

Ingénieurs/ingénieures et concepteurs/conceptrice en logiciel

CPN 2173

Introduction

Les ingénieurs e/ingénieures et les concepteurs/conceptrices en logiciel recherchent, conçoivent, évaluent, intègrent et maintiennent les applications de logiciels, les milieux techniques, les systèmes d'exploitation, les logiciels enfouis, les entrepôts de données et les logiciels de télécommunication. Ils occupent des postes dans des sociétés d'experts-conseils en technologie de l'information, dans des sociétés de recherche et de développement en technologie de l'information, dans des unités de technologie de l'information à travers les secteurs privé et public et ils peuvent aussi travailler à leur propre compte.

Compétences essentielles les plus importantes :

- Compétences numériques
- Capacité de raisonnement : Résolution de problèmes
- Communication orale

Table des matières

- Lecture
- Utilisation de documents
- Rédaction
- Calcul
- Communication orale
- Capacité de raisonnement
 - Résolution de problèmes
 - Prise de décisions
 - Pensée critique
 - Planification et organisation de leur travail
 - Utilisation particulière de la mémoire
 - Recherche de renseignements
- Travail d'équipe
- Compétences numériques
- Formation Continue

Notes

A. Lecture

Lecture

Tâches	Niveau de complexité	Exemples
À		Les ingénieurs /ingénieures et les concepteurs/conceptrices en logiciel accomplissent une partie ou l'ensemble des tâches suivantes :
Typiques	2 à 5	<ul style="list-style-type: none"> • Lire des courriels de la part de collaborateurs, de collègues, de superviseurs et de testeurs concernant des demandes pour résoudre des problèmes pouvant inclure des informations techniques et des solutions possibles. (2)
Plus complexe	5	<ul style="list-style-type: none"> • Lire des fils de discussion et des messages sur le système de billetterie qui aide à planifier les tâches. (2) • Lire des forums pour obtenir des renseignements tels que des questions du réalisateur ou des problèmes spécifiques au langage. (3) • Lire des renseignements dans des sites Web pour rechercher de nouvelles technologies ou des renseignements techniques. (3) • Lire des rapports de clients portant sur les questions et les bogues dans les fichiers électroniques et les codes source pour comprendre les problèmes et comment ils se sont développés. (3) • Lire les accords contractuels et les accords concernant le niveau de service présentant les frais prévus, l'historique et les responsabilités. (3) • Lire les rapports de test de l'assurance de la qualité qui indiquent les résultats des tests de logiciels pour garantir que le produit corresponde aux attentes de la clientèle. (3) • Lire les propositions et les documents de spécifications de conception afin de comprendre les spécifications du projet. (3) • Lire la documentation officielle telle que les lignes directrices et les normes de langage, de conception et de système d'exploitation (OS) comme faisant partie du recueil d'information. (4) • Lire la documentation sur la trousse de développement de logiciel en vue d'apprendre à utiliser le logiciel afin de développer des applications pour des appareils spécifiques ou des systèmes d'exploitation. (4) • Lire des manuels de logiciels et les livres de texte écrits par des experts dans cette matière pour repérer les informations spécifiques telles que le diagnostic de panne

		ou comment programmer une fonction spécifique. Ces manuels exigent une connaissance spécialisée de la programmation; les ingénieurs et concepteurs peuvent synthétiser l'information de plusieurs sources. (5)
--	--	--

Sommaire : Lecture

Les symboles >, >> et >>> sont expliqués dans la section Signification des codes.

Type de texte	Objectifs de la lecture			
	Rechercher et repérer des renseignements précis	Feuilleter le texte pour en dégager le sens global, en saisir l'essentiel	Lire le texte en entier pour comprendre ou apprendre	Lire le texte en entier pour le critiquer ou l'évaluer
Formulaires	>>>			
Étiquettes	>>>			
Note, lettre, notes de service	>>>	>>>	>>>	
Manuels, spécifications, règlements	>>>	>>>	>>>	
Rapports, livres, revues spécialisées	>>>	>>>	>>>	

