Durée :	1 jour soit 7 heures	Référence :	GH-600T00

Public visé :

Les participants doivent disposer d'une expertise métier dans l'exploitation, l'intégration, la supervision et la gouvernance des agents d'IA au sein de workflows SDLC (*Software Development Lifecycle*) et d'environnements de développement de niveau production, afin de garantir la fiabilité, la sécurité et la rapidité d'exécution en utilisant GitHub comme système de référence (*system of record*) et plan de contrôle (*control plane*).

Les participants travaillent en étroite collaboration avec des architectes, des ingénieurs plateforme, des ingénieurs DevOps, des développeurs d'applications, des responsables produit et des ingénieurs sécurité afin de concevoir, déployer, exploiter et gérer des agents fonctionnant au sein de la plateforme GitHub.

Les participants doivent avoir une expérience du cycle de vie du développement logiciel (*SDLC*), des workflows et des mécanismes de contrôle dans GitHub, ainsi que des pratiques liées à la qualité du code, à la sécurité et aux revues de code.

Ils doivent également avoir une expérience des agents de développement, notamment GitHub Copilot, des serveurs MCP (*Model Context Protocol*) et de la personnalisation des agents, tels que :

- les instructions personnalisées (*custom instructions*),
- les agents personnalisés (*custom agents*),
- les outils (*tools*),
- la configuration de Copilot.

Responsabilités associées à ce rôle :

- Exploiter les workflows d'agents au sein du SDLC
- Superviser les comportements autonomes à l'aide des contrôles GitHub
- Évaluer et ajuster les résultats produits par les agents à l'aide d'analyses (*scans*) et d'artefacts
- Configurer des agents personnalisés
- Coordonner l'exécution de systèmes multi-agents de manière sécurisée

Pré-requis :

Prérequis obligatoires :

- Expérience pratique de GitHub : workflows, Pull Requests, mécanismes de contrôle (branch protection, Actions)
- Pratique des enjeux de qualité de code, de sécurité et de revue de code
- Expérience d'utilisation des agents de développement (GitHub Copilot) et connaissance des notions d'instructions personnalisées, d'agents personnalisés et d'outils (tools)
- Compréhension générale du cycle de vie du développement logiciel (SDLC)

Prérequis recommandés (non obligatoires) :

- Notion préalable du Model Context Protocol (MCP) et de son usage avec des serveurs d'outils
- Une première expérience de configuration ou de personnalisation de Copilot

Objectifs pédagogiques :

- Décrire le rôle et le cycle de vie d'un agent d'IA au sein d'un workflow GitHub
- Concevoir une architecture d'agent adaptée à une étape du SDLC, avec entrées/sorties et critères de réussite définis
- Mettre en place une gouvernance des contributions générées par des agents (PR, CODEOWNERS, environment gates)
- Configurer un environnement d'exécution d'agent sécurisé via MCP
- Orchestrer et superviser un système multi-agents dans GitHub de façon fiable et observable
- Définir des indicateurs d'évaluation et des garde-fous garantissant un usage sûr et gouverné des agents en production

Compétences acquises à l'issue de la formation :

- Être capable de situer le rôle d'un agent d'IA dans le cycle de vie du développement logiciel (SDLC) et d'identifier les risques, anti-patterns et exigences de traçabilité associés à son usage.
- Savoir concevoir l'architecture d'un agent (entrées, sorties, critères de réussite) en séparant clairement les phases de planification, de raisonnement et d'exécution.
- Maîtriser la mise en place de la gouvernance des Pull Requests générées par des agents (templates, checks,

CODEOWNERS, règles d'accès aux environnements).
- Être en mesure de configurer et sécuriser les outils et environnements d'exécution des agents, notamment via les serveurs MCP (registres, listes d'autorisation, limites d'exécution).
- Savoir orchestrer un système multi-agents dans GitHub (isolation des exécutions, gestion des conflits, observabilité des transmissions entre agents).
- Être capable de mettre en œuvre une stratégie de mémoire et de suivi d'état des agents, et de définir des indicateurs d'évaluation pour analyser et améliorer leur comportement.
- Maîtriser l'application de garde-fous et d'une gouvernance basée sur le risque (principe du moindre privilège, intervention humaine, traçabilité et auditabilité des actions).

