

Sessions garanties disponibles

Formation Développeur augmenté par l'IA : pratiquer le développement agentique

Réf. SF34446

Faites des agents de code IA de vrais coéquipiers : donnez-leur le bon contexte, pilotez-les de l'exploration du code jusqu'à la pull request et gardez la maîtrise de ce qu'ils produisent.

2 j Durée	1 600 € HT Inter / participant	4 000 € HT Intra (≤12 pers.)	13 Sessions à venir
--------------	-----------------------------------	---------------------------------	------------------------

Objectifs pédagogiques

- ✓ Choisir et configurer un agent de code adapté à son contexte, en maîtrisant ses permissions, les données transmises et les coûts
- ✓ Structurer le contexte du projet et ses demandes pour obtenir de l'agent des résultats fiables
- ✓ Conduire une tâche de développement de bout en bout avec un agent, de l'exploration du code à la validation
- ✓ Utiliser commandes, skills, sous-agents, hooks et serveurs MCP pour automatiser les tâches récurrentes
- ✓ Évaluer de manière critique le code produit par un agent (revue de diff, tests, sécurité) avant de l'intégrer
- ✓ Formaliser des standards et une charte d'usage des agents de code pour son équipe

Public visé & prérequis

Public visé Architectes logiciels, Développeurs, Testeurs logiciels

Prérequis- Maîtriser au moins un langage de programmation et avoir l'expérience de projets réels - Pratiquer Git au quotidien : branches, diff, pull requests - Être à l'aise avec la ligne de commande et un IDE moderne

Programme détaillé

Du copilote à l'agent : comprendre le développement agentique

- Assistant de complétion, chat, agent de code : ce qui change quand l'IA lit le dépôt, modifie plusieurs fichiers et exécute des commandes
- La boucle agentique : lire, planifier, éditer, exécuter, vérifier, et pourquoi chaque étape peut dérailler
- Tokens et fenêtre de contexte : ce que l'agent voit réellement et pourquoi il se dégrade sur les longues sessions
- Erreurs typiques du code généré : API inventées, code plausible mais faux, tests contournés, excès de confiance
- Vibe coding ou agentic coding : génération sans contrôle ou collaboration cadrée et vérifiée
- Panorama des outils : agents en ligne de commande, agents intégrés à l'IDE, agents cloud, et critères de choix (stack, confidentialité, coût, pratiques d'équipe)
- Quand ne pas confier une tâche à un agent : code critique mal testé, exigences floues, apprentissage d'une notion

Activités pratiques :

- Même tâche traitée par un chat, un assistant d'IDE et un agent : comparaison du résultat, du temps et du contrôle
- Cartographie de ses tâches quotidiennes : celles à déléguer, à assister, à garder

Installer et encadrer son agent de code

- Mise en route : installation dans le terminal ou l'IDE, connexion au dépôt, premier échange sur le projet
- Modes d'approbation et permissions : lecture, écriture, exécution de commandes, et réglages adaptés à chaque contexte
- Exécution isolée : intérêt d'un environnement cloisonné dès que l'agent agit de façon autonome
- Confidentialité : données envoyées au fournisseur, exclusion de fichiers, secrets maintenus hors de portée de l'agent
- Coûts : abonnements, consommation de tokens, choix du modèle selon la tâche
- Git comme filet de sécurité : branche de travail, commits fréquents, retour arrière sur une modification ratée

Activités pratiques :

- Installation et configuration de l'agent sur le dépôt fil rouge, réglage des permissions
- Vérification de ce qui est transmis au fournisseur et exclusion des fichiers sensibles
- Annulation propre d'une série de modifications de l'agent avec Git

Donner le bon contexte à l'agent

- Le contexte, premier levier de qualité : pourquoi une demande juste échoue sans les bonnes informations
- Fichier d'instructions du projet (AGENTS.md et équivalents propres à chaque outil) : commandes de build et de test, conventions, architecture, interdits
- Ce qu'il ne faut pas y mettre : documentation exhaustive, règles contradictoires, informations qui changent souvent
- Structurer une demande de développeur : objectif, contraintes, fichiers concernés, critères d'acceptation
- Pointer la bonne information : fichiers ciblés, extraits de documentation, exemples de code existant, captures d'écran
- Gérer la session : repartir à neuf, faire résumer, découper une tâche trop large
- Itérer sur une réponse : corriger la trajectoire plutôt que tout redemander

