gouvernement de la Colombie-Britannique](https://www2.gov.bc.ca/gov/content/data/bc-data-catalogue) et le [gouvernement de l’Ontario](https://data.ontario.ca/about).

Étant donné que les améliorations du code sont partagées, chacun des [115 gouvernements et institutions](https://ckan.org/government) bénéficie de la production d’un meilleur produit à un coût réduit pour les consommateurs individuels. En plus d’aider les gouvernements à développer et à mettre en œuvre des instances de la plateforme adaptées à leurs besoins1, l’importante collectivité d’utilisateurs et de développeurs de CKAN génère et diffuse continuellement des connaissances via GitHub, des listes de diffusion et des canaux IRC.

1 Colombie-Britannique. Ministère de l’emploi, de la relance économique et de l’innovation, [*About the Catalogue Open Source Project*](https://www2.gov.bc.ca/gov/content/data/bc-data-catalogue/about-catalogue-open-source-project), consultation le 3 novembre 2022.

### Obstacles à l’utilisation des logiciels libres dans les différentes administrations

Chacun des aspects abordés dans les modules précédents peut présenter des défis spécifiques lorsqu’il s’agit de collaborer à des projets de logiciels libres entre administrations ou entre organismes publics.

* **Défis liés à l’organisation et aux capacités du personnel :** La capacité numérique peut varier considérablement d’un ordre de gouvernement ou d’une institution publique à l’autre, notamment au niveau municipal.[[75]](#footnote-75) Les administrations n’ont pas toutes une équipe de logiciels libres ou un responsable de produit; il peut arriver que les petites municipalités n’aient qu’une seule personne responsable de l’informatique. Lorsqu’il s’agit de gérer des projets de logiciels libres, les compétences requises sont de nature politiques et non techniques. Les développeurs, les utilisateurs et les responsables de la mise à niveau des logiciels ne sont peut-être pas les mieux placés pour gérer des questions telles que la responsabilité et la négociation des coûts partagés. De même, les responsables de produits ne comprennent pas nécessairement la dynamique et les implications des aspects techniques de la mise à niveau des logiciels.
* **Défis en matière d’approvisionnement, de budgétisation et de passation de marchés :** Les processus, modèles et normes de passation de marchés standard accordent la priorité aux produits disponibles sur le marché, auprès de fournisseurs établis, et, dans le cas de la prestation de services techniques, au développement de logiciels en cascade. Les gouvernements peuvent être enfermés dans des contrats à long terme avec les fournisseurs, ce qui ne leur permet pas d’envisager des collaborations en vue de solutions logicielles libres de rechange. En outre, certaines organisations refusent les logiciels qui ne sont pas d’ici et négligent de regarder ce que d’autres organisations gouvernementales ont fait. Un mantra courant dans la collectivité des logiciels libres est que la roue a déjà été inventée. Cela s’applique au logiciel complet, ainsi qu’aux sous-modules. Les gouvernements peuvent, et doivent, découvrir ce qui a fonctionné ailleurs.
* **Défis en matière de sécurité et de mise à niveau :** Les projets de logiciels libres entrepris par un groupe d’intervenants gouvernementaux, en particulier pour des cas d’utilisation relativement spécialisés, peuvent être négligés et devenir obsolètes si une structure de gouvernance n’est pas mise en place pour définir les responsabilités en matière de mise à niveau continue et si un financement n’est pas assuré pour la prestation de services de mise à niveau. En outre, les différences de politique en matière de services numériques entre les administrations (p. ex., le consentement de l’utilisateur et les exigences de protection des données en vertu du GDPR dans l’UE par rapport à la politique du Canada) peuvent créer des obstacles au développement de logiciels libres répondant aux besoins en matière de gestion des risques de toutes les parties dans le cadre d’une collaboration internationale.

### Fusion du code de bifurcation

La fusion du code de bifurcation à la branche principale est un moment difficile dans le cadre d’une collaboration concernant un logiciel libre. La pratique exemplaire, dans le contexte du développement de logiciels libres, consiste à effectuer de petites modifications graduelles, minimisant ainsi le risque de briser les dépendances. Lorsque les mesures prises par toutes les personnes travaillant sur le code sont synchronisées, les changements se répercutent les uns sur les autres avec un minimum de perturbations. Si quelqu’un crée une bifurcation et apporte de nombreuses modifications, puis souhaite finalement fusionner les bases de code, il peut y avoir des conflits, tant structurels que fonctionnels.

#### Encadré 5.2 : Le service de messagerie *Notify*

*Notify* est un logiciel libre que les organismes gouvernementaux participants peuvent utiliser pour envoyer facilement des milliers de courriels, de messages textes et de lettres à leurs contacts, en toute sécurité.1

Le Government Digital Service du Royaume-Uni a d’abord publié sa plateforme Notify sous forme de code source ouvert, ce qui a permis à d’autres gouvernements de l’adopter à leurs propres fins. Ainsi, le gouvernement du Canada a développé sa propre version appelée [GC Notification](https://notification.canada.ca/) en la modifiant pour qu’elle soit offerte dans plusieurs langues.2

1 Beeck Center for Social Impact and Innovation, [*Notify, Intergovernmental Software Collaborative*](https://softwarecollaborative.org/cooperatives/notify.html)*,* consultation le 18 octobre 2022.

2 Bryan Willey, [*On vous présente Notification*](https://numerique.canada.ca/2019/11/26/on-vous-pr%C3%A9sente-notification/), *Service numérique canadien* (blogue), le 26 novembre 2019.

Prenons l’exemple hypothétique d’une application open source telle que Notify (voir encadré 5.2) : l’application originale peut avoir été codée de manière à empêcher la modification des messages. Un forumeur peut ensuite créer une fonction de confiance basée sur le fait que les messages ne peuvent pas être modifiés. Cependant, dans l’intervalle, le code de la branche principale a été modifié pour permettre la modification des messages. Si le forumeur décide de fusionner à nouveau, il constatera que les fonctionnalités de la base de code originale ont changé, et que chaque branche a construit des fonctionnalités destinées aux utilisateurs sur la base de prémisses différentes. Cela crée un dilemme : *La base de code fusionnée doit-elle permettre la modification des messages (ce qui compromettrait les fonctions de confiance d’une branche) ou revenir à des messages non modifiables (ce qui compromettrait tous les outils et flux de travail impliquant des messages modifiables)?*

Pour éviter des situations comme celle décrite ci-dessus, le Guide pour l’utilisation de logiciels libres du gouvernement du Canada recommande ce qui suit au personnel : « Utiliser, dans la mesure du possible, des logiciels libres sans modification ou les contribuer au projet » et « Utiliser la configuration et personnaliser le logiciel avec des modules, des plugiciels ou des extensions et rendre ceux-ci disponibles à la collectivité ».[[76]](#footnote-76)

### Diffusion du code en tant que logiciel libre

Le gouvernement du Canada recommande aux organismes gouvernementaux qui ont développé leurs propres logiciels d’envisager la diffusion du code sous une licence de source ouverte, et fournit des directives pour le faire efficacement. « Il est recommandé que, lorsqu’ils ont le droit de le faire, les ministères publient tout le code source en tant que logiciel libre, que la solution logicielle soit (i) acquise en tant que logiciel libre, (ii) mise au point par les employés du GC à l’interne ou (iii) acquise au moyen des conditions des contrats d’approvisionnement où des conditions de licence appropriées ont été négociées ».[[77]](#footnote-77)

De nombreuses considérations juridiques entrent en jeu lors du choix d’une licence pour le projet de logiciel libre de votre organisation.[[78]](#footnote-78) Certains gouvernements (p. ex., la [Colombie-Britannique](https://github.com/bcgov/BC-Policy-Framework-For-GitHub/blob/master/BC-Open-Source-Development-Employee-Guide/Licenses.md)) ont publié des lignes directrices pour le choix d’une licence appropriée qui sont spécifiques à leur contexte législatif. Si, par exemple, votre projet est une adaptation ou un dérivé d’un projet existant, la pratique exemplaire consiste à utiliser la même licence que pour le projet original; s’il s’agit d’un nouveau projet, la licence que vous choisirez dépendra du résultat souhaité et de la licence de tout logiciel tiers utilisé dans le cadre de votre projet.[[79]](#footnote-79)

## Section 2 : Stratégies de collaboration en ce qui concerne les logiciels libres

Malgré les défis potentiels, il est possible pour différentes administrations de collaborer de façon productive en ce qui concerne les logiciels libres.

