

EVALUATIONS CERTIFICATIVES TITRE DE NIVEAU 7 EXPERT EN ARCHITECTURE ET DEVELOPPEMENT LOGICIEL 2025-2027

Référentiel d'activités	Référentiel de compétences	Référentiel d'évaluation	
		Modalités d'évaluation	Critères d'évaluation
Bloc 1. Planifier et organiser un projet de développement logiciel			
Modules en lien avec le BLOC 1 : Piscine en JS – remise à niveau, Méthode Agile, DevOps CI/CD, Gestion de projets, Veille technologique, Modélisation : BPMN (alternative UML), Urbanisation des SI, Industrialisation des SI, Green IT, RGPD, DPO, Rédiger & Répondre à un appel d'offre, Architecture Web, Evaluation des risques & PCA / PRA, Projet StartUp Agile & DevOps, Projet transverse et préparation de la soutenance			
A1 Organisation du projet de développement logiciel			
A1.1. Evaluation de la faisabilité du projet de développement logiciel	C1.1 Réaliser une étude de faisabilité du projet de développement logiciel , en analysant l'ensemble des éléments organisationnels, techniques et financiers, afin de déterminer les moyens de réalisation du projet de développement logiciel	C1.1 à C1.7 – Mise en situation professionnelle reconstituée, portant sur la conception d'un projet d'architecture logicielle, s'inscrivant dans l'orientation stratégique d'une entreprise. L'évaluation prendra la forme d'un dossier écrit et d'une présentation devant un jury composé de 2 professionnels minimum. Le candidat présente son projet pendant 20 mn devant le jury, suivi d'un entretien avec le jury de 20 mn sur le projet.	Ce1.1.1 L'analyse des besoins, de l'environnement et des risques du projet de développement logiciel permet une vision synthétique des objectifs et des contraintes du projet Ce1.1.2 L'analyse des objectifs et des contraintes (budgétaires, RH, techniques, temporelles, organisationnelles, légales, environnementales, etc.) du projet de développement logiciel permet de vérifier si la mise en place du plan sera possible et d'estimer le niveau de risque Ce1.1.3 La faisabilité (économique, technique et opérationnelle) du projet de développement logiciel est analysée par rapport au retour sur investissement du projet
A1.2 Élaboration du plan de veille technologique	C1.2 Concevoir un système de veille technologique , en analysant l'évolution technologique du secteur et de la concurrence, en comparant un assemblage d'outils, de modèles d'architectures logicielles, de solutions informatiques, d'usages et de services existants, afin d'identifier les apports les plus adaptés au projet, et de proposer une solution logicielle optimale, opérationnelle et innovante à la demande		Ce1.2.1 Le plan de veille technologique est conçu pour répondre aux objectifs et contraintes du projet de développement logiciel Ce1.2.2 Le plan de veille technologique s'appuie sur un ensemble d'outils et méthodes qui permettent d'identifier les informations pertinentes au projet de développement logiciel

	du client	
--	-----------	--

		Ce1.2.3 L'analyse des informations issues de la veille technologique permettent d'inclure des apports opérationnels et innovants à la solution de développement logiciel
--	--	--

