

TITRE EXPERT EN DEVELOPPEMENT LOGICIEL

5 - REFERENTIELS

REFERENTIEL D'ACTIVITES décrit les situations de travail et les activités exercées, les métiers ou emplois visés	REFERENTIEL DE COMPETENCES identifie les compétences et les connaissances, y compris transversales, qui découlent du référentiel d'activités	REFERENTIEL D'ÉVALUATION définit les critères et les modalités d'évaluation des acquis	
		MODALITÉS D'ÉVALUATION	CRITÈRES D'ÉVALUATION
BLOC 1 : CADRER UN PROJET DE DEVELOPPEMENT D'APPLICATIONS LOGICIELLES			
A1.1. Evaluation de l'opportunité et de la faisabilité du projet <ul style="list-style-type: none"> - Recueil et analyse des besoins clients. - Analyse des infrastructures, de l'architecture. - Identification des risques et des leviers. - Analyse des contraintes techniques, fonctionnelles et réglementaires. - Etude des avantages et bénéfices de la solution pour le client. - Identification des moyens techniques et technologiques. - Identification des futurs utilisateurs. - Elaboration de scénarios. 	C1.1.1 Analyser la demande et les données recueillies auprès du commanditaire en menant un entretien d'explicitation du besoin, des attentes, des exigences et une analyse de l'existant, afin de s'approprier le contexte, les enjeux, les objectifs du projet, cerner la problématique du client et faire émerger des pistes de solutions.	Simulation de situation de travail sous la forme de projet + présentation orale : Le candidat présente le cadrage d'un projet de logiciel développé au cours de son parcours de formation. La restitution est réalisée par l'élaboration d'un dossier technique et d'une présentation orale individuelle devant les membres du jury. Il est attendu : <ul style="list-style-type: none"> - l'analyse de la problématique et la présentation des acteurs impliqués 	<ul style="list-style-type: none"> - Les besoins, les attentes du client sont identifiés, recensés et arbitrés par le candidat. - L'analyse de la problématique est approfondie. - Une note de cadrage est rédigée. - Les objectifs de la solution sont définis.
	C1.1.2. Identifier les différents acteurs du projet (commanditaire, utilisateurs, parties prenantes) et leurs rôles pour cadrer l'environnement et le périmètre du projet.		<ul style="list-style-type: none"> - Les acteurs du projet sont listés et identifiés (utilisateur, développeur, architecte, administrateur, client, actionnaires, concepteurs d'autres systèmes, etc.) - Les futurs utilisateurs sont caractérisés.