### Construire intentionnellement une structure de gouvernance

Les collaborations qui démarrent sans une structure de gouvernance claire risquent de connaître des difficultés, en particulier s’il y a un trop grand nombre de partenaires. Il peut alors être difficile de s’entendre sur le modèle de collaboration suivi et de déterminer les rôles appropriés pour chaque partenaire.[[80]](#footnote-80)

Lorsque vous envisagez une collaboration, prenez le temps de vous harmoniser avec les partenaires potentiels. L’identification d’un besoin ou d’un objectif commun pour une application de logicielle libre est l’une des premières étapes d’une collaboration. En entamant des conversations avec vos homologues d’autres organismes gouvernementaux ou du secteur public, vous aurez une meilleure compréhension de leurs besoins, des points d’entente potentiels et de la forme que devrait prendre votre collaboration.[[81]](#footnote-81)

Une fois ce premier contact établi, les partenaires peuvent entrer dans les détails, comme la planification et la documentation de la structure de gouvernance proposée, les principes de travail en commun et l’harmonisation avec la manière de financer leur travail commun.[[82]](#footnote-82) Une façon d’aborder ces conversations est d’utiliser un outil comme le *Governance Game* mis au point par la Foundation for Public Code, un jeu interactif sur la gouvernance d’un code public. Ce jeu aide les participants à réfléchir à ce que signifie la gouvernance pour un code base, ainsi qu’à la complexité de la gouvernance, et met en évidence les aspects à prendre en compte lors de la mise en place.[[83]](#footnote-83)

### Créer des communautés de pratique en ce qui concerne les logiciels libres

Les champions du logiciel libre au sein des organismes gouvernementaux peuvent former des communautés de pratique en ce qui concerne les logiciels libres. L’exploitation de réseaux préexistants de groupes de pairs composés de fonctionnaires responsables de programmes ou de services similaires peut constituer une façon efficace de développer et de partager des logiciels libres pour certains domaines d’intérêt ou cas d’utilisation (par exemple, la planification des transports, la gestion des actifs). Les communautés de pratique axées sur les logiciels libres de façon générale peuvent être utiles pour échanger des connaissances sur les logiciels libres à un niveau organisationnel. Les communautés de pratique peuvent gérer des référentiels partagés de modules et de ressources de logiciels libres.

### Créer un accord d’achat groupé ou participer à un tel accord

Il n’est pas toujours possible de se procurer ou de développer des logiciels libres par ses propres moyens, surtout dans le cas des petites et moyennes municipalités. Avec les achats groupés (également connus sous le nom d’achats coopératifs), vous partagez le coût avec d’autres.[[84]](#footnote-84) Les achats groupés augmentent l’efficacité d’un processus d’approvisionnement et réduisent les coûts pour les organisations participantes. Les avantages des achats groupés pour le secteur gouvernemental sont les suivants :

* Prix moins élevés
* Mesures de qualité
* Délais de livraison moins longs
* Processus rationalisés
* Accès à des modes de financement alternatifs
* Coopération technique
* Transparence et partage d’information accrus

Les avantages s’étendent également aux fournisseurs :

* Réutilisation du code existant/partagé
* Réduction des coûts de transaction
* Prévisibilité de la demande de services (développement et mise à niveau des fonctionnalités)
* Financement anticipé pour le fonds de roulement et la prévision de la demande[[85]](#footnote-85)

Créer une coopérative d’achat ou se joindre à une telle coopérative.

Il existe deux approches principales pour les achats groupés, soit la sollicitation conjointe (coordination d’un processus partagé dès le départ) ou jumelage (partage d’un travail déjà effectué par plusieurs organismes publics).[[86]](#footnote-86) Voici les principales étapes à suivre pour une sollicitation conjointe :

1. La coopérative est constituée lorsqu’une ou plusieurs parties identifient un besoin commun susceptible d’être satisfait par une coopération et signent un accord écrit de coopération.
2. La partie principale sollicite des propositions et attribue un ou plusieurs contrats.
3. Le contrat est disponible pour l’utilisation.
4. Les entités participantes signent un accord (addendum de participation selon la NASPO/WSCA) dans le cadre du ou des contrats spécifiques. Cela est nécessaire pour que les exigences législatives des utilisateurs soient incluses et pour que l’entité dirigeante puisse assurer une administration efficiente.[[87]](#footnote-87)

Le jumelage est une approche plus ponctuelle. Aux États-Unis, il existe des outils tels que [CoProcure](https://www.coprocure.us/) qui peuvent vous aider à trouver des produits et des sollicitations géographiquement et topiquement pertinents. Toutefois, il n’existe pas de « guichet unique » pour les possibilités d’achats coopératifs au Canada; les occasions de profiter des contrats d’autres organisations se présenteront probablement par l’entremise de vos liens avec les réseaux de pairs existants.

### Documenter correctement les projets de logiciels libres

Vous pouvez investir beaucoup de temps et d’efforts dans le développement d’un logiciel libre, mais si personne d’autre ne peut comprendre ce qu’est le logiciel et ce qu’il fait, à un niveau élevé, et comment les modules fonctionnent, à un niveau technique, les chances qu’il soit adopté par d’autres que votre équipe sont faibles. Pour faciliter l’adoption par d’autres des logiciels libres que vous développez, il est essentiel de prendre le temps de les documenter correctement.

Envisagez l’utilisation de la norme pour le code public, un ensemble de critères qui aide les organismes publics à développer et à maintenir ensemble des logiciels et des politiques.[[88]](#footnote-88) La norme sert de cadre aux gouvernements et aux organismes publics en vue de l’élaboration de leurs propres projets de logiciels libres d’une manière transparente, responsable et compréhensible par les autres.

Ressources supplémentaires

[**Introduction to the Standard for Public Code**](https://citybycity.academy/course/standards-for-smart-cities) (CITYxCITY Academy) : Ce cours gratuit aborde le thème du code public, explique pourquoi celui-ci est important et présente le processus par lequel un logiciel et un code de politique peuvent devenir un code public certifié.

[**Intergovernmental Software Collaborative (Beeck Centre for Social Impact)**](https://softwarecollaborative.org/): Cette communauté de pratique formée de représentants de tous les ordres de gouvernements des États-Unis partage de l’information, des études de cas et les pratiques exemplaires pour établir des collaborations intergouvernementales en matière de logiciels.

**[Open Source Guides : Leadership and Governance (GitHub)](https://opensource.guide/leadership-and-governance/)**: Une ressource générale fournissant des conseils pour établir et formaliser les rôles et les structures de gouvernance au fur et à mesure que votre projet de logiciel libre se développe.

**[Co-procurement](https://www.coprocure.us/)**: Une plateforme américaine permettant de trouver les marchés publics de logiciels en cours ou de lancer un nouveau processus d’approvisionnement collaboratif.

