

Intitulé de la certification : « *Développer avec le langage RUST* »

## **Description du métier, de l'activité ou de la situation professionnelle à partir duquel le dispositif de formation visant la certification est initié**

Rust est un langage de programmation permettant aux développeurs d'écrire des applications informatiques diverses tel que les systèmes embarqués (y inclus critiques et temps-réel), cybersécurité, robotique, blockchain et transactions financières, programmation des réseaux, serveurs web, administration système, composants pour des systèmes d'exploitation de type Linux ou Android etc. L'intérêt de ce langage se situe principalement dans ses caractéristiques en termes de performance et de sûreté de fonctionnement.

Le métier de développeur informatique, tout comme celui d'ingénieur logiciel d'ailleurs, est en pleine mutation. Depuis 2019, la demande en compétences sur le langage Rust est en croissance exponentielle : le besoin en compétences sur ce langage, révélé par des études on-line ou au travers des offres d'emplois publiées, est sans équivoque : la connaissance du langage qui était "optionnelle" avant 2019 s'est transformée en une attente récurrente.

RUST est d'ailleurs le langage le plus apprécié des développeurs selon divers enquêtes, françaises ou internationales (Stack Overflow Developer Survey), et c'est le cinquième langage à la croissance la plus rapide selon l'étude State of the Octoverse de GitHub.

La certification « Développer avec le langage RUST » est destinée aux développeurs informatiques souhaitant attester de compétences complémentaires à leur métier, afin d'écrire des applications informatiques diverses, en programmant avec ce langage.

Elle vise à répondre aux défis liés au recrutement de personnel, à la demande croissante en professionnels du RUST, et à la variété des compétences requises dans le développement informatique. Elle s'intègre harmonieusement avec les diplômes existants, correspond aux besoins opérationnels des entreprises (favorisant ainsi les recrutements) et offre des opportunités d'acquisition de compétences supplémentaires pour améliorer l'employabilité (recrutements, missions en sous-traitance) et pour maximiser leurs opportunités professionnelles .

## **Précisions**

### **1- Candidats en situation de handicap**

Tout candidat peut saisir le référent handicap du certificateur, en amont de sa formation afin d'aménager les modalités d'évaluation. Sur le conseil du référent handicap et dans le respect du référentiel et de l'égalité entre les candidats, les modalités d'évaluation pourront être adaptées, ainsi que les supports et modalités nécessaires à la réalisation des évaluations.

### **2- Egalité Femmes – Hommes**

Dans ce référentiel, les termes « candidat », « référent », « professionnel » désignent tout autant des hommes que des femmes.

Référentiel de compétences	Référentiel d'évaluation	
	Modalités d'évaluation	Critères d'évaluation
<p>C1 - Identifier les opportunités d'usage du langage RUST lors de la réalisation d'une étude préliminaire, en élicitant les connaissances sur le contexte et les contraintes métier inhérentes au projet, afin de confirmer la pertinence de mobiliser ce langage et de limiter les coûts de développement pour le client</p> <p>C2 - Configurer l'environnement de développement, afin d'organiser le travail individuel et collaboratif, en se repérant dans l'écosystème Rust et en choisissant des ressources utiles à son projet (outillages et bibliothèques existantes), dans le cadre de la phase préparatoire du projet</p> <p>C3 - Traduire la logique métier en code Rust afin de répondre aux exigences client, en respectant les critères caractérisant les bonnes pratiques de l'ingénierie logicielle (prise en compte de l'accessibilité numérique, qualité du code, sécurité informatique, ...)</p>	<p>L'évaluation finale repose sur 3 épreuves :</p> <p><b>Epreuve 1 : production de code et fiche détaillée</b> Réalisation d'un projet réel ou fictif de développement d'une application, remis au jury 5 jours ouvrés avant la session de certification (pour évaluation par le jury). (épreuve notée sur 40 points).</p> <p>A partir d'un cahier des charges réel ou fictif au sujet d'une demande client, ainsi que de diverses ressources accessibles en ligne, le candidat devra produire un code pour implémenter une application, et rédiger une fiche technique détaillée (5 pages maximum). Attendus :</p> <ul style="list-style-type: none"> <li>étudier la faisabilité d'un développement en RUST sur le projet</li> <li>mettre en place et configurer un environnement de développement Rust, en vue de faciliter le travail individuel et en équipe</li> <li>convertir la logique métier inhérente au projet, en code Rust</li> <li>implémenter la fonctionnalité qui permet à un programme écrit en Rust d'interagir avec des programmes ou des bibliothèques écrites dans d'autres langages de programmation</li> <li>vérifier la qualité et la conformité du code RUST</li> <li>décrire le process à mettre en œuvre pour maintenir l'application RUST, et les correctifs à appliquer si besoin</li> </ul>	<p>Cr1 – Le candidat énumère ses facteurs de choix en termes de langage de programmation dans le contexte du projet</p> <p>Cr2 – Dans le cadre d'une application existante, il fournit une comparaison factuelle entre Rust et le langage existant, basée sur des critères mesurables (comme la performance et la sécurité)</p> <p>Cr3 – Dans le cadre d'une nouvelle application, il justifie le choix de Rust avec des arguments spécifiques liés aux besoins du projet, appuyés par des données ou des exemples pertinents.</p> <p>Cr4 – Il installe et configure de manière efficace l'environnement de développement Rust, y compris les outils essentiels comme Cargo et Rustup, et vérifie leur bon fonctionnement</p> <p>Cr5 – Il identifie et intègre des outils et bibliothèques adaptés aux besoins spécifiques du projet, en justifiant leur choix en fonction de leur utilité/compatibilité avec les objectifs du projet</p> <p>Cr6 – Il met en place une structure de projet qui facilite le travail individuel et collaboratif</p> <p>Cr7 – Il mobilise des outils de versionnage et de collaboration adaptés</p> <p>Cr8 - Il démontre une compréhension claire des meilleures pratiques pour le travail en équipe dans l'écosystème Rust</p> <p>Cr9 – Il choisit les structures de données adaptées aux exigences spécifiques du client</p>

