

Durée : 3 jours soit 21 heures

Référence : IF-IADEV2



Public visé :

Développeurs, lead developers, tech leads et architectes logiciels qui utilisent déjà un assistant IA et veulent confier des tâches complètes à des agents, sans perdre la maîtrise de la conception ni de la qualité.



Pré-requis :

- Pratique professionnelle d'au moins un langage de programmation.
- Usage courant de Git : branches, fusion, pull requests.
- Aisance avec la ligne de commande.
- Première expérience d'un assistant IA de code (niveau IF-IADEV1 ou équivalent).



Objectifs pédagogiques :

- Mettre en œuvre un cycle explorer, planifier, implémenter, vérifier avec un agent de code.
- Structurer le contexte d'un projet pour les agents : instructions, conventions, spécifications.
- Faire réaliser une fonctionnalité complète pilotée par la spécification et les tests.
- Intervenir avec un agent sur du code existant sans régression.
- Étendre un agent avec des outils (MCP) et des instructions réutilisables (skills, commandes, sous-agents).



Compétences acquises à l'issue de la formation :

- Être capable de piloter un agent de code sur une tâche complète en appliquant le cycle explorer, planifier, implémenter, vérifier.
- Savoir structurer le contexte d'un projet pour obtenir un code cohérent avec l'existant et ses conventions.
- Être en mesure de faire produire une fonctionnalité vérifiée par des tests automatisés, à partir d'une spécification.
- Savoir faire évoluer ou corriger un code existant avec un agent, sous couvert de tests de non-régression.
- Être capable d'étendre un agent (serveurs MCP, skills, sous-agents) et de régler ses permissions selon le risque de la tâche.



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
- Vidéo-projection
- 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 :

- Acquis ;
- 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

Module 1 – De l'assistant à l'agent

IFORM : Agence de Balma : 16 avenue Charles de Gaulle, bâtiment 13, 31130 Balma

Siège social : Mbe 308 46 rue André Vasseur 31200 TOULOUSE

Tél : 05 40 24 86 95 • SIRET : 431 421 742 00096 • APE : 6202A

- Boucle agentique : raisonnement, appels d'outils, lecture et écriture de fichiers, exécution de commandes
- Vibe coding et développement piloté : acceptable pour un prototype, pas pour la production
- Agents dans l'IDE (GitHub Copilot, Cursor, Kiro), en ligne de commande (Claude Code, Gemini CLI, Codex CLI) et dans le cloud (agent de code GitHub Copilot)
- Permissions, approbations et bac à sable : régler le niveau d'autonomie
- Démonstration : une même évolution réalisée en vibe coding puis en mode piloté, comparaison des résultats

Module 2 – Préparer le projet pour les agents (context engineering)

- Fichiers d'instructions : AGENTS.md, CLAUDE.md, copilot-instructions.md, règles par dossier
- Contenu utile : commandes de build et de test, conventions, architecture, interdicts
- Gestion de la fenêtre de contexte : compaction, nouvelle session, notes de travail
- Atelier : rendre un projet existant « prêt pour les agents » et mesurer l'effet sur une même tâche

Module 3 – Le cycle explorer, planifier, implémenter, vérifier

- Exploration du code avant toute modification
- Mode plan : produire, critiquer et amender un plan avant exécution
- Implémentation par petites étapes, commits intermédiaires, retour arrière
- Vérification : tests, linters, exécution, relecture du diff
- TP : implémenter une fonctionnalité transverse (pagination et filtrage d'une API avec affichage côté front) en suivant le cycle complet

Module 4 – Développer à partir de la spécification et des tests

- Du besoin à une spécification exploitable : user stories, critères d'acceptation, contraintes techniques
- Découpage en tâches vérifiables ; outillage spec-driven (GitHub Spec Kit, spécifications Kiro) : intérêt et limites
- TDD avec un agent : tests d'abord, implémentation ensuite, tests verrouillés

- Tests de bout en bout pilotés par l'agent avec Playwright
- TP : à partir d'un besoin métier rédigé, produire la spécification, le plan de tâches, les tests puis l'implémentation

Module 5 – Intervenir sur du code existant

- Cartographier un module ancien et documenter son comportement
- Refactoriser sous couvert de tests de caractérisation
- Débuguer avec méthode : reproduction, hypothèse, correctif, test de non-régression
- TP : refactoriser un module fourni sans régression, puis corriger une anomalie décrite dans un ticket

Module 6 – Étendre l'agent : MCP, skills, sous-agents, hooks

- Model Context Protocol : principe client-serveur, outils, ressources ; serveurs courants (Playwright, Context7, GitHub, base de données)
- Risques des serveurs MCP : droits accordés, données exposées, injection d'instructions via un contenu externe
- Skills (format SKILL.md), commandes et prompts réutilisables
- Sous-agents et agents personnalisés : déléguer la revue, les tests ou la recherche
- Hooks : déclencher formatage, tests ou contrôles automatiquement
- Atelier : connecter deux serveurs MCP, créer une skill de projet et un sous-agent de revue

Module 7 – Git, revue et agents en parallèle

- Branches et worktrees pour faire travailler plusieurs agents sans conflit
- Délégation asynchrone : confier un ticket à un agent cloud et relire sa pull request
- Revue de code assistée : obtenir une critique, vérifier chaque affirmation de l'agent
- Choisir son outil et maîtriser sa consommation : modèles, quotas, coûts
- Cas de synthèse évalué : faire évoluer l'application fil rouge d'une fonctionnalité complète, de la spécification à la pull request relue