# Glossaire

|  |  |
| --- | --- |
| Analyse du coût total/Analyse du coût réel | Cela comprend le coût total de la création ou de l’acquisition du logiciel, de sa mise en œuvre et de sa mise à niveau. L’analyse du coût total comprend les coûts directs et indirects, immédiats et à long terme (notamment les heures de travail du personnel, les frais de licence, etc.). |
| Bifurcation | Une duplication est une copie d’un dépôt que vous gérez. Les duplications vous permettent d’apporter des modifications à un projet sans affecter le dépôt d’origine. Vous pouvez extraire des mises à jour du dépôt d’origine ou envoyer des modifications à celui-ci avec des demandes de tirage. [Source : GitHub](https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks/about-forks) |
| Budgétisation modulaire | La budgétisation modulaire est une planification financière qui appuie la sous-traitance modulaire (voir ci-dessous). Un budget modulaire s’applique aux heures-personnes et aux autres coûts associés à un système discret. |
| Contrôle de la version distribuée | Dans le domaine du développement logiciel, le contrôle de la version distribuée (également connu sous le nom de gestion des versions) est une forme de contrôle où le code base complet, y compris son historique, se reflète dans l’ordinateur de chaque développeur. Contrairement au contrôle de la version centralisée, cela permet de gérer automatiquement les branchements et les fusions, d’accélérer la plupart des opérations (à l’exception du push et du pull), d’améliorer la capacité de travailler hors ligne et de ne pas dépendre d’un emplacement unique pour les sauvegardes. Git, le système de contrôle de la version le plus populaire au monde, est un système de contrôle de la version distribuée. Source : [Contrôle de la version distribuée – Wikipédia](https://en.wikipedia.org/wiki/Distributed_version_control) |
| Demande d’extraction | Les contributions à un référentiel de code source qui utilise un système de contrôle de la version distribuée sont généralement effectuées au moyen d’une **demande d’extraction**, également connue sous le nom de **demande de fusion**. Le contributeur demande au responsable du projet de retirer la modification du code source. Le responsable doit fusionner la demande d’extraction si la contribution doit faire partie de la base de données des sources.  Source : [Contrôle de la version distribuée – Wikipédia](https://en.wikipedia.org/wiki/Distributed_version_control#Pull_requests)  Voir également : Contrôle de la version distribuée |
| Demande de devis | Méthode de sollicitation utilisée pour obtenir des fournisseurs des renseignements sur les prix, les coûts, les livraisons et autres renseignements connexes. (Source : [U.S. General Services Administration](https://www.gsa.gov/small-business/small-business-resources/training-resources/rfp-rfi-and-rfq-what-are-the-differences)) |
| Demande de fusion | Voir : Demande de tirage |
| Demande de propositions (DP) | Une demande de propositions (DP) est à la fois le processus et la documentation utilisés pour solliciter des soumissions pour des solutions opérationnelles ou informatiques potentielles requises par une entreprise ou un organisme gouvernemental. Le document de demande de propositions décrit généralement un énoncé des exigences auxquelles doivent répondre les répondants potentiels qui souhaitent faire une offre pour fournir les solutions requises. Il peut s’agir de produits et/ou de services permettant de répondre aux exigences prévues. La documentation de l’appel d’offres couvre aussi généralement le processus d’achat correspondant, les critères d’évaluation, les modalités opérationnelles, les délais et les activités en cause, ainsi que les éléments que les soumissionnaires doivent inclure dans leur réponse à l’appel d’offres. (Source : [Gartner](https://www.gartner.com/en/information-technology/glossary/request-proposal-rfp)) |
| Demande de renseignements (RFI)/Avis de recherche de sources (SSN) | **Demande de renseignements/Avis de sources recherchées (SSN) :** Un outil d’étude de marché utilisé pour obtenir le prix, la livraison, les capacités, l’intérêt, etc. à des fins de planification. (Source : [U.S. General Services Administration](https://www.gsa.gov/small-business/small-business-resources/training-resources/rfp-rfi-and-rfq-what-are-the-differences)) |
| Dépôt central | Voir : Référentiel |
| Dépôt de la branche | Voir : Référentiel |
| Développement logiciel selon la méthode agile | La méthode agile est une approche itérative de la gestion de projet et du développement logiciel qui aide les équipes à fournir de la valeur à leurs clients plus rapidement et avec moins de maux de tête. Au lieu de tout miser sur un lancement « big bang », une équipe agile livre le travail par petites étapes. Les exigences, les plans et les résultats sont évalués en permanence, afin que les équipes disposent d’un mécanisme naturel pour réagir rapidement au changement. Le développement logiciel agile fait appel à un certain nombre de stratégies de gestion des processus, comme les réunions quotidiennes, les scrums, les parcours d’utilisateurs et d’[autres outils de base](https://agile.18f.gov/agile-fundamentals/). (Source : [Atlassian](https://www.atlassian.com/agile). Voir aussi le [Manifeste Agile](http://agilemanifesto.org/)) |
| DevOps | DevOps désigne un processus dans le cadre duquel des personnes travaillent ensemble pour concevoir, construire et livrer des logiciels sécurisés à la vitesse maximale. Les pratiques DevOps permettent aux développeurs de logiciels et aux équipes opérationnelles d’accélérer la livraison grâce à l’automatisation, à la collaboration, à la rétroaction rapide et à l’amélioration itérative. Issu d’une approche agile du développement logiciel, un processus de livraison DevOps étend l’approche interfonctionnelle de la création et de l’expédition d’applications de manière plus rapide et plus itérative. (Source : [GitLab](https://about.gitlab.com/topics/devops/)) |
| Énoncé/cadrage de problèmes | Le cadrage de problèmes est une méthode de résolution des problèmes qui vise à rallier toute l’équipe en vue de l’adoption d’une solution unique pour un projet en structurant les détails du problème de manière digeste et collaborative. Ainsi, lorsque votre équipe ne parvient pas à s’entendre sur une solution, utilisez cette méthode pour prendre du recul et vous harmoniser avec le problème à résoudre. (Source : [Atlassian](https://www.atlassian.com/team-playbook/plays/problem-framing)) |
| Fournisseur unique | La passation de marchés par l’entremise d’un processus non concurrentiel, également connue sous le nom de passation de marchés à fournisseur unique, consiste à solliciter une soumission auprès d’une seule source. Les achats à fournisseur unique doivent respecter des normes qui régissent la manière dont un organisme public peut justifier qu’un seul fournisseur est en mesure de fournir les services demandés. (Source : [US Department of Justice](https://www.ojp.gov/sites/g/files/xyckuh241/files/media/document/Sole_Source_FactSheet_C.pdf)) |
| Fractionnement des soumissions | Lorsqu’un employé de l’État divise les gros contrats en petits contrats pour éviter l’examen minutieux effectué dans le cas des gros contrats (Source : [Association of Certified Fraud Examiners](https://www.fraudconference.com/uploadedFiles/Fraud_Conference/Content/Course-Materials/presentations/23rd/ppt/post-06-Fraud-in-Procurement-Without-Competition.pdf)) |
| Logiciel à code source ouvert | Un logiciel à code source ouvert est un logiciel avec un code source que tout le monde peut consulter, modifier et améliorer. Le « code source » est la partie du logiciel que la plupart des utilisateurs d’ordinateurs ne voient jamais. C’est le code que les programmeurs informatiques peuvent manipuler pour changer le fonctionnement d’un logiciel – un « programme » ou une « application ». Les programmeurs qui ont accès au code source d’un programme informatique peuvent améliorer ce programme en y ajoutant des fonctionnalités ou en corrigeant des parties qui ne fonctionnent pas toujours correctement. (Source : [OpenSource](https://opensource.com/resources/what-open-source)) |
| Logiciel libre | Les logiciels libres sont des logiciels informatiques distribués selon des conditions qui permettent aux utilisateurs d’utiliser le logiciel à n’importe quelle fin, ainsi que d’étudier, de modifier et de distribuer le logiciel proprement dit, ainsi que toute version adaptée. Un logiciel libre est une question de liberté, et non de prix; tous les utilisateurs sont légalement libres de faire ce qu’ils veulent de leurs copies d’un tel logiciel (y compris d’en tirer profit), quel que soit le prix payé pour l’obtenir. Source : [Logiciel libre – Wikipédia](https://en.wikipedia.org/wiki/Free_software) |
| Logiciels libres et à code source ouvert (FOSS) | Les logiciels libres et à code source ouvert (FOSS) sont des logiciels qui sont à la fois des logiciels libres et des logiciels à code source ouvert. Source : [Logiciels libres et logiciels à code source ouvert – Wikipédia](https://en.wikipedia.org/wiki/Free_and_open-source_software) |
| Produits informatiques standard offerts sur le marché | Un produit informatique standard désigne tout produit informatique fabriqué en série et offert sur le marché (prêt à l’emploi), qui est adapté par la suite en fonction des besoins de l’acheteur, plutôt que la mise en service de solutions personnalisées, ou sur mesure. Source : [Wikipédia](https://en.wikipedia.org/wiki/Commercial_off-the-shelf) |
| Référentiel | Dans les systèmes de contrôle de la version, un **référentiel** est une structure de données qui stocke les métadonnées d’un ensemble de fichiers ou d’une structure de répertoire. Source : [Référentiel (contrôle de la version) – Wikipédia](https://en.wikipedia.org/wiki/Repository_(version_control))  Dans un système de contrôle de la version distribuée, chaque projet a un **dépôt central** qui est considéré comme le dépôt officiel et qui est géré par les responsables du projet. Les développeurs clonent ce dépôt pour créer des copies locales identiques de la base de code. Les modifications du code source dans le référentiel central sont synchronisées périodiquement avec le référentiel local.  Le développeur crée une nouvelle branche dans son référentiel local et modifie le code source sur cette **branche**. Une fois le développement effectué, la modification doit être intégrée dans le référentiel central.  Voir aussi : Contrôle de la version distribuée |
| Responsable de produit | Une personne qui comprend votre organisation, ainsi que le problème à résoudre, et qui est en mesure de défendre le produit que nous finirons par concevoir ensemble. Un responsable de produit est chargé d’établir et de mettre en œuvre la vision à long terme du produit, de mettre en œuvre une stratégie et d’orienter son évolution, en fonction des recherches menées auprès des utilisateurs. (Source : [18F](https://18f.gsa.gov/partnership-principles/)) |
| Sous-traitance modulaire | La sous-traitance modulaire est une stratégie d’acquisition qui permet de diviser les grands projets complexes en plusieurs marchés de portée restreinte pour la mise en œuvre de systèmes technologiques successifs et interopérables. (Source : [18F](https://18f.gsa.gov/2019/04/09/why-we-love-modular-contracting/)) |