B. Utilisation de documents

Utilisation de documents

Tâches	Niveau de complexité	Exemples
Typiques	1 à 4	<p>Les ingénieurs /ingénieures et les concepteurs/conceptrices en logiciel accomplissent une partie ou l'ensemble des tâches suivantes :</p> <ul style="list-style-type: none">• Parcourir les listes qui définissent quelles sont les différentes fonctions à inclure dans le logiciel. (1)• Repérer des éléments de travail ou des tâches professionnelles à accomplir dans des tableurs. Utiliser les tableurs pour cocher les éléments qui ont été faits. (2)• Se rapporter à des feuilles de données sur le matériel pour comprendre le produit. (2)• Utiliser des formulaires graphiques pour trouver des informations concernant l'historique et la planification. (2)• Utiliser s'il y a lieu des organigrammes pour présenter les algorithmes les plus compliqués. (2)• Saisir de multiples informations dans un logiciel de gestion de projets. Par exemple, ils entrent des détails sur les tâches qui doivent être accomplies ou qui ont été accomplies. (3)• Repérer des informations dans des fichiers numériques tels que des programmes de mise au point (fichiers .gdb) ou des codes source. Lire le langage de programmation. Chaque langage comporte ses propres fonctions et ses propres codes. (3)• Consulter et synthétiser l'information dans des documents techniques pour développer des applications logicielles. Ils se rapportent à des spécifications qui décrivent l'application. (4)
Plus complexes	4	

Sommaire : Utilisation de documents

- Lire des affiches, des étiquettes ou des listes.
- Remplir des formulaires en cochant des cases, en y inscrivant des données numériques, des mots, des expressions, des phrases ou des textes d'un paragraphe ou plus. La liste des tâches varie en fonction de ce qui a été signalé.
- Lire des formulaires déjà remplis contenant des cases à cocher, des données numériques, des expressions, des adresses, des phrases ou des textes d'un paragraphe ou plus.
- Lire des tableaux, des horaires ou d'autres textes présentés sous forme de tableaux.
- Créer des tableaux, des horaires ou d'autres textes présentés sous forme de tableaux.
- Entrer les renseignements tirés de graphiques ou de tableaux.

- Trouver des informations spécifiques tirés de graphiques ou de tableaux
- Interpréter les renseignements tirés de graphiques ou de tableaux.
- Tracer des informations sur des graphiques (par exemple, des diagrammes en lignes, en secteurs ou en barres).
- Obtenir des renseignements particuliers par la lecture de graphiques ou de tableaux.
- Construire ou concevoir des graphiques ou des tableaux.
- Extraire des renseignements des esquisses, des images ou des pictogrammes (p. ex., barre d'outils de l'ordinateur).

C. Rédaction

Rédaction

Tâches	Niveau de complexité	Exemples
Typiques	2 à 4	<p>Les ingénieurs /ingénieures et les concepteurs/conceptrices en logiciel accomplissent une partie ou l'ensemble des tâches suivantes :</p>
Plus complexes	4	<ul style="list-style-type: none"> • Écrire des lettres à des clients pour donner une mise à jour concernant un projet, répondre à une rétroaction ou à des questions et les informer de changements. Ces courriels peuvent devenir une partie des rapports au client. (2) • Écrire des courriels à des réalisateurs de logiciels ou des collègues utilisant la terminologie technique. Écrire aussi, s'il y a lieu, des courriels pour demander conseil auprès de collègues. (2) • Rédiger des leçons apprises pour soi-même ou pour les collègues comme faisant partie du processus d'apprentissage. (3) • Écrire les fichiers d'aide et des guides de l'utilisateur et de mise à niveau qui expliquent comment utiliser les programmes et répondre aux questions que les utilisateurs peuvent avoir. (3) • Rédiger le code source pour un logiciel ou des commentaires techniques sur le codage pour mettre à jour, mettre à niveau dans les langages C++ ou JavaScript. (4) • Écrire des fichiers de mise à jour suivant un format et une syntaxe établis. Par exemple, les fichiers de mise à jour comprennent les consignes de configuration, d'installation et d'exploitation ainsi que des renseignements de diagnostic de panne. (4) • Rédiger la documentation d'un projet qui indique l'ordre des tâches, les problèmes qui sont survenus et les causes des problèmes. Ces renseignements sont utilisés par l'équipe du projet, les nouveaux membres de l'équipe, et