A1.3 Description de la solution informatique proposée Rédaction du cahier de charges fonctionnel	C1.3 Concevoir la solution informatique associée au projet, en termes de spécifications techniques et de fonctionnalités, en identifiant les niveaux de performance cible, et les risques techniques et fonctionnels, afin de répondre aux besoins du client	C.1.3.1 L'analyse fonctionnelle permet de décrire et lister les fonctionnalités attendues de la solution informatique dans le cahier des charges fonctionnel, en adéquation avec les objectifs du projet C.1.3.2 Les fonctionnalités, niveaux de performance cible et les contraintes (techniques, réglementaires, budgétaires, etc.) identifiées pour la solution informatique sont adéquates par rapport aux besoins identifiés des utilisateurs C.1.3.3 Les principes d'accessibilité sont respectés, et la solution proposée est adaptée aux exigences et besoins des personnes en situation de handicap C.1.3.4 Les principes du numérique responsable sont respectés, en particulier en termes d'impact environnemental de la solution informatique proposée
A1.4 Définition du plan du projet de développement logiciel	C1.4 Établir la feuille de route du projet de développement logiciel, en intégrant la méthodologie de gestion de projet (ex. Agile, DevOps), l'organigramme des tâches (tableaux Kanban), les livrables à produire, les échéances, les budgets, et l'ensemble des ressources disponibles (humaines, matérielles, etc.), les outils de pilotage et les indicateurs de performance, afin de garantir l'opérationnalité du projet informatique	C.1.4.1 Les éléments du plan du projet de développement logiciel s'appuient sur une méthodologie spécifique (ex. Agile, DevOps) C.1.4.2 L'organisation et la gestion des tâches du projet s'appuie sur un outil pertinent (ex. tableaux Kanban) et permet une utilisation efficace des ressources de l'équipe C.1.4.3 La planification précise des livrables et des budgets est en accord avec les objectifs, ressources et contraintes du projet C.1.4.4 Les outils de pilotage et les indicateurs de performance définis permettent d'évaluer l'avancement du projet
A1.5 Implémentation du plan du projet de développement logiciel	C1.5 Mettre en place le projet de développement logiciel, selon la méthodologie définie (ex. Agile, DevOps), en suivant le plan de gestion des ressources (technologiques, budgétaires, humaines, etc.), le plan de gestion de risques, le plan de gestion du	C.1.5.1 Le plan du projet de développement logiciel est déployé selon la méthodologie choisie (ex. Agile, DevOps) C.1.5.2 L'implémentation du plan du projet de développement logiciel respecte le budget, la gestion établit des ressources humaines et matérielles, les risques identifiés, et minimise l'impact sur l'environnement

	changement, dans le respect du calendrier, en intégrant les enjeux de l'informatique durable, en répondant aux besoins spécifiques des personnes présentant un handicap, afin d'optimiser la performance de l'application logicielle		C.1.5.3 La mise en place du plan du projet de développement logiciel respecte les échéances, les normes associées aux objectifs définis et les exigences identifiées pour les personnes présentant un handicap
A1.6 Suivi de l'évolution du projet de développement logiciel	C1.6 Mesurer la performance du projet de développement logiciel , selon des indicateurs de performance (KPI) établis, en déterminant l'état d'avancement, en veillant à ce que le projet respecte les objectifs, le calendrier et les budgets établis, les normes réglementaires (RGPD) et de qualité (ITIL), afin de définir les éléments d'amélioration et de garantir son bon déroulement		C.1.6.1 Le suivi de l'évolution du projet de développement logiciel est réalisé par le biais des indicateurs clefs de performance (KPI) définis C.1.6.2 Les outils utilisés permettent de mesurer l'efficacité et l'avancement du projet de développement logiciel (calendrier, budget, qualité des livrables) par rapport au plan initial C.1.6.3 La mesure de la performance permet de mesurer la conformité du projet aux normes réglementaires, telles que le RGPD, et aux standards de qualité, comme ITIL ou les préconisations de l'ANSSI C.1.6.4 L'analyse des informations issues de l'évaluation du projet de développement logiciel permet d'identifier les axes et propositions concrètes d'amélioration du projet
A1.7 Contrôle du projet de développement logiciel	C1.7 Superviser le déroulement du projet de développement logiciel , en analysant les indicateurs de performance, en implémentant la résolution des problèmes identifiés, en adaptant le calendrier et les ressources fixées, en communiquant avec les parties prenantes, afin d'actualiser le projet de développement logiciel, de maintenir le cycle de vie de du développement du logiciel.		C.1.7.1 L'identification des actions nécessitant des corrections permet d'actualiser le projet de développement logiciel C.1.7.2 La résolution de problèmes permet de contrôler le projet de développement logiciel par la définition des nouvelles tâches à réaliser C.1.7.3 Le projet de développement logiciel est mis à jour par des ajustements effectuées au niveau du calendrier et des ressources C.1.7.4 Les parties prenantes et notamment le client final sont informés du déroulement du projet par le biais d'une communication efficace