# Annexe A : Documents d’une page à l’intention de la haute direction

## 

# Document no1 : Pourquoi un logiciel libre?

## 

**Les dirigeants et les cadres supérieurs du secteur public ont la possibilité de mener la transformation numérique de la prestation de services essentiels et des opérations au sein de leur organisation en adoptant des logiciels libres.**

**Pour ce faire, ils doivent :**

* Réorienter la culture organisationnelle de façon à ce qu’elle devienne favorable aux logiciels libres.
* Investir dans le renforcement des capacités du personnel.
* Promouvoir des approches numériques en matière d’approvisionnement, de budgétisation et de passation de marchés qui favorisent l’adoption de logiciels libres.
* Comprendre les différences fondamentales entre les logiciels propriétaires conventionnels et les logiciels libres.
* Tirer parti du potentiel de collaboration interjuridictionnelle pour créer et appuyer des applications logicielles à code source ouvert.

## Éléments clés

* **Les gouvernements et les organismes du secteur public du Canada et du monde entier utilisent déjà les logiciels libres dans une large mesure.** Alors que tous les ordres de gouvernements intègrent des logiciels dans une grande variété de processus et de services pour atteindre leurs objectifs – de la prestation efficace de services publics aux objectifs sociaux et environnementaux – les logiciels libres constituent une excellente option pour garantir la qualité, le contrôle permanent et l’efficacité des ressources.
* **Les logiciels libres sont foncièrement différents des logiciels propriétaires conventionnels** à plusieurs égards : la manière dont ils sont concédés sous licence, détenus et gérés, la manière dont ils sont conçus et mis à niveau, et la manière dont ils sont financés et commercialisés. Un logiciel libre est distribué gratuitement en vertu d’une licence libre. De nombreux projets de logiciels libres sont réalisés par une collectivité mondiale d’utilisateurs et de concepteurs, alors que d’autres ont une base de contributeurs plus restreinte. Le logiciel proprement dit est gratuit, mais il se peut que votre organisation doive investir dans les capacités du personnel interne pour assurer l’intégration et la mise à niveau du logiciel, ou encore passer un contrat avec un fournisseur pour ajouter des fonctionnalités spécifiques ou faire une intégration personnalisée en fonction de votre environnement numérique existant.
* **Les logiciels libres ne doivent pas nécessairement être conçus à partir de rien.** Des centaines de projets complets de logiciels libres et encore plus de petits modules de composantes de logiciels libres offrent autant d’options et de fonctionnalités que les logiciels propriétaires conventionnels. Ces projets et modules servent de base à la création de nouveaux logiciels libres.
* Les **logiciels libres peuvent présenter une variété d’avantages uniques pour le secteur public –** notamment la personnalisation, l’interopérabilité et la transparence. Bien qu’il existe des obstacles à l’adoption des logiciels libres dans le secteur public – notamment l’absence de capacités techniques internes et l’application de pratiques d’approvisionnement rigides – il existe heureusement une variété de stratégies et de pratiques exemplaires pouvant vous aider à surmonter ces obstacles. **Les gouvernements et les organismes du secteur public du Canada et du monde entier utilisent déjà les logiciels libres de manière significative.** À l’heure où les gouvernements, à tous les niveaux, intègrent des logiciels dans une grande variété de processus et de services pour atteindre leurs objectifs – de la prestation efficace de services publics aux objectifs sociaux et environnementaux – les logiciels libres constituent une excellente option pour garantir la qualité, le contrôle permanent et l’efficacité des ressources.

# Document no 2 : Création d’un écosystème du logiciel libre

## 

**Ce module vous permettra de vous familiariser avec :**

* Les principaux facteurs favorables et les obstacles à l’adoption des logiciels libres, liés à la dotation en personnel et à la structure organisationnelle.
* La grande variété de fonctions, de profils professionnels et d’ensembles de compétences nécessaires pour le développement, la mise en œuvre et la mise à niveau de projets de logiciels libres au sein d’une organisation gouvernementale.

Les pratiques exemplaires pour constituer et diriger une équipe technique de taille adéquate.

## Éléments clés

* **Le paysage politique et législatif appuie de plus en plus l’adoption des logiciels libres.** Cependant, les structures, les pratiques et les mentalités actuelles au sein de nombreux organismes gouvernementaux et du secteur public en général peuvent donner lieu à des obstacles lorsqu’il s’agit d’adopter des logiciels libres.
* **La dotation en personnel technique est l’un des principaux obstacles perçus en ce qui concerne l’adoption de logiciels libres.** Un manque de capacités internes et de culture de collaboration peut conduire à une dépendance excessive à l’égard des fournisseurs : Peu d’organisations gouvernementales disposent d’un personnel à temps plein (aux niveaux de la direction et de la mise en œuvre) ayant les compétences et les connaissances nécessaires pour acquérir, développer et/ou mettre à niveau des logiciels libres. Par conséquent, les gouvernements s’en remettent généralement à des fournisseurs.[[89]](#footnote-89) Cela peut entraîner l’enfermement du fournisseur, l’explosion des coûts et des frais cachés, ainsi que la perte de contrôle (sur les fonctionnalités, les mises à jour et la conformité).[[90]](#footnote-90)
* Les **dirigeants d’organisations de toutes tailles peuvent relever ces défis en réalisant des investissements stratégiques, en réduisant les risques réels et perçus et en encourageant un changement de culture organisationnelle**. Cela comprend :
  + Apprendre comment fonctionnent l’acquisition, le développement et la mise à niveau des logiciels libres et appuyer les groupes de travail et les champions internes.
  + Favoriser une culture organisationnelle qui encourage la transparence et l’engagement continu des utilisateurs.
* Mobiliser une variété de partenaires, d’organisations homologues et de prestataires de services pour compléter l’équipe interne.

# Document no3 : Approvisionnement, budgétisation et passation de marchés.

## 

Ce module vous permettra de vous familiariser avec :

* Les principales différences entre les approches en matière de licence et de développement et la raison pour laquelle les systèmes gouvernementaux ont tendance à privilégier les logiciels propriétaires et le développement en cascade.
* Les différents modèles de revenus des logiciels et de la manière dont ils affectent les structures de coûts des gouvernements.
* Les nouvelles approches en matière d’approvisionnement, de passation de marchés, d’octroi de licences et de dotation en personnel.