		<p>le directeur de l'équipe comme documentation continue du projet. La documentation peut inclure les rapports à la direction. Les projets de grande envergure demandent plus de paperasserie et d'écriture à communiquer aux actionnaires. (4)</p> <ul style="list-style-type: none"> • Rédiger des rapports d'essai, des rapports d'assurance de la qualité, des rapports de conception et des rapports de situation. Par exemple, les rapports d'assurance de la qualité résument la qualité du logiciel et offrent des informations concernant les techniques de test et des procédures utilisées pour tester le logiciel. Les documents de conception expliquent la manière dont le logiciel sera conçu et sa fonctionnalité. Les rapports de situation indiquent les progrès réalisés, les problèmes survenus et la manière dont ils ont été résolus et les étapes suivantes. (4)
--	--	--

Sommaire : Rédaction

Les symboles >, >> et >>> sont expliqués dans la section Signification des codes.

Longueur	Objectifs de la rédaction						Divertir
	Organiser, retenir	Tenir un dossier, documenter	Informé, obtenir des renseignements	Persuader, justifier une demande	Présenter une analyse ou une comparaison	Évaluer ou critiquer	
Textes comportant moins d'un nouveau paragraphe	>>>	>>>	>>>				
Textes comportant rarement plus d'un paragraphe	>>>	>>>	>>>				
Textes plus longs	>>>	>>>	>>>	>>	>>	>>	

D. Calcul

Les symboles >, >> et >>> sont expliqués dans la section Signification des codes.

Calcul

Tâches	Niveaux de complexité	Exemples
>> Calendriers des budgets et des opérations comptables	3	<p>Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel accomplissent une partie ou l'ensemble des tâches suivantes :</p> <ul style="list-style-type: none"> • Développer s'il y a lieu, des budgets pour les projets. Ils calculent les coûts tels que les heures, les ressources et les frais généraux. Par exemple, ils calculent les coûts des entrepreneurs en multipliant le taux horaire par le nombre d'heures. Ils jugent s'il faut davantage de ressources pour un projet, s'il faut faire des heures supplémentaires ou si le projet doit être réduit en ampleur. (Calendriers des budgets et des opérations comptables), (3) • Établir des calendriers au niveau quotidien, hebdomadaire et mensuel pour les membres de l'équipe et autres, y compris les équipes de production et les groupes des essais. Ils ajustent les calendriers pour tenir compte d'événements imprévisibles et pour respecter les échéances. (Calendriers des budgets et des opérations comptables), (3) • Appliquer des compétences en mathématiques, informatique et ingénierie pour concevoir, développer et tester des applications logicielles. Par exemple, ils écrivent et utilisent des algorithmes et des fonctions algorithmiques pour trier les données et résoudre un problème. Ils utilisent le calcul infinitésimal et l'algèbre linéaire et les mathématiques discrètes. (Mesures et calcul), (5) • Effectuer des calculs sommaires et utiliser les statistiques et probabilités pour analyser et exploiter des données et développer des prévisions au moyen de logiciel d'analyse statistique. Par exemple, ils utilisent un logiciel d'analyse statistique pour calculer la moyenne arithmétique afin d'effectuer une analyse de haut niveau des données du test à présenter à la direction. (Analyse de données numériques), (4) • Faire des estimations lors de la planification et de l'établissement du calendrier d'un projet. (calculs approximatifs), (2)
>>> Mesures et Calculs	5	
>>>> Analyses de données numériques	4	
>> Calculs approximatifs	2	

Sommaire : Compétences en mathématiques

a. Principes mathématiques utilisés

Les symboles >, >> et >>> sont expliqués dans la section Signification des codes.