ELEMENTS COMPLEMENTAIRES RELATIFS A LA DEMANDE

	<p>C1.1.3. Evaluer l'opportunité et la faisabilité du projet de développement en analysant l'environnement technique et fonctionnel, les contraintes et le budget du client pour décider de son lancement et déterminer les moyens nécessaires à sa réalisation.</p>	<ul style="list-style-type: none"> - la méthodologie de recherche des informations et les sources consultées - l'analyse des forces et des faiblesses du projet - le diagnostic des infrastructures existantes - l'analyse critique de la faisabilité en lien avec les contraintes identifiées - la cartographie des risques liés au projet - l'étude comparative des solutions techniques envisagées - l'identification des ressources matérielles/techniques nécessaires - la liste des spécifications et des fonctionnalités - l'estimation des coûts - les schémas de l'architecture logicielle proposée - le candidat doit convaincre le jury de la qualité et de la viabilité du concept qu'il propose. 	<ul style="list-style-type: none"> - Une analyse de type SWOT est réalisée. - Le candidat formule un avis critique sur l'opportunité et la faisabilité du projet. Il analyse les points forts et faiblesses du projet. - Une étude technique est réalisée : les langages informatiques, les caractéristiques des bases de données, l'architecture existante, les technologies utilisées, etc. - La démarche d'audit est documentée. - Les contraintes techniques et financières sont identifiées : hébergement, système d'exploitation, volume de données, nbre d'utilisateur, délais, ressources financières, techniques et humaines, etc. - Les applications et logiciels existants sont appréhendés et leurs fonctionnalités explicitées.
<p>A1.2. Identification des risques et des menaces potentiels</p> <ul style="list-style-type: none"> - Anticipation des risques. - Priorisation des risques. - Définition des indicateurs de contrôle. - Création d'un référentiel d'évaluation de risques et de suivi des incidents. - Ajustement du projet au regard des risques identifiés. 	<p>C1.2. Cartographier les risques techniques et fonctionnels associés au projet et les facteurs de dégradations en mettant en place un référentiel et des indicateurs de contrôle afin d'analyser leur criticité et limiter les impacts sur la performance du développement.</p>		<ul style="list-style-type: none"> - Les risques techniques et fonctionnels sont cartographiés et priorisés dans un référentiel : perte de données, interruption du système, juridiques, sécurité, etc. - Les facteurs de dégradation des performances du logiciel sont identifiés. - Le modèle d'estimation des risques est maîtrisé et expliqué. - Leur criticité est évaluée : nature de la menace, vulnérabilité, actif sensible. - Des mesures correctives sont préconisées.
<p>A1.3. Sélection de la solution technique</p> <ul style="list-style-type: none"> - Évaluation des solutions. - Choix des technologies appropriées. - Définition des indicateurs de performances. 	<p>C1.3. Recenser les différentes solutions techniques, leurs environnements ainsi que le langage de programmation afin d'identifier les plus adaptées au projet de développement et aux performances attendues.</p>		<ul style="list-style-type: none"> - Une analyse comparative des solutions technique est réalisée. - Les choix retenus sont justifiés et adaptés à la réalisation du projet. - Le candidat analyse les contraintes et les avantages des différentes solutions en termes d'environnements systèmes, réseaux, etc. - Les indicateurs de performance sont déterminés : tps de réponse, robustesse du système, etc. - Leur choix est justifié et cohérent aux besoins.

ELEMENTS COMPLEMENTAIRES RELATIFS A LA DEMANDE

<p>A1.4. Caractérisation fonctionnelle de la solution logicielle</p> <ul style="list-style-type: none"> - Réalisation d'un diagramme de fonctionnalités ou d'un cahier des charges fonctionnel. - Explicitation des scénarios d'interaction des utilisateurs. - Identification de fonctionnalités évolutives. 	<p>C1.4. Formaliser les caractéristiques, les fonctionnalités de la solution logicielle dans un plan de conception technique et fonctionnel en tenant compte des exigences et des contraintes du projet afin de proposer une solution logicielle répondant aux exigences du client.</p>		<ul style="list-style-type: none"> - Les fonctions sont recensées, caractérisées, ordonnées et hiérarchisées (fonctions principales, secondaires, complémentaires). - L'outil d'analyse fonctionnelle est explicité. - La couverture technique des besoins fonctionnels est argumentée et justifiée. - L'expérience utilisateur est prise en compte.
<p>A1.5. Estimation du coût du projet</p> <ul style="list-style-type: none"> - Estimation du budget prévisionnel. - Identification des coûts fixes et variables. - Formalisation du budget alloué. - Validation auprès de la direction. 	<p>C1.5. Estimer le coût associé au projet en se basant sur l'étude technique et en intégrant les équilibres stratégiques et économiques pour assurer la rentabilité de la solution envisagée et proposer des solutions alternatives, le cas échéant.</p>		<ul style="list-style-type: none"> - Le budget est élaboré en utilisant une méthode d'estimation des coûts. - Le coût du développement est estimé et détaillé. - Le ratio coût/développement est pertinent.
<p>A1.6. Modélisation de l'architecture du logiciel</p> <ul style="list-style-type: none"> - Sélection du ou des modèles d'architectures. - Schématisation du système, des interrelations et interactions. - Réalisation de diagrammes d'architecture. 	<p>C1.6. Modéliser une architecture logicielle à partir du scénario élaboré en respectant les spécifications fonctionnelles attendues et les exigences de sécurité, afin de préparer la phase de développement et faciliter l'évolution, le déploiement et la maintenance du logiciel.</p>		<ul style="list-style-type: none"> - L'architecture est schématisée et légendée (signification des formes, flèches, couleurs, positions, etc.). - Elle répond aux exigences des parties prenantes et aux contraintes de production. - Elle est adaptée au système et à l'infrastructure. - Le choix de la méthode de modélisation et du formalisme est justifié (UML, etc.). - Les interactions avec les systèmes informatiques sont explicitées. - Le candidat produit des diagrammes d'architecture. - L'architecture proposée est maintenable et extensible. - L'architecture logicielle permet une intégration de composants logiciels.
<p>A1.7. Réalisation d'une veille</p> <ul style="list-style-type: none"> - Mise en œuvre de dispositifs de veille. - Analyse des évolutions et les nouveaux usages des technologies. - Développement de l'expertise. 	<p>C1.7. Réaliser une veille technique, technologique et réglementaire en s'informant constamment des évolutions et des nouveaux usages pour suivre ou anticiper les innovations du secteur.</p>		<ul style="list-style-type: none"> - Une synthèse des sources d'information est réalisée. - Le candidat explicite la stratégie et les outils de veille mis en œuvre. - La démarche de veille et les outils de veille sont documentés.