Référentiel d'activités	Référentiel de compétences	Référentiel d'évaluation	
		Modalités d'évaluation	Critères d'évaluation
Bloc 2. Concevoir et développer des solutions logicielles			
Modules en lien avec le BLOC 2 : UX / UI, ReactJS / GitHub, NextJS / Vercel, Projet Front-End, React Native, Découverte de NodeJS / API, Mysql / Merise, MongoDB & Cloud Computing, Projet Back-End, Architecture Web, Introduction Cybersécurité et E-Réputation, ISO 27001, Modèle OSI, Docker & Containerisation (Kubernetes)—orchestration, Introduction au Big Data, Python – Pandas / Numpy, Python Machine Learning, Python Spark (Big Data) + Kafka, Natural Language Processing & Computer Vision, MLOps, Projet 3 – Big Data & IA			
A2 Construction de la solution technique du projet de développement logiciel			
A2.1 Conception de la schématisation de l'architecture de la solution logicielle	C2.1 Élaborer l'architecture de l'application logicielle , selon les contraintes techniques, les spécifications et les fonctionnalités attendues, les composants de l'interface déterminés, en respectant les exigences de sécurité et les normes réglementaires, en utilisant un langage de modélisation (ex. BPMN, UML), des outils numériques (ex. draw.io) et une méthodologie adaptée (ex. analyse PESTEL), afin de proposer un prototype de la solution informatique qui répond aux scénarios d'utilisation retenus.	C2.1 à C2.5 – Mise en situation professionnelle reconstituée, portant sur la conception et le développement d'une solution informatique innovante (ex. logiciel, application web ou mobile, solution visant l'Internet des objets (IoT), machine learning, etc.). L'évaluation prendra la forme d'un dossier écrit et d'une présentation devant un jury de composé de 2 professionnels minimum. Le candidat présente son projet pendant 20 mn devant le jury, suivi d'un entretien avec le jury de 20 mn sur le projet .	Ce2.1.1 La schématisation de l'architecture de la solution logicielle répond aux spécifications techniques et fonctionnelles, ainsi qu'aux contraintes identifiées Ce2.1.2 L'architecture de l'application logicielle répond aux cas d'utilisation identifiés et aux contraintes réglementaires et de sécurité C2.1.3 Les langages de modélisations (ex. BPMN, UML) et les outils utilisés (ex. Draw.io) permettent de conceptualiser et visualiser clairement l'architecture de l'application C2.1.4 L'architecture proposée est conformes aux exigences de sécurité et aux normes réglementaires C2.1.5 La solution d'architecture logicielle proposée répond aux divers scénarios d'utilisation, et elle est adaptée aux personnes en situation de handicap (ex. compatibilité avec les lecteurs d'écran et les sous-titres) assurant ainsi une réponse adaptée et inclusive aux besoins de tous les utilisateurs. C2.1.6 L'architecture proposée permet de réduire l'impact environnemental de la solution (ex. pas de redondance inutile, utilisation de micro services, de la virtualisation et de la conteneurisation pour réduire l'utilisation des ressources, gestion dynamique des ressources, caching etc.)

A2.2 Préparation du développement continu de logiciels

C2.2 **Préparer l'intégrité du code**, en définissant des indicateurs de référence dans une démarche de qualité, en établissant des listes de contrôle détaillées des livrables, en respectant la conformité réglementaire (normes RGPD), en s'appuyant sur des outils automatisés pour garantir le contrôle de version

C.2.2.1 Les processus et les outils à mettre en place pour intégrer la sécurité du code dès le début du cycle de développement (shift-left) sont clairement spécifiés

C.2.2.2 Les indicateurs de référence définis permettent de garantir continuellement la qualité du code