Principes mathématiques utilisés

Code	Tâches	Exemples
		Concepts numériques
>>>	Nombres entiers	Lire et écrire, compter, arrondir, additionner ou soustraire, multiplier ou diviser des nombres entiers. Par exemple, lorsqu'ils écrivent des algorithmes et des logarithmes.
>>>	Nombres entiers relatifs	Lire et écrire, additionner ou soustraire, multiplier ou diviser des nombres entiers relatifs. Par exemple, lorsqu'ils écrivent des algorithmes et des logarithmes.
		Structure et relations
>>>	Équations et formules	Résoudre des problèmes en établissant et en résolvant des équations à une inconnue; utiliser des formules en insérant les quantités représentant des variables et les résoudre; écrire, simplifier et résoudre des problèmes algébriques à deux variables; écrire, simplifier et résoudre des équations quadratiques. Par exemple, ils écrivent et utilisent des algorithmes et des fonctions logarithmiques pour trier les données ou résoudre un problème tels que programmer des contrôles audio.
		Statistiques et probabilités
>>>	Calculs sommaires	Calculer des moyennes. Par exemple, utiliser des tableaux, des horaires et d'autres textes présentés sous forme de tableaux. Utiliser des représentations graphiques. Par exemple, utiliser la moyenne arithmétique pour effectuer une analyse de haut niveau des données de test. Ils utilisent un utilitaire de nombres aléatoires pour garantir le caractère aléatoire. Ils mesurent l'usage moyen d'une unité centrale au moyen de paramètres tels que la vitesse du processeur et la mémoire disponible. Ils calculent, s'il y a lieu, le nombre d'instructions pouvant être exécutées dans un temps donné.
>>>	Statistiques et probabilités	Utiliser des statistiques descriptives (c'est-à-dire recueillir, classifier, analyser et interpréter des données). Utiliser des statistiques déductives (c'est-à-dire utiliser des théories mathématiques de probabilité, tirer des conclusions à propos d'une population ou à propos de la probabilité qu'un événement se produise). Par exemple, ils utilisent un logiciel d'analyse statistique pour analyser et exploiter des données et arriver à des prévisions. Ils créent et utilisent la logique et une table de vérité pour analyser des énoncés afin de vérifier s'ils sont logiques et vrais ou non.

b. Méthodes de calcul

- Mentalement.
- À l'aide d'une calculatrice.
- À l'aide d'un ordinateur.

c. Instruments de mesure utilisés

- Temps. Par exemple, utiliser les horloges et les montres.

E. Communication orale

Communication orale

Tâches	Niveau de complexité	Exemples
Typiques	2 à 3	<p>Les ingénieurs /ingénieures et les concepteurs/conceptrices en logiciel accomplissent une partie ou l'ensemble des tâches suivantes :</p> <ul style="list-style-type: none">• Assister à des réunions avec des réalisateurs pour communiquer des mises à jour de projet, discuter des problèmes qui se posent et la manière dont ils peuvent être résolus, poser des questions et communiquer l'état du projet. (2)• Assister à des réunions pour discuter de la distribution des tâches et d'idées concernant l'exécution des tâches. (2)• Adapter le niveau de langue en fonction des personnes à qui on parle. Par exemple, lorsqu'ils parlent à des directeurs, ils emploient une langue beaucoup moins technique. Ils posent des questions pour se former une opinion des exigences du projet ou obtiennent de l'information telle que des mots de passe pour ouvrir une session. Ils obtiennent aussi les exigences établies du chef de service des produits. Ils utilisent la langue technique lorsqu'ils parlent avec des ingénieurs et doivent faire preuve de tact lorsqu'ils répondent à des questions, ou lorsqu'ils donnent ou demandent de l'aide. (2)• Assister à des réunions avec des collaborateurs, des collègues et des superviseurs pour résoudre des problèmes liés à des logiciels, pour discuter de nouveautés et pour recueillir des renseignements à propos d'un projet. Par exemple, ils font un remue-méninge pour trouver des idées afin de résoudre des bogues ou pour discuter d'opinions et échanger des renseignements. À la fin des projets, ils se rencontrent pour analyser comment s'est déroulé le projet et les leçons qui ont été apprises. (3)• Organiser, s'il y a lieu, des réunions pour encourager
Plus complexes	3	

		<p>l'échange d'idées entre experts du sujet. (3)</p> <ul style="list-style-type: none"> • Former, enseigner et conseiller s'il y a lieu des étudiants en co-op ou des réalisateurs juniors. Par exemple, ils se mettent à leur disposition pour répondre à leurs questions et montrer comment faire certaines choses, leur indiquer comment planifier et documenter des tâches et pour discuter de manière à améliorer leurs compétences. (3) • Rencontrer les clients en personne ou au téléphone pour discuter du calendrier ou pour clarifier des détails concernant les exigences du projet. (3) • Faire des présentations à des collaborateurs et des chefs de service pour partager des connaissances. Par exemple, ils peuvent présenter les projets actuels ou des nouveautés technologiques. (3)
--	--	--

Modes de communication utilisés

- En personne. Par exemple, parler avec des collègues de questions concernant le projet.
- Par téléphone. Par exemple, parler avec les clients de mises à jour du projet
- Autre. Par exemple, organiser des vidéo conférences avec des clients.