ELEMENTS COMPLEMENTAIRES RELATIFS A LA DEMANDE

<p>A1.8. Gestion des interactions professionnelles</p> <ul style="list-style-type: none">- Vulgarisation du vocabulaire technique.- Elaboration de l'argumentaire.- Utilisation des supports de communication (papier ou numérique).- Traitement des objections.- Validation de la faisabilité du projet.	<p>C1.8. Présenter les décisions et les axes de solutions préconisées, en français et en anglais, auprès des différentes parties prenantes en développant un argumentaire adapté afin d'obtenir leur adhésion et validation.</p>		<ul style="list-style-type: none">- Le candidat expose avec fluidité et clarté le cadre du projet et les solutions techniques préconisées.- Les choix sont argumentés avec aisance en français et en anglais.- Le vocabulaire est professionnel et les termes techniques sont appropriés.- Le candidat vulgarise son discours pour s'adapter à son auditoire- Les objections sont prises en compte et traitées par le candidat.- Le candidat est convaincant : implication dans sa présentation, pertinence des arguments, etc.
--	--	--	--

ELEMENTS COMPLEMENTAIRES RELATIFS A LA DEMANDE

BLOC 2 : CONCEVOIR ET DEVELOPPER DES APPLICATIONS LOGICIELLES

<p>A2.1. Prototypage du logiciel</p> <ul style="list-style-type: none"> - Exploration des scénarios d'utilisation. - Test de la validité de la conception. - Recueil des remarques des utilisateurs. - Traçage des évolutions du prototype. 	<p>C2.1. Concevoir un prototype de l'application logicielle en tenant compte des spécificités ergonomiques, des équipements ciblés (web, mobile...) et des fonctionnalités attendues, afin d'explorer de façon interactive les scénarios d'utilisation et valider les orientations.</p>	<p>Simulation de situation de travail sous la forme de projet + présentation orale :</p> <p>Le candidat présente un logiciel développé au cours de son parcours de formation. Il doit avoir assuré son développement en autonomie. La restitution est réalisée par l'élaboration d'un dossier technique et d'une présentation orale individuelle devant les membres du jury.</p> <p>Il est attendu :</p> <ul style="list-style-type: none"> - un logiciel fonctionnel, viable, fiable. - une présentation des prototypes réalisés - l'utilisation de Framework et des paradigmes de développement - une architecture logicielle structurée permettant la maintenabilité du logiciel - le protocole de sécurisation - les protocoles et cahiers de recette - le protocole d'intégration continue - l'historique des différentes versions - les critères de qualités et de performance établis 	<ul style="list-style-type: none"> - La phase de prototypage réalisée est documentée. - Le prototype est fonctionnel et cohérent avec les besoins identifiés. - Il met en œuvre un ensemble cohérent de fonctionnalités principales du logiciel et les scénarios utilisateur. - Les composants de l'interface (fenêtres, boutons, menus, etc.) sont présents et fonctionnels. - La maquette prend en compte les exigences de sécurité. - Les évolutions du prototype sont tracées.
<p>A2.2. Programmation du logiciel</p> <ul style="list-style-type: none"> - Rédaction du code source. - Manipulation des environnements de développement. - Mise en œuvre de bibliothèques logicielles, de Framework et patrons de conception. - Mise en œuvre de la sécurisation du code. 	<p>C2.2. Développer le logiciel en veillant à l'évolutivité et à sécurisation du code source dans le respect des spécifications techniques et fonctionnelles définies, pour garantir une exécution conforme et satisfaire le client.</p>	<ul style="list-style-type: none"> - Les bonnes pratiques de développement sont respectées. - Le candidat utilise des Framework, des bibliothèques, des librairies, des API à bon escient. - Les langages de programmation sont adaptés au domaine d'application. - L'environnement de développement est explicite (éditeur de code, compilateur, etc.). - Le développement du code source est propre. - Le candidat utilise des balises de commentaires au sein de son code. - Le candidat utilise la méthode OWASP et la technique du Fuzzing pour la sécurisation du code. 	