	du code (ex. GitHub), en assurant le contrôle de la performance, afin de réduire le cycle de développement, de veiller au développement en toute sécurité et de garantir un produit final sans faille de sécurité dans le code		C.2.2.3 Les listes de contrôle permettent d'assurer la complétude, la conformité, l'intégrité et la qualité du code tout au long du processus de développement C.2.2.4 Les exigences attendues sur la sécurité dans le développement et la confidentialité du code source, ainsi que le respect réglementaire des normes (ITIL, RGPD) sont intégrés dans le cadre du projet tout au long du processus de développement C.2.2.5 Les outils et les fonctionnalités collaboratives de versionning (ex. GitHub) sont clairement précisés et permettent de gérer les versions et les mises à jour et d'éviter les conflits C.2.2.6 Les bonnes pratiques de codage établies permettent l'identification précoce des vulnérabilités, la réduction du temps de mise en production et une gestion des performances adéquate
A2.3 Développement front-end (côté client)	C2.3 Piloter le développement front-end d'une application ou d'un site web , en gérant les contenus graphiques, interactifs et dynamiques, en intégrant les dernières évolutions des outils et techniques de développement, avec des langages spécifiques (ex. HTML5, CSS3, JavaScript), des bibliothèques de fonctions, des frameworks (ex. ReactJS, AngularJS, Vue.js), en collaboration avec d'autres professionnels (ex. web-designer, UX/UI designer), afin de proposer une navigation fluide et une interface ergonomique et inclusive, accessible à tous les utilisateurs, y compris en situation de handicap		C.2.3.1 Les outils et les solutions de développement utilisés pour le front-end s'appuient sur la veille technologique des meilleures pratiques et standards actuels C.2.3.2 Les éléments graphiques, dynamiques et le design responsive de l'interface utilisateur intègrent avec succès les contributions d'autres professionnels (ex. web-designer, UX/UI designer) pour enrichir le projet et améliorer la qualité du produit final C.2.3.3 L'application intègre l'ensemble des dernières évolutions des outils et techniques de développement front-end, en utilisant des langages spécifiques tels que HTML5, CSS3, JavaScript, et en mettant à jour et appliquant les meilleures pratiques et standards actuels dans le développement front-end C.2.3.4 L'utilisation des bibliothèques de fonctions et de frameworks (ex. ReactJS, AngularJS, VueJS) répond aux besoins spécifiques du projet et améliore la fonctionnalité et l'interactivité des applications

			C.2.3.5 L'interface propose une navigation fluide, ergonomique, intuitive et inclusive, et elle est accessible à tous, y compris aux personnes en situation de handicap (ex. sous-titres, transcriptions, etc.)
A2.4 Développement back-end (côté serveur)	C2.4 Piloter le développement back-end d'une application ou d'un site web , concernant des solutions techniques et fonctionnelles, en gérant le serveur, des bases de données, la liaison avec le front-end (API), le back-office, en intervenant sur le choix des outils, des technologies (ex. SGBDR, versioning), des systèmes d'exploitation et des ressources réseaux, avec des langages objet (ex. Python, Java, Node.js), selon des standards techniques (ex. W3C) et des normes de sécurisation (OWASP, RGPD), en collaborant avec d'autres équipes, afin de rendre la solution informatique fonctionnelle .		C.2.4.1 L'application back-end répond aux objectifs et aux contraintes techniques identifiées C.2.4.2 Le back-office permet la gestion de l'application dans le respect du cahier des charges et la liaison avec le front-end est assurée par APIs C.2.4.3 Les outils, technologies et langages objet (Python, Java ou Node.js) sont correctement mis en oeuvre pour répondre aux besoins spécifiques du projet et en les utilisant pour développer des solutions back-end robustes et conformes aux normes de sécurisation C.2.4.4 La configuration des serveurs et des bases de données permet de répondre aux objectifs et contraintes du projet, et d'assurer la sécurité des données C.2.4.5 Les standards techniques, les normes de sécurisation (ex. W3C, OWASP) et les conformités réglementaires (ex. RGPD) sont respectés C.2.4.6 Les choix d'implémentation garantissent une utilisation efficace des ressources et permet de minimiser l'impact environnemental dans le cadre d'un développement numérique responsable (ex. code efficace et optimisé, gestion dynamique des ressources, caching etc.)
A2.5 Traitement des données massives	C2.5 Coordonner le développement des solutions technologiques visant le traitement de la donnée à grande échelle , comportant la collecte (ex. avec Open Refine ou Nifi), le traitement par lots ou en temps réel (ex. Apache Spark ou Storm) et le stockage de données massives structurées ou non structurées, en utilisant des bases de données analytiques ou opérationnelles, et en utilisant l'analyse descriptive ou prédictive (ex. machine learning), afin d'extraire toute la valeur des		C.2.5.1 La mise en œuvre de la collecte et du traitement par lot ou en temps réel des données par des outils technologiques (ex. Open Refine, Nifi, Apache Spark ou Storm) permet de répondre aux contraintes techniques et fonctionnelles du projet (ex) C.2.5.2 La mise en œuvre du stockage de données massives structurées ou non structurées, dans des bases de données analytiques ou opérationnelles, permet de