Facteurs environnementaux ayant une incidence sur la communication

Aucun signalé.

Sommaire : de communication orale

Les symboles >, >> et >>> sont expliqués dans la section Signification des codes.

Objectifs de la communication orale (Partie I)						
Type	Accueillir	Prendre des messages	Donner ou recevoir des renseignements, des explications, des directives	Chercher ou obtenir des renseignements	Coordonner son travail avec celui des autres	Rassurer, reconforter
Écouter (peu ou pas d'interaction)			>>			
Parler (peu ou pas d'interaction)			>>			
Interagir avec les collègues			>>>	>>>	>>>	
Interagir avec les subalternes			>>>	>>>	>>>	
Interagir avec les supérieurs			>>>	>>>	>>>	
Interagir avec des pairs et collègues d'autres organisations			>>	>>	>>	
Interagir avec les clients ou le grand public			>>	>>		
Interagir avec les fournisseurs de produits et de services						
Participer à des discussions de groupe			>>>	>>>	>>>	

Présenter de l'information à un petit groupe			>>	>>		
Présenter de l'information à un grand groupe						

Les symboles >, >> et >>> sont expliqués dans la section Signification des codes.

Objectifs de la communication orale (Partie II)						
Type	Discuter, échanger des renseignements, des opinions	Persuader	Faciliter, animer	Transmettre des connaissances, susciter la compréhension, le savoir	Négocier, résoudre des conflits	Divertir
Écouter (peu ou pas d'interaction)			>>			
Parler (peu ou pas d'interaction)			>>			
Interagir avec les collègues	>>>	>>>	>>>	>>>	>>>	
Interagir avec les subalternes	>>			>>		
Interagir avec les supérieurs	>>>					
Interagir avec des pairs et collègues d'autres organisations	>>					
Interagir avec les clients ou le grand public	>>	>>		>>	>>	
Interagir avec les fournisseurs de produits et de services						
Participer à des discussions de groupe	>>>	>>>		>>>		
Présenter de l'information	>>		>>	>>		

à un petit groupe						
Présenter de l'information à un grand groupe						

F. Capacité de raisonnement

1. Résolution de problèmes

Résolution de problèmes

Tâches	Niveau de complexité	Exemples
Typiques	2 à 4	<p>Les ingénieurs /ingénieures et les concepteurs/conceptrices en logiciel accomplissent une partie ou l'ensemble des tâches suivantes :</p>
Plus complexe	4	<ul style="list-style-type: none">• S'occuper de la communication et des situations de travail d'équipe. Dans de gros organismes, plusieurs personnes peuvent travailler sur le même projet mais les équipes ne communiquent pas efficacement entre elles. (2)• Anticiper les questions des clients, par exemple, demander comment exécuter une tâche spécifique avec le logiciel. Apporter des changements, s'il y a lieu, dans les fonctions qui existent déjà ou créer de nouvelles fonctions. (3)• Trouver une conception appropriée en se basant sur les exigences des clients ou sur l'infrastructure existante du logiciel. Si l'infrastructure actuelle du logiciel est faible, ils devront découvrir des manières de contourner le problème ou des solutions qui pourraient nécessiter une reconstruction de l'infrastructure. (4)• Découvrir comment nettoyer ou réécrire un code qui a été écrit par une autre personne. Si le code s'est accumulé au fil du temps, cela pourrait bien prendre plusieurs mois. (4)• Effectuer un diagnostic de panne et réparer les problèmes dus aux bogues ou aux défauts. Ils consultent leurs collaborateurs, font des tests pour déterminer où se trouvent les problèmes, utilisent des outils de mise au point, puis refont les tests. Par exemple, ils peuvent utiliser des données sur les temps d'exécution. Ils décomposent le problème en plusieurs étapes pour donner des instructions machine et s'assurer du fonctionnement. Ils peuvent construire un prototype pour effectuer un diagnostic de panne. (4)

