

Nom du candidat : _____
Prénom du candidat : _____

Date de l'évaluation : _____

BLOC	Compétences	Livrable attendu	Critères d'évaluation	Résultat de l'évaluation *	Commentaires
BLOC 2 : CONCEVOIR ET DEVELOPPER DES APPLICATIONS LOGICIELLES	C2.1.1 Mettre en œuvre des environnements de déploiement et de test en y intégrant les outils de suivi de performance et de qualité afin de permettre le bon déroulement de la phase de développement du logiciel	Le protocole de déploiement continu Les critères de qualité et de performance	Le protocole de déploiement continu est explicité. L'environnement de développement est détaillé (ex : éditeur de code, compilateur, etc.). Les outils mobilisés permettent d'identifier les composants comme : - le compilateur, - le serveur d'application, - les outils de gestion de sources, Le protocole permet de définir les différentes séquences de déploiement. Les critères de qualité et de performance permettent de répondre aux exigences du projet.		
	C2.1.2. Configurer le système d'intégration continue dans le cycle de développement du logiciel en fusionnant les codes sources et en testant régulièrement les blocs de code afin d'assurer un développement efficient qui réduit les risques de régression.	Le protocole d'intégration continue	Le protocole d'intégration continue est explicité clairement. Il permet de définir les séquences d'intégration.		
	C2.2.1. Concevoir un prototype de l'application logicielle en tenant compte des spécificités ergonomiques et des équipements cibles (ex : web, mobile...) afin de répondre aux fonctionnalités attendues et aux exigences en termes de sécurité.	Une architecture logicielle structurée permettant la maintenabilité du logiciel Une présentation d'un des prototypes réalisés L'utilisation de Framework et des paradigmes de développement	Les bonnes pratiques de développement sont respectées (ex : utilisation de framework, paradigmes de développement, ...). Le prototype est fonctionnel et il permet de répondre aux besoins identifiés. Le prototype met en œuvre un ensemble cohérent de fonctionnalités principales du logiciel et les user stories. Les composants de l'interface sont présents et fonctionnels. (ex : fenêtres, boutons, menus, ...) Le prototype permet de satisfaire aux exigences de sécurité.		
	C2.2.2 Développer un harnais de test unitaire en tenant compte des fonctionnalités demandées afin de prévenir les régressions et de s'assurer du bon fonctionnement du logiciel	Un jeu de tests unitaires couvrant une fonctionnalité demandée	Les tests unitaires couvrent la majorité du code développé.		
	C2.2.3 Développer le logiciel en veillant à l'évolutivité et à la sécurisation du code source, aux exigences d'accessibilité et aux spécifications techniques et fonctionnelles définies, pour garantir une exécution conforme aux exigences du client.	Une présentation des mesures de sécurité mises en œuvre Une présentation des actions mises en œuvre pour permettre l'accès à l'application aux personnes en situation de handicap	Les mesures prises permettent de couvrir les 10 failles de sécurité principales décrites par l'OWASP. Le référentiel d'accessibilité choisi est présenté et justifié. (ex : RGAA, OPQUAST, etc.) Le prototype permet de répondre aux exigences du référentiel d'accessibilité préalablement établi.		
	C2.2.4. 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.	L'historique des différentes versions La dernière version du logiciel fonctionnel, fiable et viable	Un système de gestion de versions est utilisé. Les évolutions du prototype sont tracées. Le logiciel est fonctionnel et manipulable en autonomie par un utilisateur.		
	C2.3.1 Élaborer le cahier de recettes en rédigeant les scénarios de tests et les résultats attendus afin de détecter les anomalies de fonctionnement et les régressions éventuelles	Le cahier de recettes	Le cahier de recettes reprend l'ensemble des fonctionnalités attendues. Les tests fonctionnels, structurels et de sécurité exécutés sont conformes au plan défini.		
	C2.3.2 Élaborer un plan de correction des bogues à partir de l'analyse des anomalies et des régressions détectées au cours de la recette afin de garantir le fonctionnement du logiciel conformément à l'attendu.	le plan de correction des bogues	Les bogues de codes sont détectés, qualifiés et traités. Une analyse des points d'amélioration est réalisée pour chaque test en échec. Les corrections et les améliorations proposées sont conformes à l'attendu et garantissent le bon fonctionnement du logiciel.		
	C2.4.1 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.	Le manuel de déploiement Le manuel d'utilisation Le manuel de mise à jour.	Les manuels sont rédigés avec clarté. La documentation permet de décrire les choix opérés en termes de technologies, de langages etc.		

* Acquis / Non Acquis

Membres du jury d'évaluation

NOM : _____

Prénom : _____

Signature

NOM : _____

Prénom : _____

Signature

NOM : _____

Prénom : _____

Signature