Activités pratiques :

- Rédaction du fichier d'instructions du dépôt fil rouge, puis comparaison d'une même demande avant et après
- Réécriture d'une demande vague en demande cadrée avec critères d'acceptation

Le workflow agentique : explorer, planifier, implémenter, valider

- Explorer une codebase inconnue : faire cartographier l'architecture, les flux et les zones à risque avant de toucher au code
- Mode plan : faire proposer un plan, le discuter et l'amender avant toute modification
- Implémenter par petites étapes vérifiables : un changement, un test, un commit
- Tests d'abord quand c'est possible : écrire ou faire écrire le test qui définit le comportement attendu
- Déboguer avec l'agent : reproduire, isoler, corriger, puis ajouter le test de non-régression
- Refactoring de code legacy : sécuriser par des tests de caractérisation avant de restructurer
- Documentation et messages de commit : produire à partir du code et du diff une documentation cohérente

Activités pratiques :

- Scénario legacy : exploration d'un module ancien, tests de caractérisation, puis refactoring guidé
- Scénario fonctionnalité : plan validé puis implémentation multi-fichiers sur le dépôt fil rouge
- Correction d'un bug réel du dépôt, du symptôme au test de non-régression
- Génération de la documentation technique d'un module existant

Étendre l'agent : commandes, skills, sous-agents, hooks et MCP

- Des mécanismes communs aux principaux agents : mêmes concepts, noms et formats propres à chaque outil
- Commandes et prompts réutilisables : encapsuler une tâche récurrente (revue, préparation de PR, rédaction de tests)
- Skills : procédures et connaissances chargées seulement quand la tâche les demande
- Sous-agents : déléguer une recherche ou une revue dans un contexte isolé, sans encombrer la session principale
- Hooks : déclencher automatiquement un contrôle à chaque modification (formatage, lint, tests)
- MCP (Model Context Protocol) : brancher l'agent sur des outils existants (navigateur, documentation à jour, ticketing) et évaluer la confiance accordée à un serveur
- Choisir le bon mécanisme : instruction, commande, skill, hook ou serveur MCP selon le besoin

Activités pratiques :

- Création d'une commande de revue de code et d'un hook de lint sur le dépôt fil rouge
- Connexion d'un serveur MCP de navigateur pour tester une interface et générer un test de bout en bout
- Délégation d'une recherche dans la codebase à un sous-agent

Prochaines sessions

DATE	LIEU	STATUT
07/12/2026	à distance	Sur confirmation
19/01/2027	Lyon	Sur confirmation
19/01/2027	Nantes	Sur confirmation
19/01/2027	Toulouse	Sur confirmation
19/01/2027	à distance	Sur confirmation

DATE	LIEU	STATUT
29/04/2027	Grenoble	Sur confirmation

Pourquoi choisir Sparks Formation ?

20 ans d'expertise IT, des formateurs experts et des sessions garanties contre l'annulation.

20 ans
d'expertise

1 068+
formations

11 178
apprenants formés/an

+67
Net Promoter Score

Avis des participants

4,9/5 — 165 avis

Selim B. — Très bonne formation, très instructif même après quelques années de Python dans les bras. Une très petite équipe de stagiaires qui facilite les échanges. Je redoutais le mode...

Alexis C. — Anthony explique bien les concepts attendus et prend le temps de répondre aux questions et aux différents problèmes rencontrés. Le support présenté créé par lui-même est différent des...

Votre formateur

Gérald B. — Formateur

Gérald intervient notamment sur la gestion de projet et les certifications associées, ainsi que sur l'IA appliquée aux métiers et sa gouvernance. Son activité de formation remonte à 2015, soit...

Ils nous font confiance

Cette année notre filière Développement et Ingénierie de Logiciels a accompagné plus de 459 entreprises dont :

SOPRA STERIA RENAULT SAS ORANGE CAMPUS TECH ORANGE INNOVATION SCHOOL STMICROELECTRONICS ARPEGE GRANT THORNTON FRANCE ORANGE BUSINESS RTE SII ILE DE FRANCE +449 autres

Financement & certifications

Cette formation est éligible aux financements OPCO et FNE-Formation. Nos équipes vous accompagnent dans le montage de votre dossier.



Réservez votre place dès maintenant

20 ans d'expertise · 1 000+ formations · NPS +67 · réponse sous 24h ouvrées

0 805 950 800

demande@sparks-formation.com