2. Prise de décision

Prise de décision

Tâches	Niveau de complexité	Exemples
Typiques	2 à 4	<p>Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel accomplissent une partie ou l'ensemble des tâches suivantes :</p>
Plus complexes	4	<ul style="list-style-type: none">• Décider, s'il y a lieu, quelle tâche attribuer aux membres de l'équipe. Ils estiment les compétences et l'expérience de chaque personne. (2)• Anticiper ce qui doit être fait dans un projet pendant l'étape de la conception. Par exemple, on leur donne une spécification et ils doivent concevoir une solution qui correspond à la spécification. Ils décident de l'outil à utiliser. (2)• Décider où faire des ajustements au budget lorsqu'il n'y a pas assez pour terminer le projet. Ils peuvent réduire l'envergure du projet, inclure les heures supplémentaires ou embaucher plus d'entrepreneurs. (3)• Examiner les plans de projets, rechercher les spécifications et la technologie ainsi que les algorithmes pour juger s'il faut davantage de ressources. Par exemple, si l'entreprise a assez d'entrepreneurs et s'il faut en embaucher davantage pour terminer le projet. (3)• Déterminer quelle conception mettre en œuvre en fonction des avantages et des inconvénients de chaque option. Ils pourraient décider parmi plusieurs options de conception toutes adaptées aux exigences du client. Par exemple, la meilleure solution technique peut prendre plus de temps et exiger l'embauche de plus d'entrepreneurs, et par conséquent, entraîner une livraison tardive et des coûts plus élevés. Une solution plus rapide pourrait correspondre aux besoins actuels mais être plus difficile à maintenir à long terme. (4)

3. Pensée critique

Pensée critique

Tâches	Niveau de complexité	Exemples
Typiques	2 à 4	<p>Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel accomplissent une partie ou l'ensemble des tâches suivantes :</p> <ul style="list-style-type: none"> Évaluer la qualité de travail effectué par des collaborateurs et de nouveaux membres juniors de l'équipe. Ils peuvent évaluer la documentation pour s'assurer de sa pertinence, son exactitude et de la clarté. (2) Tester, résoudre des problèmes et analyser les logiciels qui peuvent se prolonger à travers de multiples sous systèmes de logiciels. Par exemple, un logiciel qui a des implications sur la sécurité exige plusieurs vérifications. (4) Évaluer les options de conceptions. Par exemple, ils peuvent avoir une solution pour un client basé sur un projet antérieur. Ou un client demande peut-être certaines fonctions qui, selon l'analyse de tendances, passera de mode en peu de temps. Cela implique peut-être de développer une solution rapide qui fonctionnera pour un temps sans investir beaucoup de ressources et qui permettra au personnel technique de passer à la génération suivante de produits. (4)
Plus complexe	4	

4. Planification et organisation du travail

Planification et organisation du travail

Niveau de complexité	Description
3	<p>Planification de son propre travail :</p> <ul style="list-style-type: none"> Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel planifient leur travail indépendamment. Ils examinent les priorités et modifient l'ordre des tâches en réponse aux demandes des chefs de service et d'autres membres de l'équipe ou des modifications apportées au projet. <p>Planification organisationnelle :</p> <ul style="list-style-type: none"> Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel attribuent des tâches aux membres juniors de l'équipe. Ils peuvent coordonner les activités de l'équipe de développement.

5. Utilisation particulière de la mémoire

Exemples

- Se souvenir où trouver les dossiers dans leur ordinateur.
- Se souvenir des mots de passe et des numéros d'identification.
- Se souvenir des problèmes qui ont été résolus avant et où le problème a été résolu afin qu'il y ait une référence concernant la manière de résoudre des problèmes de même nature.
- Se souvenir de problèmes et de bogues de programmation antérieurs et utiliser ce renseignement pour résoudre le problème.
- Se souvenir des informations et des séquences de programmation.

6. Recherche de renseignements

Recherche d'information

Tâches	Niveau de complexité	Exemples
Typiques	3	<p>Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel accomplissent une partie ou l'ensemble des tâches suivantes :</p> <ul style="list-style-type: none">• demander à des collègues et des membres d'autres équipes des informations sur la façon de traiter les questions. Par exemple, ils se renseignent sur des questions de bogues lors des réunions d'équipe, échangent des courriels pour demander des conseils et posent des questions sur le code de programmation au cours de présentations de perfectionnement professionnel. (3)• trouver des informations en ligne. Ils demandent des informations à des groupes d'utilisateurs de Java ou à des groupes d'utilisateurs de gestionnaire de système sur, par exemple, les manières d'éviter les pannes de système. Ils recherchent des informations en ligne à l'aide de Google et visitent les forums pour en savoir plus sur les problèmes spécifiques à la langue de l'ordinateur. (3)• rechercher des informations sur le code de programmation tout en faisant la programmation régulière. Par exemple, ils recherchent des manuels techniques, des services d'assistance et d'autres ressources pour résoudre les bogues et autres défauts. Ils recherchent les fichiers gdb pour trouver des solutions à des problèmes de code. (3)
Plus complexe	3	