## Éléments clés

* **Les structures et les normes en matière d’approvisionnement, de budgétisation et de passation de marchés sont conçues pour des actifs antérieurs à l’ère numérique – ce qui complique l’obtention de logiciels libres.** Cependant, à mesure que les organisations gouvernementales adoptent des approches axées sur le numérique, les logiciels libres et la méthode agile deviennent la voie la plus facile.
* **Les logiciels libres nécessitent une approche différente de celle des logiciels propriétaires en matière de budgétisation.** Alors que les logiciels propriétaires sont tarifés selon l’un des nombreux modèles de revenus standard, les logiciels libres sont distribués gratuitement, mais nécessitent des investissements en personnel et/ou en fournisseurs de services pour déployer et maintenir les logiciels.
* Les **logiciels libres s’harmonisent bien avec les méthodes « agiles »**, qui se révèlent également être les pratiques exemplaires en matière de développement de logiciels. Lors de l’acquisition de services logiciels personnalisés, le gouvernement devrait utiliser des approches contractuelles et budgétaires modulaires, qui permettent un développement agile.
* **La passation de marchés est un processus créatif et exploratoire.** En utilisant correctement les marchés publics, vous pouvez non seulement trouver la meilleure solution au moindre coût, mais aussi générer des retombées positives.
* L**’achat groupé est une stratégie efficace** permettant à un groupe d’organisations gouvernementales de partager le coût de l’obtention et de la mise à niveau de logiciels libres répondant à un besoin commun.

# Document no 4 : Sécurité et mise à niveau des logiciels libres

## 

Ce module vous permettra de vous familiariser avec :

* Les considérations liées à la sécurité et à la mise à niveau associées aux logiciels propriétaires conventionnels et aux logiciels libres.
* Les principaux défis auxquels les organisations sont confrontées dans la mise à niveau et la sécurisation de leurs solutions logicielles libres.
* Les stratégies pour assurer une gestion et une mise à niveau réussies des logiciels libres.

## Éléments clés

* **Les logiciels à risque zéro n’existent pas : Tous les logiciels, qu’il s’agisse de logiciels propriétaires ou de logiciels libres, comportent des sous-modules et présentent des risques devant être gérés.** En fait, la plupart des entreprises de logiciels propriétaires conventionnels utilisent le code source ouvert dans leurs produits et participent à la mise à niveau des logiciels libres.
* **Les logiciels libres ne sont pas intrinsèquement moins sûrs que les logiciels propriétaires conventionnels**. Comme tout logiciel, les logiciels libres comportent certains risques inévitables, mais cela ne doit pas vous décourager d’envisager leur utilisation au sein de votre organisation. La meilleure solution est de procéder à une évaluation approfondie des risques et d’élaborer des plans appropriés pour les atténuer.
* **Plus il y a d’intervenants qui utilisent un logiciel libre, plus il y a de chances que les problèmes de mise à niveau critiques et les failles de sécurité soient résolus.** Les logiciels libres appuyés par une grande collectivité active d’utilisateurs et de développeurs sont plus susceptibles de recevoir régulièrement des mises à jour et des rustines de sécurité. Toutefois, si vous utilisez un logiciel libre qui n’est pas largement appuyé par une collectivité d’utilisateurs, vous devrez compenser en vous assurant que vous disposez des capacités internes appropriées (ou que vous êtes en mesure de faire appel à une assistance externe) pour surveiller les vulnérabilités de sécurité.
* **Ne créez pas de « bifurcation » (fork) si vous n’êtes pas en mesure d’assurer la mise à niveau; plus votre organisation crée de codes personnalisés, plus vous devrez effectuer de mises à niveau**. Si vous créez une instance locale indépendante (en bifurquant le code), les mises à jour et les rustines de sécurité deviendront votre responsabilité – et deviendront de plus en plus difficiles, car votre version s’éloignera de la version initiale au fil du temps.

# Document no5 : Collaboration en ce qui concerne les logiciels libres

## 

Ce module vous permettra de vous familiariser avec :

* Les questions interjuridictionnelles soulevées dans les modules précédents et les considérations clés lors de la formation d’équipes multijuridictionnelles autour d’un logiciel particulier, y compris le partage des ressources entre les différentes administrations ou ordres de gouvernements.
* La participation à une collectivité mondiale de gestion des codes bases.
* L’utilité de diffuser votre logiciel sous une licence libre.

## Éléments clés

* **La collaboration en ce qui concerne les logiciels libres présente des avantages pour tous les intervenants.** Le logiciel que vous développez, ou les adaptations que vous faites, seront également utiles à d’autres administrations. Les logiciels libres sont plus performants et plus sûrs à mesure que la collectivité qui les utilise se développe. Il est donc dans l’intérêt de chaque utilisateur de contribuer, et d’ajouter d’autres utilisateurs contributeurs.
* **Les organisations gouvernementales n’ont pas besoin de réinventer la roue.** Lorsque le code source est publié sous une licence libre, une administration peut procéder à de petites adaptations du logiciel afin de veiller à ce qu’il soit le mieux adapté aux cas d’utilisation, aux réglementations et aux normes locales. Lorsqu’un ensemble de fonctionnalités de base est partagé, le temps, les efforts et les ressources de chaque administration peuvent être consacrés à de petites adaptations en fonction du contexte local (plutôt que de créer un ensemble complet de fonctionnalités de base à partir de rien).
* **Les gouvernements de même niveau sont des collaborateurs naturels lorsqu’il s’agit de concevoir, d’obtenir et de mettre à niveau des logiciels libres.** Ils sont chargés de fournir un ensemble similaire de services et ont un ensemble similaire de besoins (internes, administratifs et externes, axés sur les citoyens). Ils sont donc des collaborateurs naturels lorsqu’il s’agit de concevoir, d’obtenir et de mettre à niveau des logiciels libres.
* **Les administrations collaborent déjà dans le cadre de différents réseaux et groupes de pairs bien établis qui peuvent être activés en tant que collectivités de régie de logiciels libres.** Les plateformes de contrôle de la version distribuée (comme GitHub) permettent aux utilisateurs de contribuer ou de suggérer des modifications de manière structurée et collaborative.

# 

# Annexe B : Mise en œuvre du cours

Le matériel de formation peut être distribué dans sa forme actuelle, en tant que document statique. En cas de distribution sous forme de page web ou de cours en ligne, veuillez tenir compte de ce qui suit :

* Nous vous recommandons d’utiliser la section [Aperçu du cours](#_heading=h.k7bllgkqq56i) comme page d’accueil. Les pages [Avant-propos](#_heading=h.fcmx27u5o5w9), le [Glossaire](#_heading=h.zu0gcz) et les [Documents d’une page à l’intention de la haute direction](#_heading=h.hzug8chicr9f) (1 ou 2 pages) peuvent également être liés à cette page.
* Les citations ont été fournies sous forme de notes de bas de page afin de maintenir leur lien avec le matériel cité si le contenu est réorganisé à l’avenir.
* Les modules peuvent être publiés sous forme de page unique (HTML) ou divisés en diapositives/sous-sections au niveau de l’en-tête 1 ou de l’en-tête 2.
* L’annexe B (c’est-à-dire cette page) doit être exclue de la version de ce cours accessible au public.