Modalités pédagogiques :

Session dispensée en présentiel ou téléprésentiel, selon la modalité inter-entreprises ou intra-entreprises sur mesure. La formation est animée par un(e) formateur(trice) durant toute la durée de la session et présentant une suite de modules théoriques clôturés par des ateliers pratiques validant l'acquisition des connaissances. Les ateliers peuvent être accompagnés de Quizz. L'animateur(trice) présente la partie théorique à l'aide de support de présentation, d'animation réalisée sur un environnement de démonstration. En présentiel comme en téléprésentiel, l'animateur(trice) accompagne les participants durant la réalisation des ateliers.

Note relative au programme : Les ateliers, démonstrations, travaux pratiques et scénarios présentés dans ce programme sont fournis à titre d'exemples illustratifs et ne sont pas exhaustifs. Afin de garantir une adéquation permanente avec les évolutions technologiques, le contenu pédagogique ainsi que les environnements de travaux pratiques (TP) sont susceptibles d'être adaptés, ajustés ou mis à jour au cours du temps.

Moyens et supports pédagogiques :

Cadre présentiel

Salles de formation équipées et accessibles aux personnes à mobilité réduite.

- Un poste de travail par participant
- Un support de cours numérique ou papier (au choix)
- Un bloc-notes + stylo
- Vidéoprojection
- Connexion Internet
- Accès extranet pour partage de documents et émargement électronique

Cadre téléprésentiel

Session dispensée via notre solution iClassRoom s'appuyant sur Microsoft Teams.

- Un compte Office 365 par participant
- Un poste virtuel par participant
- Un support numérique (PDF ou Web)
- Accès extranet pour partage de documents et émargement électronique

Modalités d'évaluation et de suivi :

Avant

Afin de valider le choix d'un programme de formation, une évaluation des prérequis est réalisée à l'aide d'un questionnaire en ligne ou lors d'un échange avec le formateur(trice), qui valide la base de connaissances nécessaires.

Les stagiaires disposent également, depuis leur extranet, d'un **outil d'auto-positionnement** leur permettant d'évaluer leur niveau de maîtrise des compétences visées par la formation. Cette auto-évaluation est réalisée avant le démarrage de la formation à l'aide d'un curseur de positionnement allant du niveau le plus faible au niveau le plus élevé.

Pendant

Après chaque module théorique, un ou plusieurs ateliers pratiques permettent de valider progressivement l'acquisition des connaissances. Un quiz peut accompagner les ateliers pratiques afin de mesurer la compréhension des notions abordées.

Après

À l'issue de la formation, les stagiaires réalisent une nouvelle **auto-évaluation** sur les mêmes compétences que celles évaluées

avant la formation. Cette comparaison permet de mesurer leur perception de la progression réalisée et de visualiser les compétences développées au cours de la formation.

Lorsque le programme de formation le prévoit, un examen de certification est organisé conformément aux modalités définies par l'éditeur ou le centre de certification (TOSA, Pearson VUE, ENI, PeopleCert…).

Le formateur réalise également une **évaluation individuelle des compétences** de chaque stagiaire sur l'ensemble des compétences visées par la formation. Pour chacune d'elles, il indique si la compétence est :

- Acquise ;
- En cours d'acquisition ;
- Non acquise.

Lorsque la compétence est évaluée comme **en cours d'acquisition** ou **non acquise**, le formateur formule des observations précisant les difficultés rencontrées ainsi que les actions ou recommandations permettant au stagiaire de poursuivre sa montée en compétences.

Enfin

Un questionnaire de satisfaction permet au participant d'évaluer la qualité de la prestation de formation.

Informations sur l'accessibilité :

Nos formations sont, dans la mesure du possible, conçues pour être accessibles à toutes et à tous. Afin de garantir les meilleures conditions d'accueil et d'apprentissage pour les personnes en situation de handicap, nous vous invitons à contacter notre référente handicap certifiée :

Céline SOLATGES – 05 61 34 39 80 – csolatges@iform.fr

Nous vous remercions de bien vouloir nous communiquer toute information utile à ce sujet en amont de la formation, afin de mettre en place les adaptations nécessaires et d'assurer un accompagnement optimal.

