

Référentiel de certification et d'évaluation

Exploiter les outils de développement de la chaîne DevOps

Référentiels d'activités ,de compétences et d'évaluation

Prérequis : Justifier d'une expérience professionnelle d'un an minimum en tant que développeur.se.

REFERENTIEL D'ACTIVITES <i>Décrit les situations de travail et les activités exercées, les métiers ou emplois visés</i>	REFERENTIEL DE COMPETENCES <i>Identifie les compétences et les connaissances, y compris transversales, qui découlent du référentiel d'activités</i>	REFERENTIEL D'EVALUATION <i>Définit les critères et les modalités d'évaluation des acquis</i>	
		MODALITES D'EVALUATION	CRITÈRES D'ÉVALUATION
A1. Amélioration du processus de développement d'applications à l'aide des principes de l'intégration continue - Configuration de l'outil de versionnement* du code source	C1. Assurer le versionnement d'un code source d'une application organisée en fonctionnalités et lots à l'aide d'un logiciel de contrôle de version de manière à garantir la fiabilité du code source dans un environnement multi-contributeurs	E1. Projet professionnel : Le/la candidat(e) doit : - versionner le code source d'une application en respectant les bonnes pratiques de collaboration et de partage de sources.	- Une méthode de versionnement est choisie et appliquée (<i>git flow</i> , <i>Github flow</i> , ...), - les modifications apportées sur le code source lors du versionnement sont rédigées de manière claire et consistante.

<ul style="list-style-type: none"> - Versionnement du code source avancé <ul style="list-style-type: none"> - branches* - pull requests* - etc. - Partage du code source versionné en équipe sur un serveur prévu à cet effet - Réalisation de tests avancés (unitaires, fonctionnels, <i>smoke*</i>...) - Choix des outils d'analyses statiques du code source - Mise en oeuvre d'outils d'analyses statiques du code source - Choix de l'outil d'intégration continue - Production des fichiers de configuration à destination du serveur d'intégration continue 			Indicateur(s) : <ul style="list-style-type: none"> - chaque fonctionnalité est développée dans une branche différentes
	C2. Contrôler l'exécution du code source à l'aide de tests et d'outils d'analyses statiques* du code source afin de minimiser le risque d'erreur dans un contexte de livraison continue	E1. Projet professionnel : Dans le cadre d'une mise en situation professionnelle, le/la candidat(e) doit : <ul style="list-style-type: none"> - Sélectionner et mettre en oeuvre des outils d'analyses statiques du code source, - réaliser l'ensemble des tests nécessaires, - détecter/identifier les erreurs, le cas échéant. 	<ul style="list-style-type: none"> - Le choix des outils est argumenté et l'intérêt de leur utilisation et/ou l'adaptation de ses outils à la situation est démontré, - les tests sont correctement réalisés (unitaires, fonctionnels, intégration, smoke, etc), - la couverture du code par les tests est correctement évaluée. Indicateur(s) : <ul style="list-style-type: none"> - les tests sont bien exécutés, - le rapport du test est correctement interprété
	C3. Automatiser les phases de tests unitaires et d'analyses statiques du code source lors du partage des sources à l'aide d'un outil d'intégration continue* de manière à prévenir les erreurs potentielles	E1. Projet professionnel : Dans le cadre d'une mise en situation professionnelle, le/la candidat(e) doit : <ul style="list-style-type: none"> - Sélectionner un outil d'intégration continue, - Produire un fichier de configuration définissant 	<ul style="list-style-type: none"> - Un retour d'expérience justifiant le choix et l'utilisation de l'outil d'intégration continue est proposé, - un fichier de configuration permettant que les builds et tests s'exécute correctement est produit correctement, Indicateur(s) : <ul style="list-style-type: none"> - lors d'un <i>commit*</i>, les tests