G. Travail d'équipe

Travail d'équipe

Niveau de complexité	Description
3	Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel travaillent en tant que membres d'une équipe. Ils coordonnent et harmonisent leur travail avec d'autres ingénieurs et concepteurs de logiciels, les programmeurs informatiques, les chefs de projets, les chefs de produits, les ingénieurs commerciaux et autres.

Participation aux activités de supervision et de leadership

- Participer à des discussions formelles sur les méthodes de travail ou l'amélioration des produits.
- Formuler des suggestions afin d'améliorer les méthodes de travail.
- Superviser le rendement d'autres employés.
- Renseigner d'autres employés ou leur enseigner certaines tâches.
- Orienter les nouveaux employés.
- Formuler des recommandations d'embauche.
- Prendre des décisions d'embauche.
- Choisir des entrepreneurs et des fournisseurs.
- Assigner des tâches courantes à d'autres employés.
- Assigner des tâches nouvelles ou inhabituelles à d'autres employés.
- Déterminer la formation nécessaire qui serait utile à d'autres employés.
- Traiter les griefs ou les plaintes des autres employés.

H. Compétences numériques

Compétences numériques

Tâches	Niveau de complexité	Exemples
Typique	2 à 5	Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel accomplissent une partie ou l'ensemble des tâches suivantes :
Plus complexe	5	<ul style="list-style-type: none">• Utiliser des logiciels de communication. Par exemple, ils échangent des courriels et y ajoutent des annexes, avec des collègues et des clients pour discuter les détails des projets, organiser des réunions et envoyer des rapports aux clients. (2)• Utiliser des applications qui intègrent la vidéo, la voix et le

		<p>partage de données lors de conférences et de réunions. (2)</p> <ul style="list-style-type: none"> • Utiliser des éditeurs visuels tels que VI et Emacs qui affichent le texte en cours d'édition sur l'écran. Les éditeurs ont des compilateurs qui exécutent le programme et transforment le fichier source dans une forme qui peut être exécutée (fichier binaire). Ils peuvent aussi utiliser les éditeurs de texte tels que le Bloc-notes, qui est uniquement un éditeur de texte. (2) • Visualiser la conception d'un système utilisant Unified Modeling Language (UML) qui donne une représentation graphique de la version finale du logiciel. (2) • Utiliser Internet pour rechercher des informations. Par exemple, ils utilisent des navigateurs Internet pour accéder aux sites Web des fournisseurs, des publications spécialisées en ligne et des forums pour obtenir des informations sur le code de programmation et des lignes directrices sur le dépannage. Ils utilisent des plateformes telles que SharePoint pour intégrer la gestion de contenu intranet et la gestion des documents. (3) • Utiliser un logiciel de gestion de projet comme JIRA et Projets de Microsoft pour attribuer le travail et suivre l'activité de l'équipe, suivre les bogues et les problèmes, et de partager des fichiers à leur ordinateur de bureau ou à leur portable. (3) • Utiliser un logiciel de présentation: tels que PowerPoint pour créer des diaporamas pour les collègues et les clients. Par exemple, ils créent des présentations de perfectionnement professionnel avec des collègues. (3) • Utiliser un logiciel de traitement de texte tel que Word pour créer, éditer et formater des documents comme des rapports de projets et des rapports d'essais, des notes de service, de référence, des procédures et des calendriers. (3) • Utiliser un logiciel d'analyse statistique pour analyser, interpréter et exploiter les informations et construire des graphiques d'analyse, des cartes et des tableaux. (4) • Développer et déboguer des programmes informatiques. Par exemple, ils utilisent des environnements de développement intégrés (IDE) comme Eclipse et QT Creator pour développer et déboguer les erreurs et IDE tel que Visual Studio pour développer des programmes informatiques ainsi que des applications Web. Les ingénieurs et les concepteurs en logiciel utilisent des systèmes de contrôle de code source pour contrôler les changements et suivre le développement du code source. Ils utilisent des langages de programmation, tels que JavaScript, C ++ et Ruby pour développer des programmes informatiques tels que des applications, des utilités, des serveurs et des programmes de système. (5)
--	--	---