1. Secrétariat du Conseil du Trésor du Canada, [*Livre blanc Ouvert en premier : Marchés ouverts*](https://www.canada.ca/fr/gouvernement/systeme/gouvernement-numerique/innovations-gouvernementales-numeriques/logiciels-libres/livre-blanc-ouvert-premier/livre-blanc-ouvert-premier-marches.html), le 28 juillet 2020. [↑](#footnote-ref-1)
2. [*Public Digital, Open Source in government: creating conditions for success*](https://public.digital/research), juin 2021. [↑](#footnote-ref-2)
3. Matthew Claudel et Bianca Wylie, [*Technology Procurement : Shaping Future Public Value, Community Solutions Network Brief*](https://opennorth.ca/fr/) (Nord Ouvert, 2021). [↑](#footnote-ref-3)
4. Exigences et directives en matière de logiciels libres (ébauche), HTML (2018; repr., gouvernement du Canada, 14 juin 2022) <https://github.com/canada-ca/open-source-logiciel-libre>. [↑](#footnote-ref-4)
5. Gouvernement de l’Ontario, [Normes des services numériques](https://www.ontario.ca/fr/page/norme-des-services-numeriques), consultées le 22 novembre 2022. [↑](#footnote-ref-5)
6. Gouvernement du Québec. Secrétariat du Conseil du Trésor, [*Logiciels libres*](file:///C:\Users\ngolden\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\MMNTKCT7\Logiciels%20libres), 2009. [↑](#footnote-ref-6)
7. Gouvernement du Québec. [*Forge gouvernementale – Licence libre du Québec (LiLiQ)*](https://www.forge.gouv.qc.ca/), consultation le 10 novembre 2022. [↑](#footnote-ref-7)
8. Apolitical, [*How Opening up Government Source Code Improves Everything*](https://apolitical.co/solution-articles/en/how-opening-up-government-source-code-improves-everything), le 7 décembre 2021 [↑](#footnote-ref-8)
9. Commissariat à la protection de la vie privée du Canada, [*Aperçu des lois sur la protection des renseignements personnels au Canada*](https://www.priv.gc.ca/fr/sujets-lies-a-la-protection-de-la-vie-privee/lois-sur-la-protection-des-renseignements-personnels-au-canada/02_05_d_15/), le 15 mai 2014. [↑](#footnote-ref-9)
10. [*Teaching Public Service in the Digital Age: Les compétences de l’ère numérique*](https://www.teachingpublicservice.digital/fr/competencies) [↑](#footnote-ref-10)
11. Paulina Grzegorzewska, [*What can governmental OSPOs learn from the private sector ones?*](https://joinup.ec.europa.eu/collection/open-source-observatory-osor/news/lessons-private-sector-governmental-ospos), Joinup, le 9 février 202.1 [↑](#footnote-ref-11)
12. Idem [↑](#footnote-ref-12)
13. New America, [*Section Four : Managing the Details – Building and Reusing Open Source Tools for Government*](https://www.newamerica.org/digital-impact-governance-initiative/reports/building-and-reusing-open-source-tools-government/section-four-managing-the-details/), consultation le 4 juillet 2022. [↑](#footnote-ref-13)
14. [Public Digital, *Open Source in Government*](https://www.zotero.org/google-docs/?ZHVaPD)*.* [↑](#footnote-ref-14)
15. [Public Digital, *Open Source in Government*](https://www.zotero.org/google-docs/?ZHVaPD)*.* [↑](#footnote-ref-15)
16. Secrétariat du Conseil du Trésor du Canada, [*Livre blanc Ouvert en premier*](https://www.canada.ca/fr/gouvernement/systeme/gouvernement-numerique/innovations-gouvernementales-numeriques/logiciels-libres/livre-blanc-ouvert-premier.html), le 28 juillet 2020. [↑](#footnote-ref-16)
17. Aaron Burk and Rebecca Refoy, [*A Day in the Life of an 18F Product Owner*](https://18f.gsa.gov/2017/09/18/a-day-in-the-life-of-an-18f-product-owner/), *18F* (blog), le 18 septembre 2017. [↑](#footnote-ref-17)
18. [*18F Partnership Principles*](https://18f.gsa.gov/partnership-principles/), consultation le 18 octobre 2022. [↑](#footnote-ref-18)
19. New America,  [*Section Two – Building Open Source Software – Building and Reusing Open Source Tools for Government*](http://newamerica.org/digital-impact-governance-initiative/reports/building-and-reusing-open-source-tools-government/), consultation le 4 juillet 2022, [↑](#footnote-ref-19)
20. Secrétariat du Conseil du Trésor du Canada, [*6. Intégrer l’accessibilité dès le départ*](https://canada-ca.github.io/digital-playbook-guide-numerique/views-vues/standards-normes/fr/6-build-in-accessibility-from-start.html) *– Outils numériques du gouvernement du Canada (ébauche)*, consultation le 18 octobre 2022. [↑](#footnote-ref-20)
21. Shira Ovide, [*Government Tech Moves Too Slooowly*](https://www.nytimes.com/2021/07/07/technology/jedi-government-tech.html), The New York Times, le 7 juillet 2021. [↑](#footnote-ref-21)
22. Heather J. Meeker, [*Chapter 13. Enforcement of Open Source Licenses*, The Open Source Alternative : Understanding Risks and Leveraging Opportunities](https://www.oreilly.com/library/view/the-open-source/9780470194959/ch13.html)(Hoboken, NJ : Wiley & Sons, 2008). [↑](#footnote-ref-22)
23. AltexSoft, [*Revenue Model Types and Examples*,](https://www.altexsoft.com/blog/revenue-model-types/) le 12 juin 2020. [↑](#footnote-ref-23)
24. Waldo Jacquith, *[The Disconnect between Software Development and Government Contracting](https://waldo.jaquith.org/blog/2021/05/disconnect-development-and-government-contracting/),* le 8 mai 2021. [↑](#footnote-ref-24)
25. Waldo Jacquith, [*The Disconnect between Software Development and Government Contracting*](https://waldo.jaquith.org/blog/2021/05/disconnect-development-and-government-contracting/). [↑](#footnote-ref-25)
26. Waldo Jacquith, [*The Disconnect between Software Development and Government Contracting*](https://waldo.jaquith.org/blog/2021/05/disconnect-development-and-government-contracting/)*.* [↑](#footnote-ref-26)
27. Matthew Claudel et Bianca Wylie, [*Technology Procurement : Shaping Future Public Value*](https://opennorth.ca/publications/technology-procurement-shaping-future-public-value/), Community Solutions Network Research Brief (Nord Ouvert, 2021). [↑](#footnote-ref-27)
28. Matthew Claudel et Bianca Wylie, [*Technology Procurement : Shaping Future Public Value*](https://opennorth.ca/publications/technology-procurement-shaping-future-public-value/), Community Solutions Network Research Brief (Nord Ouvert, 2021). [↑](#footnote-ref-28)
29. Josh Lowe, [*Canadian Budget Leads with Digital Tax and IT Investments*](https://www.globalgovernmentforum.com/canadian-budget-leads-with-digital-tax-and-it-investments/), Global Government Forum, le 20 avril 2021 [↑](#footnote-ref-29)
30. Banque de développement du Canada, [*Perspectives du secteur des technologies : Comment les changements dans l’économie affectent le secteur technologique canadien*](https://www.bdc.ca/globalassets/digizuite/28336-st-outlookmfg-f2010-2.pdf)*,* janvier 2021. [↑](#footnote-ref-30)
31. Luke DeCoste, [*Outdated Procurement Rules Hindering Digital Government*](https://policyoptions.irpp.org/magazines/february-2019/outdated-procurement-rules-hindering-digital-government/) Policy Options, le 12 février 2019. [↑](#footnote-ref-31)
32. Kaye Sklar, [*A Procurement Path to Equity : Strategies for Government and the Business Ecosystem*](https://www.open-contracting.org/wp-content/uploads/2020/11/OCP-AspenCUI-2020-Pathway-to-Equity.pdf) (Open Contracting Partnership et Aspen Institute Center for Urban Innovation, 2020). [↑](#footnote-ref-32)
33. Jerrod Larson, Jen Hocko et Richard Bye, [*User-Centered Procurement : Evaluating the Usability of Off-the Shelf Software User Experience Magazine*](https://uxpamagazine.org/user_centered_procurement/), UX Development, mars 2010. [↑](#footnote-ref-33)