ELEMENTS COMPLEMENTAIRES RELATIFS A LA DEMANDE

<p>A2.3. Utilisation de la méthode d'intégration continue</p> <ul style="list-style-type: none"> - Configuration du serveur d'intégration. - Surveillance des régressions. - Corrections des bogues. - Implémentation de l'ensemble des fonctionnalités du logiciel. - Enregistrement de l'historique des modifications et des versions. 	<p>C2.3. Concevoir le système d'intégration continue dans le cycle de développement du logiciel en fusionnant les codes sources, dans le but de tester régulièrement les blocs de code et effectuer des modifications incrémentielles permettant d'assurer un développement efficient et réduire les risques.</p>	<ul style="list-style-type: none"> - la présentation de la documentation technique du logiciel 	<ul style="list-style-type: none"> - Le protocole d'intégration continue est explicité. - Les incidents de sécurité sont détectés, qualifier et traiter. - Les bogues de codes sont détectés, qualifiés et traités. - Une matrice de traçabilité est mise en œuvre. - Le candidat utilise un système de gestion de versions.
<p>A2.4. Elaboration d'un cahier de recette</p> <ul style="list-style-type: none"> - Planification du cahier de recette. - Rédaction des scénarii de recette. - Choix des outils. - Recensement des problèmes techniques rencontrés. - Réalisation d'un plan de correction des bogues. - Mise en œuvre des outils de suivi de performance et de qualité. 	<p>C2.4.1. Elaborer un plan de tests itératifs permettant d'identifier les comportements problématiques et de détecter les anomalies afin de garantir la qualité, la fiabilité, la non-régression et sécuriser le déploiement du logiciel.</p> <p>C2.4.2. Prendre en compte les retours et demandes des utilisateurs en recueillant les données et mesures liées à son utilisation et en analysant les tableaux de bord associés, pour exécuter un développement qualitatif.</p>		<ul style="list-style-type: none"> - La recette est planifiée de bout en bout du projet. - Le cahier de recette est exhaustif et reprend l'ensemble des cas d'utilisation des processus, reprises et interfaces du logiciel. - Les tests fonctionnels, structurels et de sécurité exécutés sont conformes au plan définis. - Les protocoles de mise en œuvre des tests unitaire, d'intégration, de validation et de non-régressions sont documentés. - Une analyse des points d'amélioration est réalisée pour chaque test en échec. - Le rapport d'exécution de tests est documenté. - Les corrections et des améliorations sont proposées. - Le candidat propose une automatisation des tests.
<p>A2.5. Réalisation du déploiement du logiciel</p> <ul style="list-style-type: none"> - Configuration du déploiement. - Traitement par pans. - Vérification des fonctionnalités. - Accompagnement des utilisateurs dans l'appropriation. 	<p>C2.5. Déployer le logiciel à chaque modification de code et de façon progressive en vérifiant la performance fonctionnelle et technique auprès des utilisateurs afin de présenter une solution stable et conforme à l'attendu.</p>		<ul style="list-style-type: none"> - Le candidat évalue les impacts, avantages et risques d'un déploiement continu. - Les étapes de déploiement sont définies. - Un run blanc est planifié. - La procédure se déclenche en cas de détection de nouvelle contribution. - Le candidat utilise des techniques de types feating flipping.

