Développeur Augmenté par l'IA - Maîtriser l'Agentic Coding

Durée: 2 Jours **Réf de cours: SFI-GENAI200DEVAUG** **Méthodes d'apprentissage: Intra-entreprise & sur-mesure**

Résumé:

Accélérez la productivité de vos équipes de développement grâce à l'Agentic Coding.

Dans un marché où la rapidité de développement et la qualité du code font la différence, cette formation transforme vos développeurs en "développeurs augmentés", capables de tirer parti des agents d'IA les plus avancés.
Centrée sur des cas d'utilisation réels, la formation met l'accent sur la production de valeur immédiate et l'amélioration concrète de la productivité des développeurs. Les participants repartent avec des méthodes, workflows et assets directement applicables à leurs projets en entreprise.
Vos équipes apprendront à orchestrer et collaborer avec des agents intelligents pour générer du code fiable et maintenable, automatiser les tests et la documentation, accélérer le refactoring et la résolution de bugs, et standardiser les pratiques de développement assisté par IA.
La formation est adaptée aux outils leaders du marché (Claude Code, Gemini CLI, GitHub Copilot, Cursor AI, etc.) et apprend aux participants à créer leurs propres agents métiers personnalisés. À noter : le choix des outils est actualisé en fonction des solutions les plus pertinentes et matures du marché.
En adoptant cette approche, vos équipes passeront du "vibe coding" à une méthodologie agentique structurée, garantissant qualité, sécurité et maintenabilité du code tout en accélérant drastiquement le delivery.
À la fin de cette formation, l'évaluation finale sous la forme d'un test de certification permet aux participants d'obtenir la certification SFEIR Certified AI-Augmented Developer.

Mise à jour Octobre 2026

Formation intra-entreprise

Cette formation est délivrable en session intra-entreprise, dans vos locaux ou dans les nôtres. Son contenu peut être adapté sur-mesure pour répondre aux besoins de vos collaborateurs. Contactez votre conseiller formation Global Knowledge ou adressez votre demande à info@globalknowledge.fr.

Public visé:

Software Engineer (développeur.se.s opérationnel.le.s backend/frontend), Software Architect, Tech Leader en entreprise, ESN, startups et scale-ups qui souhaitent booster leur efficacité grâce à l'IA, tout en conservant un haut niveau de qualité de code., Équipes soucieuses de la maintenabilité, de la robustesse et des bonnes pratiques, qui cherchent à utiliser l'IA pour moderniser leurs méthodes tout en améliorant la qualité de leurs livrables.

Objectifs pédagogiques:

- **Cette formation abordera les thèmes suivants :**
- Accélérer toutes les phases du cycle de développement, de la conception architecturale jusqu'à la résolution de bugs complexes, en augmentant leur vélocité tout en assurant la qualité, la sécurité, les tests, la documentation et la maintenabilité du code.
- Collaborer en continu avec l'IA, en adoptant des méthodes de travail augmentées par l'IA qui renforcent la productivité et l'efficacité.
- Maîtriser l'écosystème des outils phares permettant de coder avec l'IA et framework IA (type Claude Code, Gemini CLI, GitHub Copilot, Cursor AI), évaluer et adopter les nouveaux outils émergents, en comprenant leurs forces et limites respectives.
- Transformer une équipe classique en une équipe augmentée, en orchestrant des agents IA collaboratifs, en mettant en place des standards d'équipe (contextes partagés, prompts réutilisables) et en diffusant de bonnes pratiques de développement assisté par IA qui accélèrent l'onboarding et homogénéisent les workflows.

Pré-requis:

- Maîtrise pratique d'au moins un langage de programmation (TypeScript, Java, Python, JavaScript, C#, Go...).
- Expérience quotidienne avec Git et l'utilisation d'un IDE moderne

Test et certification

Aucun

(VS Code, IntelliJ, WebStorm, etc.).
- Posséder des compétences de base en ligne de commande et en édition de fichiers.
- Expérience du travail en équipe : revue de code, workflows collaboratifs.
- Une sensibilité à l'IA générative et au prompt engineering est un atout pour maximiser les bénéfices de la formation.
- Un ordinateur portable standard (16 Go recommandé) avec les droits pour installer des logiciels.
- Système d'exploitation récent (Windows 10+, macOS 10.15+, Linux).
- Git installé et configuré avec accès à GitHub/GitLab.
- Version récente de NodeJS et npm installés et configurés.
- Un IDE au choix (VS Code, IntelliJ, WebStorm...).
- Une connexion internet stable.