34. LaBrie, Suzanne et Ferron, Pierre-Antoine, [*Qu’est-ce que l’approvisionnement ouvert?*](https://lms.opennorth.ca/catalog/info/id:145), Réseau de solutions pour les communautés, Nord Ouvert. [↑](#footnote-ref-34)
35. Bien qu’il ne couvre qu’une poignée de grandes villes, [Open Source Cities](https://github.com/opensourcecities) vous donne une idée de la variété des intervenants et des organisations impliqués dans les écosystèmes locaux de logiciels libres. Une fois que vous aurez commencé à chercher, vous serez peut-être surpris d’apprendre combien d’activités de logiciels libres se déroulent déjà dans votre région. [↑](#footnote-ref-35)
36. Sascha Haselmayer, [*Unit 3 : Choosing Your Procurement Path, Citymart Procurement Institute* (blog)](https://medium.com/citymartinsights/unit-3-choosing-your-procurement-path-8e1b711d0268), le 26 mars 2020. [↑](#footnote-ref-36)
37. Ben Winter, [*4 Key Steps to Performing an Effective Spend Analysis, Fairmarkit (blog)*](https://www.fairmarkit.com/blog/4-key-steps-to-performing-an-effective-spend-analysis), le 25 août 2021; Doug Greer et Jen Scarlato, [*Best Practices for TCO Costing*](https://cdn2.hubspot.net/hubfs/2652075/Downloadable_Files/regoUniversity%202018/Functional/TCO%20Best%20Practices%20for%20TCO Costing_SVMGT01.pdf). [↑](#footnote-ref-37)
38. Secrétariat du Conseil du Trésor du Canada, [*Guide pour l’utilisation de logiciels libres – Évaluer les options de soutien*](https://www.canada.ca/fr/gouvernement/systeme/gouvernement-numerique/innovations-gouvernementales-numeriques/logiciels-libres/guide-pour-lutilisation-de-logiciels-libres.html), le 28 juillet 2020. [↑](#footnote-ref-38)
39. Laura Gerhardt et Mark Headd, [*Why We Love Modular Contracting*](https://18f.gsa.gov/2019/04/09/why-we-love-modular-contracting/), *18F* (blog), le 9 avril 2019. [↑](#footnote-ref-39)
40. Mark Hopson, [*Improving Government Outcomes through an Agile Contract Format*](https://18f.gsa.gov/2017/11/30/improving-government-outcomes-through-an-agile-contract-format/), *18F* (blog), le 30 novembre 2017. [↑](#footnote-ref-40)
41. Association of Certified Fraud Examiners, [*Contract and Procurement Fraud: Fraud in Procurement without Competition*](https://www.acfe.com/training-events-and-products/all-events/seminar-topics/contract-and-procurement-fraud). [↑](#footnote-ref-41)
42. US General Services Administration, [*2nd Generation IT Blanket Purchase Agreements*](https://www.gsa.gov/technology/technology-purchasing-programs/mas-information-technology/buy-from-mas-information-technology/2nd-generation-it-blanket-purchase-agreements)*.* [↑](#footnote-ref-42)
43. Rémy Bernard, [*Les problèmes du financement par projet pour les organisations TI*](https://sara-sabr.github.io/ITStrategy/2021/05/18/problemes-financement-par-projet.html) (blogue), le 18 mai 2021. [↑](#footnote-ref-43)
44. Rémy Bernard, [*Les problèmes du financement par projet pour les organisations de la TI*](https://sara-sabr.github.io/ITStrategy/2021/05/18/problemes-financement-par-projet.html) [↑](#footnote-ref-44)
45. Rémy Bernard, [*Le financement d’activités reliées au logiciel*](https://sara-sabr.github.io/ITStrategy/2022/07/21/financement-activites-logiciel.html)*,* Équipe de la stratégie des technologies de l’information (blogue)*,* 21 juillet 2022. [↑](#footnote-ref-45)
46. Zachary Spicer, [*Digital Dilemmas : Technology, Governance, and Canadian Municipalities*](https://imfg.munkschool.utoronto.ca/research/doc/?doc_id=597), IMFG Forum Papers (Institute on Municipal Finance & Governance, 2022), 7-8. [↑](#footnote-ref-46)
47. Ben Balter, [*Eight Reasons Why Government Contractors Should Embrace Open Source Software*](https://ben.balter.com/2014/10/08/why-government-contractors-should-embrace-open-source/), *Ben Balter* (blog), le 8 octobre 2014. [↑](#footnote-ref-47)
48. Maija Jokiniemi, [*La Commission européenne a reconnu l’outil de passation des marchés publics de Forum Virium Helsinki comme une innovation de premier plan*](https://forumvirium.fi/en/the-european-commission-acknowledged-forum-virium-helsinkis-public-procurement-tool-as-a-leading-innovation/), Forum Virium Helsinki, le 19 août 2022. [↑](#footnote-ref-48)
49. Chris Romeo, [*A Security Practitioner’s Guide to Software Obsolescence*](https://techbeacon.com/security/security-practitioners-guide-software-obsolescence), TechBeacon, consultation le 29 septembre 2022. [↑](#footnote-ref-49)
50. Keith Turpin, [*Secure Coding Practices – Quick Reference Guide*](https://owasp.org/www-pdf-archive/OWASP_SCP_Quick_Reference_Guide_v2.pdf) (OWASP Fondation, 2010), 4. [↑](#footnote-ref-50)
51. Nadia Eghbal, [*Road and Bridges : The Unseen Labor Behind Our Digital Infrastructure*](https://www.fordfoundation.org/work/learning/research-reports/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure/) (Ford Foundation, 2016), 77. [↑](#footnote-ref-51)
52. Liam Tung, [*Open Source Security Fears Are Fading Away*](https://www.zdnet.com/article/is-open-source-as-proprietary-software-these-tech-chiefs-think-it-is/), ZDNET, le 3 mars 2022 [↑](#footnote-ref-52)
53. Rafael Laguna, [*4 Myths About Open Source We Should Put to Rest*](https://www.wired.com/insights/2013/03/4-myths-about-open-source-to-put-to-rest/) | WIRED, mars 2013. [↑](#footnote-ref-53)
54. Centre canadien pour la cybersécurité, [*Évaluation des cybermenaces nationales 2020*](https://cyber.gc.ca/sites/default/files/cyber/publications/ncta-2020-f-web.pdf) (Gouvernement du Canada, 2020). [↑](#footnote-ref-54)
55. Commissariat à la protection de la vie privée du Canada, [*Conversation avec le commissaire à la protection de la vie privée du Canada à l’occasion de la Journée de la protection des données*](https://www.priv.gc.ca/fr/nouvelles-du-commissariat/allocutions/2020/sp-d_20200128/), le 5 février 2020. [↑](#footnote-ref-55)
56. Centre canadien pour la cybersécurité, [*Considérations liées à la sécurité dans le cadre de l’utilisation de logiciels libres*](https://www.cyber.gc.ca/fr/orientation/considerations-liees-la-securite-dans-le-cadre-de-lutilisation-de-logiciels-libres) (ITSAP.10.059), le 9 juillet 2020. [↑](#footnote-ref-56)
57. Centre canadien pour la cybersécurité, *[Considérations liées à la sécurité dans le cadre de l’utilisation de logiciels libres](https://www.cyber.gc.ca/fr/orientation/considerations-liees-la-securite-dans-le-cadre-de-lutilisation-de-logiciels-libres).* [↑](#footnote-ref-57)
58. Centre canadien pour la cybersécurité, [*Considérations liées à la sécurité dans le cadre de l’utilisation de logiciels libres*](https://www.cyber.gc.ca/fr/orientation/considerations-liees-la-securite-dans-le-cadre-de-lutilisation-de-logiciels-libres) (ITSAP.10.059). [↑](#footnote-ref-58)
59. Centre canadien pour la cybersécurité, [*Considérations liées à la sécurité dans le cadre de l’utilisation de logiciels libres*](https://www.cyber.gc.ca/fr/orientation/considerations-liees-la-securite-dans-le-cadre-de-lutilisation-de-logiciels-libres) (ITSAP.10.059). [↑](#footnote-ref-59)
60. Eric Brewer et al., [*Know, Prevent, Fix : A Framework for Shifting the Discussion around Vulnerabilities in Open Source*](https://opensource.googleblog.com/2021/02/know-prevent-fix-framework-for-shifting-discussion-around-vulnerabilities-in-open-source.html), *Google Open Source Blog* (blog), le 3 février 2021. [↑](#footnote-ref-60)