	données disponibles pour l’aide à la décision et automatisation de tâches répétitives		répondre aux contraintes techniques et fonctionnelles du projet, et d’assurer la sécurité et l’intégrité des données C.2.5.3 La mise en œuvre d’analyses descriptives et prédictives sur des données massives, y compris l’utilisation de techniques de machine learning, permettent d’extraire de la valeur de la donnée de manière innovante et de répondre aux objectifs et contraintes du projet C.2.5.4 Les choix technologiques, les frameworks et les outils utilisés tout au long du projet sont présentés clairement, de manière argumentée, pour démontrer comment ces choix contribuent à l’efficacité et à l’innovation de la solution développée
Référentiel d’activités	Référentiel de compétences	Référentiel d’évaluation	
		Modalités d’évaluation	Critères d’évaluation
Bloc 3. Piloter la mise en production des solutions logicielles et leur évolution			
Modules en lien avec le BLOC 3 : DevOps CI/CD, Docker & Containerisation (Kubernetes) – orchestration, Haute Disponibilité, Evaluation des risques & PCA / PRA, ISO 27001, Introduction Cybersécurité et E-Réputation, Modèle OSI, Urbanisation des SI, Industrialisation des SI, Projet StartUp Agile & DevOps, Projet transverse et préparation de la soutenance			
A3 Organisation du maintien en fonctionnement et des évolutions de solutions logicielles			
A3.1 Intégration des changements de code dans le dépôt de code principal du projet	C3.1 Piloter le système d’intégration continue, dans un environnement Agile et DevOps, en implémentant et en harmonisant les composants et fonctionnalités de la solution logicielle, dans un dépôt de code partagé, en déclenchant des tests automatiques, en s’appuyant sur le système de contrôle de versions (ex. Git, Subversion), sur l’automatisation des builds, par le biais d’outils spécifiques (ex. Jenkins), en respectant les contraintes réglementaires et de sécurité (RGPD), afin de résoudre des erreurs, réduire les risques, et construire une solution fiable dans un temps réduit	C3.1 à C3.6 – Mise en situation professionnelle reconstituée, portant sur le pilotage de la mise en production d’une solution logicielle. L’évaluation prendra la forme d’un dossier écrit et d’une présentation devant un jury composé de 2 professionnels minimum. Le candidat présente son projet pendant 20 mn devant le jury, suivi d’un entretien avec le jury	C.3.1.1 La mise en place du dépôt de code et du système de contrôle de versions (ex. Git, Subversion) permet la gestion des branches (forks) et le suivi de l’historique des modifications du code C.3.1.2 La mise en place de l’Intégration Continue (CI) permet la réalisation des différentes étapes de build et de tests rapidement et sans intervention humaine C.3.1.3 La mise en œuvre de l’Intégration Continue permet la résolution fiable des erreurs dans un temps réduit C.3.1.4 L’ensemble des contraintes de sécurité et réglementaires (ex. RGPD) sont respectées

A3.2 Réalisation de tests automatisés

C3.2 **Organiser l'élaboration du plan de tests itératifs** (tests unitaires, d'intégration, de sécurité), dans un environnement Agile et DevOps, en même temps que le développement, en reprenant l'ensemble des scénarii et des problèmes techniques rencontrés, selon un plan de recensement des dysfonctionnements, de

de 20 mn sur le projet.

C.3.2.1 Le plan de tests itératifs (tests unitaires, d'intégration, de sécurité) couvre l'ensemble des cas d'utilisation, des besoins fonctionnels et des problématiques rencontrées