Après cette formation, nous vous conseillons le(s) module(s) suivant(s):

Aucun

Contenu:

01- Introduction à l'IA pour Développeurs

- Rappels sur l'IA: Rappel rapide, histoire de l'IA / ML / NLP / IA Générative, arrivée de ChatGPT/LLM
- Modèles du marché : GPT-5, Claude, Gemini - différences clés
- Art du prompting : comment bien communiquer avec l'IA pour obtenir du code de qualité
- Évolution des outils : de l'assistance ponctuelle aux agents collaboratifs
- Contexte et tokens : comprendre les contraintes
- Panorama des outils : ChatGPT, Claude, etc.
- Prompt engineering : comparaison de 3 approches sur un cas concret

02- Vibe Coding vs Agentic Coding

- Vibe Coding : définition, limites et dangers
- Agentic Coding : vision du développeur augmenté
- Collaboration continue vs génération passive
- Impact sur qualité, maintenabilité et documentation
- Mise en place de l'outil de vibe coding
- Expérimentation guidée : vibe coder une application sans regarder le code généré et expliquer les résultats, les contraintes, les avantages

03- Découverte et pratique de l'Agentic Coding

- Introduction pratique : qu'est-ce que l'Agentic Coding pour un développeur ?
- Démonstration live : développement traditionnel vs développement augmenté
- Focus qualité : comment l'IA peut améliorer la qualité du code (conventions, patterns, best practices)
- Workflow agentic : Specify ? Plan ? Tasks ? Implement ? Validation
- Gestion du contexte : le fichier .md de contexte efficace (en local pour le moment)
- Structuration des demandes et itérations
- Bonnes pratiques : documentation, tests, qualité
- Scénario 1: legacy - Refactoring legacy : prise en main du projet pour qu'il soit Agentic Coding compliant. Aller jusqu'au plan de migration
- Scénario 2: from scratch et évolutions - Projet from scratch : architecture, choix techniques, structuration; Développement d'une fonctionnalité complète avec le workflow agentic

04- Deep Dive sur les outils agentiques

- Les points suivants sont abordés et pratiqués en fonction des capacités réelles de l'outil agentique choisi (Claude Code, GitHub Copilot, Gemini CLI, Codex CLI, etc.)
- Tour d'horizon de l'outil
- Fonctionnalités clés, commandes et raccourcis CLI pour chaque outil
- Agents personnalisés : création de sous-agents spécialisés avec contexte isolé pour des workflows ciblés
- Commandes / prompts personnalisés : templates de prompts réutilisables pour les tâches récurrentes
- Agent skills : connaissances composables et chargées à la demande
- Hooks : commandes shell déclenchées à des événements du cycle de vie
- Plugins : packages distribuables regroupant agents, skills, hooks et serveurs MCP
- Fichiers de configuration
- Fichier de contexte
- Choisir le bon type de fonctionnalité pour chaque scénario
- Deep dive sur l'outil de son choix : créer des agents personnalisés, commandes, skills, hooks, plugins, etc.

05- Utilisation de serveurs MCP

- Model Context Protocol (MCP) : concepts et architecture
- Installation et utilisation de clients MCP (exemples : playwright, context7, Atlassian...)
- Cas d'usage concret : MCP Playwright pour faciliter les tests avec visibilité des outputs dans le navigateur
- Installation d'un MCP et interaction complexe : capture d'écran + génération de tests

06- Développement Augmenté en Équipe

- Utilisation des skills d'agent en contexte d'équipe : apprendre à exploiter les skills réutilisables existants pour appliquer de manière cohérente les règles d'architecture, les standards de codage et les pratiques de test à travers toute l'équipe. (Claude/Copilot/Gemini)
- Exploitation des agents personnalisés pour les rôles : savoir interagir avec les agents spécifiques aux rôles (par ex. Architecture Guardian, Test Strategist, API Contract Agent) pour obtenir des conseils contextuels adaptés à vos tâches. (Claude/Copilot)
- Exécution des commandes personnalisées : pratique de l'invocation

07- Risques, Responsabilités et Perspectives

- L'importance du human-in-the-loop
- Maintien des compétences en développement classique et perte de connaissance technique: mythe ou réalité ?
- Dépendance aux outils IA : risques et mitigation
- Sécurité : code review, vulnérabilités, données sensibles
- Propriété intellectuelle et conformité
- Éthique et responsabilité du développeur augmenté
- Perspectives d'évolution