61. Jim Boehm et al, [*The Risk-Based Approach to Cybersecurity*](https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/the-risk-based-approach-to-cybersecurity), McKinsey & Company, le 8 octobre 2019. [↑](#footnote-ref-61)
62. Zeljka Zorz, [*The Percentage of Open Source Code in Proprietary Apps Is Rising*](https://www.helpnetsecurity.com/2018/05/22/open-source-code-security-risk/), *Help Net Security* (blog), le 22 mai 2018. [↑](#footnote-ref-62)
63. Oskar Barcz, [*What is Dependency Hell and How to Avoid It?*](https://www.boldare.com/blog/software-dependency-hell-what-is-it-and-how-to-avoid-it), le 26 novembre 2021. [↑](#footnote-ref-63)
64. L’objectif du coussinet gauche est très simple : il « remplit » le côté gauche d’une chaîne (valeurs entières ou caractères) en ajoutant des zéros ou des espaces pour atteindre la longueur de chaîne souhaitée. [↑](#footnote-ref-64)
65. Keith Collins, *How One Programmer Broke the Internet by Deleting a Tiny Piece of Code*, Quartz, le 27 mars 2016. [↑](#footnote-ref-65)
66. David Haney, [*NPM & Left-Pad : Have We Forgotten How To Program?*](https://www.davidhaney.io/npm-left-pad-have-we-forgotten-how-to-program/), consultation le 29 août 2022. [↑](#footnote-ref-66)
67. United States Cybersecurity & Infrastructure Security Agency, [*Software Bill of Materials*](https://www.cisa.gov/sbom), consultation le 13 novembre 2022. [↑](#footnote-ref-67)
68. New America, [*Section Two : Building Open Source Software, Building and Reusing Open Source Tools for Government*](http://newamerica.org/digital-impact-governance-initiative/reports/building-and-reusing-open-source-tools-government/)*,* consultation le 4 juillet 2022. [↑](#footnote-ref-68)
69. Secrétariat du Conseil du Trésor du Canada, [*Guide pour la publication du code source libre*](https://www.canada.ca/fr/gouvernement/systeme/gouvernement-numerique/innovations-gouvernementales-numeriques/logiciels-libres/guide-pour-la-publication-du-code-source-libre.html). [↑](#footnote-ref-69)
70. Ann Cavoukian et Mark Dixon, [*Privacy and Security by Design : An Enterprise Architecture Approach*](https://www.ipc.on.ca/wp-content/uploads/Resources/pbd-privacy-and-security-by-design-oracle.pdf) (Commissaire à l’information et à la protection de la vie privée de l’Ontario, 2013). [↑](#footnote-ref-70)
71. Nicole Olsen, [*Open Source Projects and the GDPR, Privacy Policies*](https://www.privacypolicies.com/blog/gdpr-open-source-projects/), consultation le 2 septembre 2022. [↑](#footnote-ref-71)
72. Secrétariat du Conseil du Trésor du Canada, [*Guide pour la publication du code source libre*](https://www.canada.ca/en/government/system/digital-government/digital-government-innovations/open-source-software/guide-for-using-open-source-software.html), le 28 juillet 2020,*[Guide pour la publication du code source libre](https://www.canada.ca/en/government/system/digital-government/digital-government-innovations/open-source-software/guide-for-using-open-source-software.html).* [↑](#footnote-ref-72)
73. Open Source Security Foundation et The Linux Foundation, [*Home, OpenSSF Scorecard*](https://securityscorecards.dev/), consultation le 14 novembre 2022. [↑](#footnote-ref-73)
74. Secrétariat du Conseil du Trésor du Canada, [*Guide pour l’utilisation de logiciels libres*](https://www.canada.ca/fr/gouvernement/systeme/gouvernement-numerique/innovations-gouvernementales-numeriques/logiciels-libres/guide-pour-lutilisation-de-logiciels-libres.html)*.* [↑](#footnote-ref-74)
75. Justin C. Colannino, [*Free and Open Source Software in Municipal Procurement : The Challenges and Benefits of Cooperation*](https://ir.lawnet.fordham.edu/ulj/vol39/iss4/2), *Fordham Urban Law Journal* 39 (février 2016) : 903-29. [↑](#footnote-ref-75)
76. Secrétariat du Conseil du Trésor du Canada, *[Guide pour l’utilisation de logiciels libres](https://www.canada.ca/fr/gouvernement/systeme/gouvernement-numerique/innovations-gouvernementales-numeriques/logiciels-libres/guide-pour-lutilisation-de-logiciels-libres.html)*, le 28 juillet 2020. [↑](#footnote-ref-76)
77. [*Guide pour la publication du code source libre*](https://www.canada.ca/fr/gouvernement/systeme/gouvernement-numerique/innovations-gouvernementales-numeriques/logiciels-libres/guide-pour-la-publication-du-code-source-libre.html)*.* [↑](#footnote-ref-77)
78. Ben Balter, [*Everything a Government Attorneu needs to Know about Open Source Software Licensing*](https://ben.balter.com/2014/10/08/open-source-licensing-for-government-attorneys/), *Ben Balter* (blog), le 8 octobre 2014;GitHub, [*Choose an Open Source License*](https://choosealicense.com/), consultation le 6 septembre 2022; GitHub, [*The Legal Side of Open Source*](https://opensource.guide/legal/), Open Source Guides, 17 octobre 2022. [↑](#footnote-ref-78)
79. Secrétariat du Conseil du Trésor du Canada, [*Guide pour l’utilisation de logiciels libres*](https://www.canada.ca/fr/gouvernement/systeme/gouvernement-numerique/innovations-gouvernementales-numeriques/logiciels-libres/guide-pour-lutilisation-de-logiciels-libres.html), le 28 juillet 2020. [↑](#footnote-ref-79)
80. Margaret Lin, [*Learning from Failure: When Sharing Software Doesn’t Work*](https://beeckcenter.georgetown.edu/learning-from-failure-when-sharing-software-doesnt-work/), Beeck Centre (blog), le 11 avril 2022. [↑](#footnote-ref-80)
81. Waldo Jaquith et Robin Carnahan, [*Sharing Government Software: How Agencies Are Cooperatively Building Mission-Critical Software*](https://beeckcenter.georgetown.edu/wp-content/uploads/2021/04/Sharing-Government-Software.pdf) (Beeck Center for Social Impact + Innovation, Georgetown University, 2021). [↑](#footnote-ref-81)
82. Margaret Lin, [*Learning from Failure: When Sharing Software Doesn’t Work*](https://beeckcenter.georgetown.edu/learning-from-failure-when-sharing-software-doesnt-work/), Beeck Centre (blog), le 11 avril 2022. [↑](#footnote-ref-82)
83. [*Foundation for Public Code, Governance Game*](https://about.publiccode.net/activities/supporting-codebase-governance/game/), consultation le 3 novembre 2022. [↑](#footnote-ref-83)
84. Miller, Ben. [*How Government is Reforming IT Procurement and What it Means for Vendors*](https://www.govtech.com/biz/how-government-is-reforming-it-procurement-and-what-it-means-for-vendors.html), GovTech*,* le 6 avril 2017. [↑](#footnote-ref-84)
85. Centre for Global Development, [*Better Together: Exploring the Role of Pooled Procurement in Improving Access to Medicines amid COVID-19*](https://www.cgdev.org/blog/better-together-exploring-role-pooled-procurement-improving-access-medicines-amid-covid-19). [↑](#footnote-ref-85)
86. CoProcure, [*What is Cooperative Procurement?*](https://www.coprocure.us/blog/cooperative-contracts-joint-solicitation-vs-piggybacking/) [↑](#footnote-ref-86)
87. National Association of State Procurement Officials (NASPO), [*Strength in Numbers: An Introduction to Cooperative Procurements*](https://naspovaluepoint.org/wp-content/uploads/2020/08/Cooperative_Purchasing0410update.pdf). [↑](#footnote-ref-87)
88. Foundation for Public Code, [*Introduction, Standard for Public Code*](https://standard.publiccode.net/introduction.html), le 7 septembre 2022. [↑](#footnote-ref-88)
89. [*Public Digital, Open Source in government : creating conditions for success*](https://public.digital/research), juin 2021. [↑](#footnote-ref-89)
90. Matthew Claudel et Bianca Wylie, [*Technology Procurement : Shaping Future Public Value, Community Solutions Network Brief*](https://opennorth.ca/fr/) (Nord Ouvert, 2021). [↑](#footnote-ref-90)