- Configuration de l'automatisation de la communication avec le serveur d'intégration continue et le serveur de dépôts		l'intégration continue, - Synchroniser le logiciel de versionnement et l'exécution des tests.	s'exécutent correctement, - le rapport des tests est bien réceptionné.
A2. Mise en oeuvre des conditions préalables à la livraison continue - Définition des étapes de la chaîne de livraison continue - Configuration des étapes automatisées de la chaîne de livraison continue - Choix de l'outil de conteneurs - Configuration de l'outil de conteneurs pour l'environnement de développement - Réalisation d'une architecture en micro-services - Orchestration simple des conteneurs	C4. Concevoir un processus de livraison continue à l'aide d'outils d'automatisation de manière à l'intégrer au processus de développement	E1. Projet professionnel : Dans le cadre d'une mise en situation professionnelle, le/la candidat(e) doit : - Identifier les étapes de la chaîne de livraison continue - Proposer l'ajout de tests complémentaires - Automatiser la gestion des artefacts et de la livraisons en environnements de <i>staging</i> *.	- Toutes les étapes de la chaîne de livraison continue sont clairement identifiées, - l'identification des étapes permet la définition et l'exécution des tests complémentaire (d'intégration et/ou de performance et/ou de sécurité.... etc). Indicateur(s) : - l'enchaînement des étapes est automatisable, - le processus de livraison continue permet l'automatisation de la livraison de l'application en environnement de <i>staging</i> .
	C5. Développer l'architecture d'une application en micro-services à l'aide d'outils et de bibliothèques logicielles adaptées afin de réduire la complexité globale du système	E1. Projet professionnel : Dans le cadre d'une mise en situation professionnelle, le/la candidat(e) doit :	- les outils et bibliothèques logicielles nécessaires à la

		<ul style="list-style-type: none"> - Sélection les outils et les bibliothèques logicielles, - créer l'architecture d'une application en micro-services ou améliorer l'architecture d'une application existante en micro-services 	<p>réalisation de son architecture sont clairement identifiés,</p> <ul style="list-style-type: none"> - les outils et bibliothèques sélectionnés sont adaptés à l'application et aux besoins techniques, - l'architecture de l'application est développé en respectant les bonnes pratiques du domaine en matière de micro-services. <p>Indicateur(s) :</p> <ul style="list-style-type: none"> - l'application est fonctionnelle et son architecture est organisée en micro-services.
<p>A3. Conception et mise en oeuvre d'un système de veille technologique pour aider à la prise de décision</p> <ul style="list-style-type: none"> - Sélection des sources d'information pertinentes et état de l'art en français et anglais - Collecte des données/informations liées aux problématiques rencontrées par l'organisation (évolutions ou émergences de nouvelles techniques...) 	<p>C6. Concevoir un système de veille technologique permettant la collecte, la classification, l'analyse et la diffusion de l'information aux différents acteurs de l'organisation afin d'améliorer la prise de décisions techniques</p>	<p>E1. Projet professionnel :</p> <p>Le/la candidat(e) doit :</p> <ul style="list-style-type: none"> - réaliser un état de l'art en explicitant le choix de leur sources et en proposer une analyse. 	<ul style="list-style-type: none"> - Un état de l'art des méthodologies et outils existants est dressé sur la thématique donnée. - Les sources d'information sont identifiées et leur fiabilité évaluée. - Une analyse pertinente de cet état de l'art est exposée. - Une conclusion technique en fonction de cet état de l'art et de l'analyse est proposée. <p>Indicateur(s) :</p>

<ul style="list-style-type: none"> - Analyse des informations collectées - Mise en oeuvre d'un outil d'aide à la décision afin de résoudre un problème concret 			<ul style="list-style-type: none"> - les sources, canaux et fréquences sont identifiées dans la méthodologie de collecte
<p>A4. Diffusion des méthodes DevOps auprès des équipes de l'entreprise/organisation</p> <ul style="list-style-type: none"> - Définition des rôles d'une équipe DevOps - Constitution ou accompagnement à la création d'une équipe DevOps - Accompagnement à la sélection des outils de l'ensemble de la chaîne "Dev" et "Ops" 	<p>C7. Accompagner les collaborateurs au sein de l'équipe projet dans la sensibilisation et l'acculturation des méthodes d'organisation et de production DevOps de manière à optimiser le cycle de livraison d'un projet</p>	<p>E2. Etude de cas :</p> <p>A partir d'une étude de cas présentant une situation initiale, le/la candidat(e) doit :</p> <ul style="list-style-type: none"> - analyser la situation initiale, - exposer des objectifs permettant d'arrivée à la situation finale, - proposer des préconisations. 	<ul style="list-style-type: none"> - les acteurs impliqués dans une équipe DevOps, et les impacts de cette culture sur les conditions de travail sont listés précisément, - la compréhension des enjeux du DevOps lors de l'analyse de la situation initiale est démontrée, - des préconisations comprenant l'identification des outils DevOps sont proposées. leurs utilisations sont explicitées, - des leviers permettant l'acculturation des équipes sont décrits, - les freins et résistances aux changements potentiels sont identifiés et des réponses concrètes sont apportées. <p>Indicateur(s) :</p> <ul style="list-style-type: none"> - les acteurs et impacts sont listés de manière exhaustive, - une liste de préconisations est présentée