08- Examen de certification SFEIR Certified AI-Augmented Developer

- Examen en ligne (type QCM) comportant 20 questions pratiques couvrant les thèmes abordés pendant la formation. Score de passage minimum: 80%.

des slash commands personnalisées (par ex. /code-reviewer, /prepare-pr, /trace-flow) pour accélérer la revue de PR, l'analyse des diffs et la planification des tests, etc. (Claude/Copilot/Gemini)
- Utilisation de plugins marketplace : installer et utiliser des plugins Claude Code depuis des marketplaces pour étendre les fonctionnalités (commandes, agents, skills, hooks et serveurs MCP). (Claude)
- Configuration des custom instructions : définir des règles repository-wide et path-specific pour guider l'IA. (Copilot)
- Utilisation des hooks pour des gardes automatisées : appliquer des hooks pre-commit et post-edit pour le quality gate. (Claude)
- Utilisation des hooks pour l'automatisation : utiliser les hooks Gemini (par ex. BeforeTool, AfterTool, BeforeAgent) pour validation et automation. (Gemini)
- Accès aux connaissances du dépôt : explorer docs d'architecture, standards, glossaire métier. (Claude/Copilot/Gemini)
- Revue de code collaborative assistée par l'IA : produire des commentaires cohérents et actionnables. (Claude/Copilot/Gemini)
- Analyse des diffs et gestion des risques en pratique : interpréter les recommandations sur changements à risque. (Claude/Copilot/Gemini)
- Hands-on : exécuter /code-reviewer sur un diff de PR et analyser les recommandations. (Claude/Copilot/Gemini)
- Hands-on : interagir avec l'agent Architecture Guardian pour vérifier la conformité architecturale. (Claude/Copilot)
- Hands-on : interagir avec le skill topology-awareness pour évaluer les changements. (Claude/Copilot/Gemini)
- Hands-on : installer un plugin depuis un marketplace et utiliser ses commandes/agents/skills. (Claude)
- Hands-on : utiliser /prepare-pr pour générer une description de PR et une checklist. (Claude/Copilot/Gemini)
- Hands-on : exécuter /trace-flow sur un endpoint backend et examiner l'explication générée. (Claude/Copilot/Gemini)
- Hands-on : observer les hooks pre-commit et post-edit en action sur des changements d'exemple. (Claude)
- Hands-on : observer les hooks Gemini (par ex. BeforeTool, AfterTool, BeforeAgent) en action. (Gemini)
- Hands-on : configurer les instructions partagées et observer l'impact sur les recommandations. (Copilot)
- Hands-on : consulter les références

partagées lors d'une commande/agent pour améliorer le contexte. (Claude/Copilot/Gemini)
- Hands-on : analyser un diff à risque et discuter de la décision de merge. (Claude/Copilot/Gemini)

Méthodes pédagogiques :

Support de cours officiel Google en anglais remis aux participants

Autres moyens pédagogiques et de suivi:

• Compétence du formateur : Les experts qui animent la formation sont des spécialistes des matières abordées et ont au minimum cinq ans d'expérience d'animation. Nos équipes ont validé à la fois leurs connaissances techniques (certifications le cas échéant) ainsi que leur compétence pédagogique.
• Suivi d'exécution : Une feuille d'émargement par demi-journée de présence est signée par tous les participants et le formateur.
• En fin de formation, le participant est invité à s'auto-évaluer sur l'atteinte des objectifs énoncés, et à répondre à un questionnaire de satisfaction qui sera ensuite étudié par nos équipes pédagogiques en vue de maintenir et d'améliorer la qualité de nos prestations.

Délais d'inscription :

• Vous pouvez vous inscrire sur l'une de nos sessions planifiées en inter-entreprises jusqu'à 5 jours ouvrés avant le début de la formation sous réserve de disponibilité de places et de labs le cas échéant.
• Votre place sera confirmée à la réception d'un devis ou """"booking form""""" signé. Vous recevrez ensuite la convocation et les modalités d'accès en présentiel ou distanciel.
• Attention, si cette formation est éligible au Compte Personnel de Formation, vous devrez respecter un délai minimum et non négociable fixé à 11 jours ouvrés avant le début de la session pour vous inscrire via moncompteformation.gouv.fr.

Accueil des bénéficiaires :

• En cas de handicap : plus d'info sur globalknowledge.fr/handicap
• Le Règlement intérieur est disponible sur globalknowledge.fr/reglement