

Développeur Augmenté par l'IA

Formation Agentic Coding : augmentez la productivité de vos développeurs avec l'IA. Code fiable, automatisation, refactoring rapide et certification SFEIR

2 jours / 14h

Présentation du cours

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,
- 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. A 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.

A la fin de cette formation, l'évaluation finale sous la forme d'un test de certification vous permettra d'obtenir la certification SFEIR Certified AI-Augmented Developer.

Méthodes mobilisées : Ce cours alterne parties théoriques sous forme de lectures (slides), démos et parties pratiques sous forme de labs dirigés

Objectifs pédagogiques

À la fin de la formation, les participants seront capables de :

- **Accélérer toutes les phases du cycle de développement**, de la conception architecturale jusqu'à la résolution de bugs complexes, en augmentant leur vitesse 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 homogénéisant les workflows.

En résumé : les participants ressortiront avec la capacité de **produire plus vite, mieux, et ensemble**, et donneront un véritable **avantage compétitif** à leur organisation.

Modalités d'évaluation : Les objectifs pédagogiques sont évalués à travers la réalisation des parties pratiques (labs dirigés) sous la supervision du formateur délivrant la session de formation.

Public cible

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.

Prérequis

Connaissances

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

Outils

- Une ordinateur portable standard (16 Go recommandé) avec les droits pour installer des logiciels
- Une connexion internet stable
- 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...)
- Docker (optionnel mais très fortement recommandé pour bénéficier d'un setup automatisé via notre devcontainer)

Programme

Jour 1 – Fondamentaux et Outils d'Agentic Coding

Module 1 : 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.

Ateliers :

- Prompt engineering : comparaison de 3 approches sur un cas concret

Module 2 : 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

Ateliers :

- Mise en place de l'outils de vibe coding
- Expérimentation guidée : vibe coder une application sans regarder le code généré et expliquer les résultats, les contraintes, les avantages

Module 3 : 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

Ateliers :

- 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

Jour 2 – Outils, Extensions, MCP

Module 4 : Évaluer, comparer et combiner les solutions IA pour maximiser la productivité

- Gestion des tokens
- IDE nouvelle génération (ex Cursor AI)
- Comparaison pratique : quel outil pour quel besoin ?
- Workflow hybride : combiner plusieurs assistants IA
- Matrice de possibilité d'utilisation des outils : AI coding agent tool (Github Copilot / Claude Code / Open Code / Gemini CLI) versus modèles (Sonnet, Gemini, ...)
- Gestion des licences / clés
- Démonstrations

Module 5 : Extensions et 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
- Spécificités Claude Code : les sub-agents, les Hooks, les Skills...

Ateliers :

- Installation d'un MCP et interaction complexe : capture d'écran + génération de tests

Module 6 : Développement en Équipe Augmentée

- Standards d'équipe : contexte et instructions partagées (ex AGENTS.md)
- Standards partagés : définir des conventions d'équipe – plugins, marketplace Claude Code, partage de hooks et de configuration
- Sécurité des agents : configurer ses settings (.env non partagés) et ses agents selon leurs rôles
- Partage de prompts et patterns réutilisables
- Onboarding accéléré des nouveaux développeurs
- Revue de code en équipe augmentée
- Intégration CI/CD et automatisation

- Collaboration homme-agent-équipe : review assistée, pair programming augmenté

Ateliers :

- Création de plusieurs AGENTS.md et définition du contexte d'équipe
- Simulation de workflow collaboratif : feature development → code review → merge
- Mise en place d'un workflow Git avec agents (pre-commit hooks, automated reviews)
- Développement collaboratif : plusieurs devs + agents sur le même projet (avec conflits, résolution)
- Simulation d'équipe : onboarding d'un nouveau développeur avec assistance IA

Module 7 : 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

Module 8 : 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%.