ELEMENTS COMPLEMENTAIRES RELATIFS A LA DEMANDE

<p>A2.6. Rédaction de la documentation technique</p> <ul style="list-style-type: none"> - Description de la modélisation, de son architecture, code source, déploiement. 	<p>C2.6. Rédiger la documentation technique d'exploitation du logiciel détaillant son fonctionnement afin d'assurer une traçabilité pour le suivi des équipes et des futures évolutions du logiciel.</p>		<ul style="list-style-type: none"> - Le document technique est exhaustif et rédigé avec clarté. - La présentation est claire et la compréhension aisée. - La documentation décrit les choix opérés : technologies, langages, processus de génération, etc.
<p>A2.7. Accompagnement des utilisateurs dans la prise en main du logiciel</p> <ul style="list-style-type: none"> - Formation des utilisateurs - Rédaction de guides d'utilisation - Conduite du changement - Facilitation de l'adhésion des utilisateurs 	<p>C2.7. Gérer la conception de guides/manuels, de tutoriels ou de la formation à l'utilisation du logiciel pour accompagner la bonne prise en main des différentes fonctionnalités.</p>		<ul style="list-style-type: none"> - Les supports de prise en main sont rédigés avec clarté. - Les informations sont utiles et correspondantes aux préoccupations de l'utilisateurs. - La présentation est agréable et fluide à lire.

ELEMENTS COMPLEMENTAIRES RELATIFS A LA DEMANDE

BLOC 3 : COORDONNER ET PILOTER UN PROJET DE DEVELOPPEMENT D'APPLICATIONS LOGICIELLES

<p>A3.1. Organisation de l'exécution</p> <ul style="list-style-type: none"> - Elaboration de la feuille de route et du calendrier de réalisation (retroplanning). - Choix de la méthodologie de conduite de projet : Cycle en V, Agile, Scrum, Extrem Programing, etc. - Identification des ressources : humaines, financières et matérielles. - Définition des rôles et responsabilités des acteurs du projet. - Listing et affectation des tâches. 	<p>C3.1. Organiser le cadre méthodologique du projet, la répartition et l'ordonnancement des activités, le planning prévisionnel de réalisation et les ressources nécessaires à son exécution afin de clarifier les responsabilités des différents acteurs et assurer une bonne coordination.</p>	<p>Etude de cas</p> <p>L'étude de cas individuelle est réalisée à partir d'un cas d'entreprise réel ou fictif et de documents fournis. Elle est complétée par l'élaboration de livrables et d'une présentation orale individuelle devant les membres du jury.</p> <p>Il est attendu que :</p> <ul style="list-style-type: none"> - le candidat réalise le planning détaillé du projet en utilisant des outils d'élaboration de planning. - le candidat présente la manière dont il a positionné les points de suivi avec le client. 	<ul style="list-style-type: none"> - Une méthode de gestion de projet (agile, Lean, etc.) est choisie. - Le planning est construit au moyen d'outils adaptés (ex : diagramme de Gantt, ...). - Le planning est réaliste, il tient compte des délais et aléas. - Le candidat identifie toutes les étapes clés du projet et séquence le développement en différent sprint. - Les ressources nécessaires à la réalisation du projet sont identifiées. - Le candidat identifie les rôles, activités et compétences des différents professionnels devant intervenir sur le projet (matrice RACI, etc.).
<p>A3.2. Supervision de la production</p> <ul style="list-style-type: none"> - Définition d'outils de suivi. - Identification des indicateurs de suivi de projet. - Suivi du budget. - Suivi de la production de l'équipe projet. - Suivi des dérives et ajustement des écarts. 	<p>C3.2. Superviser la production en assurant un suivi régulier de l'avancée, en apportant des conseils et indications pour guider la production, afin de garantir l'avancement du projet dans le respect des délais, de la qualité et des coûts.</p>	<ul style="list-style-type: none"> - Le candidat détaille des ressources nécessaires - le candidat identifie les compétences à mobiliser - le candidat affecte les différentes missions à réaliser - le candidat met des avis et recommandations argumentés d'amélioration. - au cours de la soutenance orale le candidat présente la manière dont il qualifie et analyse sa posture managériale, ainsi qu'une analyse réflexive de la gestion et de l'animation d'une équipe. Celle-ci est argumentée et permet d'identifier la prise de recul du candidat face à des situations rencontrées 	<ul style="list-style-type: none"> - Les jalons de suivi et de reporting (réunions, mails, notes...) sont indiqués. - Les indicateurs de suivi de projet sont mesurables : coûts, délai, etc. - Les mesures de l'état d'avancement du projet sont explicitées. - Les contrôles de la qualité des développements sont documentés. - Le candidat intègre les notions de coût, délai, qualité et risques dans son suivi. - Le candidat utilise des tableaux de bord intégrant le suivi des coûts, des délais, des RH, etc. - Le candidat met en place des revues de code. - Le candidat utilise des outils collaboratifs de partage de ressources.
<p>A3.3. Management d'une équipe</p> <ul style="list-style-type: none"> - Maintien de la motivation et de l'implication des équipes. 	<p>C3.3. Animer l'équipe tout au long du projet en utilisant les différentes techniques de communication et managériales en veillant au respect du plan établi pour favoriser le</p>	<p>est argumentée et permet d'identifier la prise de recul du candidat face à des situations rencontrées</p>	<ul style="list-style-type: none"> - Le style managérial est identifié et décrit : capacité d'écoute, reformulation, empathie, réponses aux objections.