	correction des bogues et de suivi de la performance, afin de garantir que la base de code fonctionne correctement, selon les attentes documentées des utilisateurs		C.3.2.2 La planification de tests itératifs est organisée pour sa réalisation en parallèle du développement du code et respecte le cahier de recette documenté C.3.2.3 Les tests automatisés permettent l'identification, la correction des bogues et le suivi de la performance des versions du code C.3.2.4 Le plan de tests itératifs permet de vérifier le bon fonctionnement de chaque partie de code, selon les attentes documentées des utilisateurs
A3.3 Surveillance continue des mises à jour logicielles	C3.3 Surveiller les automatisations visant les mises à jour logicielles , dans des conditions les plus proches possibles de la réalité, sur la base du retour d'informations continu sur l'exécution et les performances globales des versions logicielles, en lui intégrant le contrôle humain, en conservant la stabilité de l'infrastructure de développement (sur site ou cloud), en veillant à ce que tous les aspects du pipeline DevOps soient conformes aux normes de sécurité, afin d'orienter la feuille de route du développement logiciel et de guider l'itération des codes générés		C.3.3.1 Les outils et les méthodologies avancés utilisés pour effectuer les actions de surveillance, permettent d'ajuster l'automatisation rapidement, en fonction des besoins de mises à jour logicielles C.3.3.2 La surveillance continue permettent d'orienter la feuille de route du développement logiciel C.3.3.3 Les actions de surveillance permettent de guider l'itération des codes générés et d'assurer une adaptation constante aux besoins des utilisateurs finaux C.3.3.4 Les actions de surveillance entreprises permettent de minimiser les vulnérabilités et de protéger l'intégrité des systèmes et des données
A3.4 Distribution automatique de la version du logiciel aux utilisateurs finaux	C3.4 Piloter le déploiement continu de la solution logicielle , visant la mise en production de l'application, en la rendant disponible aux utilisateurs, de manière automatisée, sans intervention manuelle, sous forme d'une seule procédure de livraison pour toutes les plateformes d'intégration continue et les environnements cibles, permettant des mises à jour fréquentes de l'application, afin de recueillir les retours des utilisateurs, et d'évaluer l'adéquation, la robustesse et la qualité de la solution logicielle proposée		C.3.4.1 La mise en production de l'application permet la distribution automatique de la solution logicielle aux utilisateurs C.3.4.2 La distribution automatique permet la livraison sur l'ensemble des plateformes d'intégration C.3.4.3 Le déploiement continu permet d'obtenir le retour des utilisateurs pour les différentes versions de la solution informatique C.3.4.4 Le déploiement continu permet la mise à jour fréquente de la solution informatique

A3.5 Opérations continues du cycle de vie DevOps	C3.5 Piloter l'optimisation des applications et des environnements , par le biais d'opérations continues, à l'aide d'alertes et des notifications en cas d'anomalie de performance logicielle ou de l'infrastructure, dans le cadre d'un environnement unifié pour le développement et l'intégration, en utilisant une chaîne d'outils intégrés, qui permettent l'automatisation, le contrôle et le travail collaboratif et adaptatif entre les équipes projet, afin de développer rapidement des logiciels et de satisfaire les demandes des clients visant l'amélioration continue des fonctionnalités des logiciels		C.3.5.1 La mise en place d'opérations continues permet des ajustements et des évolutions de la solution selon des cycles courts C.3.5.2 Les alertes et les notifications permettent de détecter efficacement les anomalies de performance logicielle et de l'infrastructure C.3.5.3 L'utilisation d'une chaîne d'outils intégrée permet l'automatisation des contrôles et un travail collaboratif entre les équipes projet C.3.5.4 Les opérations continues permettent de garantir la stabilité et les performances des applications et des environnements
A3.6 Élaboration de la documentation technique	C3.6 Organiser la rédaction de la documentation technique (manuel utilisateur, guide d'utilisation), en décrivant la solution informatique et ses fonctionnalités (User stories), la structure des bases de données et le schéma général de la sécurité, de façon progressive, en intégrant des exemples, en utilisant un langage qui permet de visualiser les composantes de façon standardisée (UML, BPMN), en français et en anglais, accompagnée d'un plan de formation des utilisateurs, en l'adaptant à la diversité du public, et aux personnes présentant un handicap, afin d'améliorer l'expérience utilisateur		C.3.6.1 La documentation technique décrit de façon claire et structurée la solution informatique et ses fonctionnalités, en incluant des <i>user stories</i> , la structure des bases de données, le schéma de sécurité, des exemples pertinents et utilise un langage qui permet de visualiser les composantes de façon standardisée (ex. UML, BPMN) C.3.6.2 La documentation technique est rédigée en français et en anglais (équivalent niveau B2) C.3.6.3 La documentation technique est adaptée aux différentes versions de la solution informatique C.3.6.4 La documentation technique inclut des solutions pour les utilisateurs en situation de handicap (ex. documentation accessible, lisible, audible) C.3.6.5 Le plan de formation des utilisateurs est adapté à la diversité du public, y compris les personnes présentant un handicap (ex. stratégies d'enseignement adaptées à différents groupes d'utilisateurs.)