Pour en savoir plus sur les dispositifs d'accompagnement existants, vous pouvez consulter les sites suivants :

- AGEFIPH
- FIPHFP
- MDPH de votre département

Description / Contenu

Ce cours est conçu pour développer des compétences pratiques dans la conception, le déploiement et la gestion de systèmes d'IA agentique dans le cadre de workflows de développement logiciel basés sur GitHub.

Le cours explore l'intégration d'agents d'IA dans le cycle de vie du développement logiciel (*Software Development Lifecycle - SDLC*), notamment la conception d'architectures d'agents, la configuration des outils et des environnements, ainsi que la gestion de la mémoire, de l'état et de l'exécution des agents.

Les participants apprendront à évaluer et optimiser les performances des agents, à mettre en œuvre des mécanismes de gouvernance et des garde-fous (*guardrails*), ainsi qu'à orchestrer des systèmes multi-agents afin de garantir des résultats sûrs, fiables et efficaces.

Grâce à une approche pratique (*hands-on learning*), les participants acquerront les compétences nécessaires pour exploiter, superviser et gouverner des agents d'IA dans des environnements de production, en utilisant GitHub comme plan de contrôle (*control plane*).

Module 1 : Fondamentaux de l'IA agentique dans GitHub

- Définir l'IA agentique dans le SDLC (*Software Development Lifecycle*)
- Expliquer le cycle de vie d'un agent : planifier, agir, évaluer
- Décrire GitHub comme système de référence (*system of record*) et plan de contrôle (*control plane*)
- Identifier les responsabilités, les risques, les anti-modèles (*anti-patterns*) et les besoins de traçabilité
- Appliquer le modèle de contribution aux travaux générés par des agents

Module 2 : Concevoir l'architecture des agents et leur intégration dans le SDLC (*Software Development Lifecycle*)

- Mapper les responsabilités des agents au SDLC (*Software Development Lifecycle*)
- Définir les entrées, les sorties et les critères de réussite
- Séparer la planification, le raisonnement et l'exécution
- Exemples de mise en œuvre de la gouvernance des Pull Requests (PR) avec des modèles (*templates*), des

contrôles (*checks*), CODEOWNERS, des règles et des contrôles d'accès aux environnements (*environment gates*)
- Construire des workflows fiables : sorties, contextes, déclencheurs et transmissions entre tâches (*cross-job handoffs*)
- Contrôler et exploiter les agents : observabilité, outils, MCP (*Model Context Protocol*), secrets, hooks et fiabilité

Module 3 : Outils, MCP (*Model Context Protocol*) et environnements d'exécution des agents
- Comment les agents interagissent avec les API GitHub et les workflows
- Serveurs MCP (*Model Context Protocol*), registres et listes d'autorisation (*allow lists*)
- Contexte d'exécution et limites
- Limites d'exécution et protections des agents

Module 4 : Systèmes multi-agents et orchestration
- Définir les responsabilités des systèmes multi-agents dans le SDLC (*Software Development Lifecycle*)
- Orchestrer les agents à l'aide des workflows GitHub
- Isoler l'exécution : branches, workflows, autorisations et concurrence
- Détecter et résoudre les conflits à l'aide de mécanismes d'arbitrage natifs GitHub
- Rendre le système observable : attribution, preuves et transmissions (*handoffs*)
- Exploiter de manière fiable à grande échelle : diagnostiquer les échecs et effectuer des récupérations sécurisées

Module 5 : Mémoire, état et évaluation
- Mettre en œuvre des stratégies de mémoire pour les agents
- Conserver l'état des agents et gérer la dérive du contexte (*context drift*)
- Garantir la continuité de la mémoire et de l'état des agents à travers les outils et les environnements
- Définir des indicateurs d'évaluation et appliquer des contrôles qualité (*quality gates*)
- Analyser les échecs des agents et améliorer leur comportement

Module 6 : Gouvernance, garde-fous (*guardrails*) et exploitation
- Définir une autonomie basée sur les risques et les limites d'action
- Appliquer la gouvernance avec les contrôles GitHub
- Concevoir des workflows avec intervention humaine (*human-in-the-loop*)
- Contrôler les capacités des agents selon le principe du moindre privilège (*least privilege*)
- Rendre les actions observables, traçables et auditables
- Maintenir la gouvernance et la fiabilité opérationnelle