ELEMENTS COMPLEMENTAIRES RELATIFS A LA DEMANDE

<ul style="list-style-type: none"> - Gestion des conflits et des personnalités (écoute, empathie, directivité, ...). - Conduite de réunions, groupes d'échanges ou entretiens avec l'équipe. 	<p>bon fonctionnement de l'équipe et la qualité du livrable.</p>		<ul style="list-style-type: none"> - Le candidat repère des points négatifs de la posture qui nuisent à la relation managériale. - Le candidat émet des conseils opérationnels sur la posture managériale. - Les recommandations sont réalistes et réalisables dans leur mise en œuvre. - Le candidat maîtrise les grands principes et techniques de gestion managériale et d'animation d'équipe (empathie, écoute, bienveillance, leadership)
<p>A3.4. Réalisation de reporting</p> <ul style="list-style-type: none"> - Réalisation de comptes-rendus. - Validation avec le commanditaire des étapes du développement depuis la genèse jusqu'au déploiement final. 	<p>C3.4. Effectuer des reporting d'activités sur l'état d'avancement auprès des différentes parties prenantes en présentant les évolutions et améliorations du projet pour favoriser l'adhésion et permettre la validation des avancées de production.</p>		<ul style="list-style-type: none"> - Un process de reporting est mis en place. - Les comptes-rendus sont exhaustifs. - Les points de validation sont organisés de manière à faciliter les prises de décision du client. - La satisfaction du client est mesurée.
<p>A3.5. Maintien et développement de la qualité du logiciel</p> <ul style="list-style-type: none"> - Définition les indicateurs de qualité. - Proposition d'axes d'amélioration/de perfectionnement. - Maintien du cycle de vie du logiciel. - Traçabilité des actions de maintenance et d'évolution. 	<p>C3.5. Etablir une démarche d'amélioration continue en analysant les indicateurs de performance et en traitant les retours des utilisateurs pour garantir la satisfaction du client et créer de nouvelles opportunités.</p>		<ul style="list-style-type: none"> - Le candidat justifie la méthode de gestion de projet utilisée (Merise, RUP, etc.). - Les critères de qualité et de performance sont décrits. Ils sont adaptés au projet. - Les exigences en termes de performance et de qualité et d'évolutivité sont formalisées : fréquence des bogues, typologie du bogues, risque, durée de correction, etc. - Une matrice de gestion des retours client est mise en place. - Un plan de maintenance est élaboré. - Des pistes d'optimisation du logiciel pertinentes sont proposées.