Référentiel d'activités	Référentiel de compétences	Référentiel d'évaluation	
		Modalités d'évaluation	Critères d'évaluation
Bloc 4. Piloter l'équipe du projet de développement logiciel			
Modules en lien avec le BLOC 4 : Gestion de projets, Management d'équipes, Méthode Agile, Veille technologique, Rédiger & Répondre à un appel d'offre, Projet StartUp Agile & DevOps, Projet transverse et préparation de la soutenance			
A4 Management de l'équipe du projet de développement logiciel			
A4.1 Définition des besoins en compétences de l'équipe du projet de développement logiciel	C4.1 Déterminer les compétences nécessaires à l'accomplissement du projet de développement logiciel, ainsi que les interactions prévues avec les autres équipes, en concordance avec les objectifs établis pour la solution proposée au client, en accord avec le cycle de vie du développement du logiciel, en définissant les modalités afin de constituer une équipe projet performante	C4.1 à C4.6 – Mise en situation professionnelle reconstituée, portant sur le pilotage d'une équipe de projet de développement logiciel. L'évaluation prendra la forme d'un dossier écrit et d'une présentation devant un jury composé de 2 professionnels minimum. Le candidat présente son projet pendant 20 mn devant le jury, suivi d'un entretien avec le jury de 20 mn sur le projet .	C4.1.1 Les compétences techniques et non techniques nécessaires sont identifiées par rapport aux objectifs et tâches figurant dans le plan du projet de développement logiciel C4.1.2 Les rôles et responsabilités de chaque équipe sont détaillées C4.1.3 Les interactions de chaque profil au sein de l'équipe et avec les autres équipes sont reliées aux différentes étapes du cycle de vie du développement du logiciel (ex. RACI) C4.1.4 Les modalités d'accès et d'intégration des personnes en situation de handicap sont clairement définies, et les modalités de travail pour les membres de l'équipe en situation de handicap sont adaptées
A4.2 Création de l'équipe du projet de développement logiciel	C4.2 Constituer l'équipe du projet de développement logiciel, par le biais de la formation interne et du recrutement, en collaboration avec l'équipe RH de l'entreprise, en identifiant les missions et les responsabilités associées à la solution de développement logiciel, afin d'atteindre les objectifs du projet, fixés dans le cahier de charges		C4.2.1 Le plan de constitution de l'équipe, par recrutement et par formation interne, en collaboration avec les équipes RH, permet de couvrir l'ensemble des besoins identifiés en termes de compétences, d'expériences et d'aptitudes C4.2.2 Les descriptifs de poste permettent d'identifier les missions et responsabilités de chacun, et l'ensemble des tâches et responsabilités sont allouées C4.2.3 La stratégie d'intégration et d'adaptation mis en œuvre pour les personnes en situation de handicap, incluant des aménagements spécifiques et une sensibilisation de l'équipe, leur permet de réaliser le travail dans des conditions satisfaisantes