Sommaire : Compétences numériques

- Utiliser un logiciel de traitement de texte
- Utiliser un logiciel de graphisme
- Utiliser une base de données
- Utiliser des tableurs
- Utiliser un logiciel de communication

I. Formation continue

Formation continue

Niveau de complexité	Description
4	Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel acquièrent constamment de nouvelles compétences, parce que la technologie évolue si rapidement. Ils doivent être en mesure d'acquérir rapidement les connaissances nécessaires car ils vont d'un projet à un autre. Ils apprennent sur le tas, des autres membres de l'équipe et de collègues ayant plus d'expérience qu'eux. Les ingénieurs et les concepteurs de logiciels chevronnés servent de mentor aux ingénieurs et aux concepteurs juniors. Ils peuvent recevoir une formation par le travail, en ligne ou suivre des cours universitaires. Ils font des recherches et trouvent de l'information sur le Web.

Les connaissances peuvent être acquises :

- Dans le cadre des activités habituelles de travail.
- Auprès de collègues.
- Par une formation en cours d'emploi.
- Par la lecture ou par d'autres formes d'apprentissage autodidactique.
 - Au travail.
 - En dehors des heures de travail.
 - À l'aide de matériel fourni au travail.
 - À l'aide de matériel fourni par une association professionnelle ou un syndicat.
 - À l'aide de matériel que l'employé a trouvé de sa propre initiative.
- Par une formation externe.
 - Durant les heures de travail, sans frais pour l'employé.
 - Partiellement subventionnée.

J. Renseignements supplémentaires

Les entrevues menées auprès des titulaires de poste ont permis de recueillir les renseignements qui ont servi à élaborer le présent profil des compétences essentielles et de leur poser des questions sur les sujets suivants:

Aspects physiques

Principalement assis.

Attitudes

Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel doivent être des apprenants rapides, avoir de bonnes compétences en recherche et une approche créative à la résolution de problèmes. Ils doivent avoir une attitude positive par rapport au travail et à la communication dans l'équipe et la capacité de travailler avec des personnes de tous les niveaux. Ils doivent pouvoir travailler sous pression et respecter les échéances.

Répercussions des technologies numériques

Toutes les compétences essentielles sont touchées par l'introduction de la technologie dans les lieux de travail. La capacité d'adaptation des ingénieurs et concepteurs de logiciels aux nouvelles technologies est fortement liée à leur niveau de compétences essentielles, notamment en lecture, en rédaction, en raisonnement et en communication. Les technologies transforment les manières dont les travailleurs obtiennent, traitent et communiquent l'information et les types de compétences nécessaires pour exécuter leur travail. Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel ont particulièrement besoin de compétences numériques avancées pour travailler avec des technologies qui évoluent rapidement. Par exemple, ils ont besoin de compétences en conception et en dépannage pour écrire et éditer le code, utiliser des logiciels et des plateformes pour gérer et développer des fichiers et des programmes. Ils doivent aussi posséder les compétences nécessaires pour résoudre les problèmes potentiels à la satisfaction des clients et gérer des tâches et les échéances de projets. Les nouveautés en matière de technologie numérique exigent de ces professionnels de perfectionner leurs compétences continuellement afin de rester au courant des évolutions.

La technologie au travail a aussi des répercussions sur la complexité des tâches liées aux compétences essentielles demandées pour ce métier. Les ingénieurs/ingénieures et les concepteurs/conceptrices en logiciel ont besoin de leurs compétences pour utiliser des logiciels et des plateformes de plus en plus complexes et spécialisées. Par exemple, ils utilisent TFS et IDE pour écrire le code source. D'autre part, l'utilisation de la technologie pour rechercher de l'information et communiquer avec les autres dans le domaine facilite la recherche de solutions aux problèmes pour les travailleurs.

K. Notes

Ce profil d'emploi table sur les entrevues avec des titulaires et a été validé par l'entremise de consultations avec les spécialistes du secteur à l'échelle du pays.

Pour plus d'information sur la recherche, les définitions et les processus de mise à l'échelle des profils des compétences essentielles, s'il vous plaît consulter le Guide d'interprétation des profils de compétences essentielles