Glossaire :

- **DevOps** : Le devops est un mouvement en ingénierie informatique et une pratique technique visant à l'unification du développement logiciel (dev) et de l'administration des infrastructures informatiques (ops), notamment l'administration système.
- **Branche** : Une branche est une dérivation dans l'historique des changements du code source. On peut voir cela comme une copie des sources d'une application à un instant T à laquelle des modifications sont apportés en parallèle de celles apportées sur la branche principale. C'est un moyen de cloisonner le développement et le versionnement d'une application en nature/types de modifications (tests, fonctionnalités, environnements, etc).
- **Commit** : Ou "validation d'une transaction" correspond à une arborescence de fichiers (tree) à un instant T dans l'évolution du projet, enrichie de métadonnées comme un message de description, le nom de l'auteur, etc. Le "commit" pointe également vers un ou plusieurs objets commit parents pour former un graphe d'historiques.
- **Smoke test** : Les "smoke tests" sont une catégorie de tests réaliser dans le but d'attester du bon fonctionnement des fonctionnalités principales et essentielles de l'applications. Ces tests sont utilisés aussi pour déterminer si une application doit être sujette à une suite de tests plus précis.
- **Pull request** : Il s'agit d'une demande d'intégration des modifications du code source depuis un branche spécifique vers un autre branche, et plus généralement vers la branche principale (master).
- **Environnement de staging** : Il s'agit de l'environnement informatique reproduisant l'ensemble des caractéristiques de l'environnement de production. Il est principalement utilisé pour valider via des tests le bon fonctionnement de l'application sur un environnement distant (en opposition à celui de développement, local) reproduisant celui de production.

Modalités d'évaluations :

EVALUATIONS	DEROULEMENT (Contenu, durée, support autorisé, jury, nombre de page attendu, etc.)
E1 : Projet professionnel	<p>Contenu :</p> <p>À partir d'un cas d'entreprise réelle, le/la candidat(e) doit :</p> <ul style="list-style-type: none"> - améliorer un processus de développement d'applications à l'aide des principes de l'intégration continue. - mettre en oeuvre des conditions préalables à la livraison continue (identification des étapes de la chaîne de valeur, proposition de tests complémentaires, automatisation de la gestion des artefacts, sélections d'outils et de bibliothèques...). <p>Correction : Un jury composé de 3 personnes, dont au moins un professionnel.</p> <p>Rendus attendus :</p> <ul style="list-style-type: none"> • A l'écrit : Un rapport de 15 à 20 pages comprenant : <ul style="list-style-type: none"> ○ une introduction : contexte du projet, résumé exécutif, etc, ○ une première partie sur la compréhension du besoin, la traduction technique de la réponse apportée, ○ une seconde partie sur les enjeux de la mise en oeuvre du projet : justification des choix et des arbitrages réalisés, problèmes rencontrés et solutions apportés, etc, ○ une troisième partie sur le bilan de projet et les améliorations envisagées, ○ une conclusion : apprentissages, perspectives pour le projet professionnel envisagé, etc. • A l'oral : Une présentation orale de 30 mn découpée en 2 parties : <ul style="list-style-type: none"> ○ Présentation du rapport (15 mn), ○ Echange avec le jury (15 mn).

<p>E2. Etude de cas :</p>	<p>Contenu : A partir d'une étude de cas présentant une situation initiale, le/la candidat(e) doit :</p> <ul style="list-style-type: none"> - Analyser la situation initiale, - Exposer des objectifs permettant d'arrivée à la situation finale, - Proposer des préconisations. <p>Correction : Un jury composé de 3 personnes, dont au moins un professionnel.</p> <p>Rendus attendus :</p> <ul style="list-style-type: none"> - A l'écrit : Un rapport de 10 à 15 pages comprenant : <ul style="list-style-type: none"> - une introduction : contexte de l'étude, résumé exécutif, etc, - une première partie sur la compréhension besoin et la réponse apportée, - une seconde partie sur les enjeux de la mise en oeuvre de l'étude: justification des choix et des arbitrages réalisés, problèmes rencontrés et solutions apportés, etc, - une troisième partie sur le bilan de l'étude et les améliorations envisagées, - une conclusion : apprentissages, perspectives pour le projet professionnel envisagé par le candidat, etc. - A l'oral : Une présentation orale de 50 mn découpée en 2 parties, <ul style="list-style-type: none"> - présentation du rapport (25mn) - échange avec le jury (25mn).
----------------------------------	---