A4.3 Gestion opérationnelle de l'équipe du projet de développement logiciel	C4.3 Coordonner l'activité de l'équipe du projet de développement logiciel , par la gestion de l'intégration des nouveaux membres, en allouant les tâches et responsabilités et en veillant à l'équilibre des charges de travail, en français et en anglais selon les besoins, afin de garantir la productivité de l'équipe		C.4.3.1 Le plan d'intégration des nouveaux membres est défini et permet une insertion efficace au sein de l'équipe C.4.3.2 Le plan d'allocation des tâches et de gestion des ressources humaines permet une répartition équilibrée et efficace du travail C.4.3.3 Le pilotage de l'équipe est réalisé en français ou en anglais, selon les besoins (équivalent niveau B2)
A4.4 Animation de l'équipe du projet de développement logiciel	C4.4 Accompagner les membres de l'équipe du projet de développement logiciel , en mettant en place des stratégies pour fluidiser la communication interne dans un contexte agile, les processus de développement et pour la résolution de problèmes, par le biais des échanges et des réunions spécifiques, en utilisant une plateforme collaborative inclusive et des outils numériques afin de faciliter la collaboration et la productivité de l'équipe		C.4.4.1 La stratégie de communication au sein de l'équipe du projet de développement se réalise par des échanges réguliers en présentiel ou à distance, à travers des outils collaboratifs C.4.4.2 La résolution des problèmes au sein de l'équipe du projet de développement logiciel par des réunions spécifiques permet d'améliorer le processus de travail et stimule la productivité C.4.4.3 L'accompagnement des membres de l'équipe du projet de développement logiciel (réunions, débriefings, etc.) facilitent l'engagement et l'avancée du projet C.4.4.4 L'utilisation de plateformes, outils et modes de communication accessibles à tous les membres de l'équipe, y compris ceux en situation de handicap, permet de garantir une participation égale au sein d l'équipe
A4.5 Plan de formation de l'équipe du projet de développement logiciel	C4.5 Planifier la formation des membres de l'équipe du projet de développement logiciel , en mettant en place des actions de développement des compétences, afin d'acquérir, de maintenir et d'actualiser les compétences de l'équipe sur les avancées technologiques et méthodologiques, et afin de maintenir la performance de l'équipe projet et d'obtenir des résultats optimaux dans l'activité		C.4.5.1 Les actions de formation individuelle et collective mises en place permettent d'acquérir, de maintenir et d'actualiser les compétences techniques et non techniques nécessaires au projet de développement logiciel C.4.5.2 Les actions de formation des membres des membres de l'équipe du projet de développement logiciel prennent des formes variées (ex. en présentiel, e-learning, etc.) et favorisent la mise en situation C.4.5.3 Les actions de formation présentent les aménagements nécessaires pour les personnes en situation de handicap, garantissant leur participation active C.4.5.4 Le suivi de la performance collective et individuelle permet de s'assurer de l'efficacité des actions de formation

A4.6 Suivi de la performance de l'équipe du projet de développement logiciel	C4.6 Évaluer la performance de l'équipe du projet de développement logiciel, en établissant un référentiel de performance pour l'équipe et des référentiels de performance individuels, en analysant la performance collective et les performances individuels au regard de ce référentiel, en réalisant des feedbacks réguliers et constructifs à double sens, en assurant des opportunités de développement au sein de l'équipe, et en établissant des plans de carrière en collaboration avec le service RH, afin d'optimiser la performance de l'équipe tout en assurant le bien-être des employés et en maintenant un environnement de travail positif et inclusif		C.4.6.1 Les référentiels de performance sont adaptés au profils et niveau d'expérience des individus, et ils permettent de mesurer l'ensemble des dimensions nécessaires à la réussite du projet C.4.6.2 L'analyse de la performance s'appuie sur des indicateurs objectifs, et elle intègre le suivi du bien-être au travail C.4.6.3 Les feedbacks sont réguliers, constructifs, à double sens et structurés C.4.6.4 Les plans de développement sont cohérents avec les plans de carrière, et ils assurent des opportunités de développement au sein de l'équipe à chacun C.4.6.5 Les référentiels, la mesure de la performance, les plans de développements et les plans de carrière intègrent les aménagements nécessaires aux individus en situation de handicap
---	---	--	---

La totalité des blocs doit être validée pour obtenir la certification. Il ne peut pas y avoir de validation partielle du diplôme.

Compétences évaluables en milieu professionnel

Capacité à émettre un avis constructif sur un projet, des informations.

Capacité à prendre du recul.

Capacité à faire émerger des idées nouvelles

Capacité à penser collectif, à renforcer la cohésion. "Seul on va plus vite, ensemble on va plus loin"

Capacité à s'intégrer au sein d'une équipe, à collaborer en bonne intelligence et fédérer l'équipe pour atteindre un objectif commun

Capacité à savoir prendre du recul pour analyser ses émotions et celles des autres pour ne pas les subir.

Capacité à se mettre au service de l'entreprise, de ses collègues mais aussi capacité à se mettre au service du client.

Capacité à argumenter et contre argumenter, à vendre ses idées, à convaincre.

Capacité à écouter, rebondir et interagir.

Capacité à s'adapter à des personnes, à des situations changeantes, en évolution.

Capacité à être proactif, audacieux, force de proposition, aller vers le Maître d'Apprentissage.

Capacité à s'exprimer clairement, à délivrer un message précis, concis, structuré et clair.

Capacité à être autonome (agir seul)

Capacité à se faire confiance et à avoir confiance en l'autre.

Capacité à voir le chemin pour atteindre l'objectif.

Capacité à prioriser, à identifier les choses essentielles des autres qui le sont moins.