Référentiel de compétences	Référentiel d'évaluation	
	Modalités d'évaluation	Critères d'évaluation
<p>C4 - Intégrer un code Rust avec du code écrit dans d'autres langages, en écrivant les passerelles (bindings) d'un langage à l'autre, afin d'assurer l'interopérabilité avec des composants existants</p> <p>C5 - Tester du code Rust afin de s'assurer de la qualité et de la conformité de la solution (ou d'une fonctionnalité), en écrivant et en exécutant des tests unitaires ou d'intégration en Rust</p> <p>C6 - Maintenir une application Rust afin de garantir son bon fonctionnement dans la durée et de s'adapter aux changements des besoins, en identifiant les évolutions et en appliquant les correctifs nécessaires</p>	<ul style="list-style-type: none"> <li>développer une application qui intègre des composants écrits en Rust avec d'autres composants</li> <li>écrire des artefacts appropriés et utiles au packaging de l'application</li> </ul> <p><b>Epreuve 2 : questionnaire</b> La complétion en ligne (sur plateforme avec accès personnel et contrôle de l'identité du candidat) d'un questionnaire professionnel de 40 questions tirées de manière aléatoire parmi une banque de plus de 200 questions (épreuve notée sur 20 points). Durée : 1 heure.</p> <p>Les questions portent sur l'ensemble des connaissances théoriques et pratiques liées au langage RUST et qui permettent d'évaluer la connaissance sur ce langage.</p> <p>Une fois l'heure écoulée, les réponses du candidat seront automatiquement enregistrées. Elles seront par la suite transmises au Jury pour la notation.</p>	<p>Cr10 – Il écrit du code Rust idiomatique</p> <p>Cr11 – Il se sert de la chaîne d'outils Rust afin de respecter les pratiques de l'ingénierie logicielle</p> <p>Cr12 - Les principes d'accessibilité numérique, la propreté et la maintenabilité du code sont respectés, ainsi que les pratiques recommandées pour la sécurité informatique</p> <p>Cr13 – L'utilisation d'une bibliothèque C dans une application Rust avec la génération automatique des bindings est démontrée</p> <p>Cr14 – L'utilisation d'une bibliothèque Rust dans une application C ou Python avec la génération automatique des bindings est démontrée</p> <p>Cr15 - Le candidat développe des tests unitaires et d'intégration en Rust qui couvrent de manière exhaustive les fonctionnalités clés du code</p> <p>Cr16 – Il écrit et organise le code de test dans le respect des pratiques de l'ingénierie logicielle</p> <p>Cr17 – Il identifie les besoins d'évolution de l'application Rust et implémente les modifications ou les améliorations nécessaires de manière appropriée</p> <p>Cr18 – Il démontre l'usage des outils appropriés (debugueur) pour analyser un bug et le corriger</p> <p>Cr19 – Il s'assure que les composants intégrés fonctionnent de manière stable et sont compatibles avec l'ensemble du système</p>

Référentiel de compétences	Référentiel d'évaluation	
	Modalités d'évaluation	Critères d'évaluation
<p>C7 - Construire une application incluant du code Rust, en réalisant l'intégration du/des composants Rust avec les autres, afin de répondre à l'ensemble des exigences clients</p> <p>C8 - Préparer une application écrite en Rust afin de la déployer, en écrivant les scripts nécessaires à la compilation de cette application Rust, et en écrivant les artefacts nécessaires au packaging de l'application, dans la cadre de la mise en production</p>	<p><b>Epreuve 3</b> La soutenance devant un jury professionnel (épreuve notée sur 40 points). Durée : 40 minutes.</p> <p>Le candidat :</p> <ul style="list-style-type: none"> <li>• soutiendra sa production devant le jury</li> <li>• et répondra aux questions du jury</li> </ul> <p>Composition du jury : 1 représentant du certificateur, et 2 professionnels, experts du langage RUST et extérieurs à l'organisme certificateur.</p> <p>La certification est acquise si le candidat obtient une note supérieure ou égale à 54 points sur 100, sans avoir obtenu de notes éliminatoires aux 3 épreuves (notes inférieures à 20 pour la production écrite et la soutenance, note inférieure à 14 pour le questionnaire).</p>	<p>Cr20 - Il résout les problèmes d'interopérabilité qui pourraient survenir</p> <p>Cr21 - Le candidat identifie les exigences liées à la mise en production</p> <p>Cr22 - Le candidat compile du code Rust pour la mise en production</p> <p>Cr23 - Le candidat compile du code Rust dans un pipeline d'intégration continue</p> <p>Cr24 - Le candidat package une application Rust pour son déploiement</p> <p>Epreuve 2 : le candidat obtient à minima 70% de bonnes réponses, soit plus de 28 réponses justes sur 40 questions posées.</p>