

## Maîtriser le code augmenté par l'intelligence artificielle avec le vibe coding et le SDLC 2.0

Date et durée
Code formation : DEV019FR Durée : 4 jours Nombre d'heures : 28 heures
Description
<p>L'intelligence artificielle générative transforme radicalement l'<b>ingénierie logicielle</b>, faisant évoluer le développeur vers un rôle d'orchestrateur. Cette formation intensive de 4 jours vous permet de maîtriser le <b>code augmenté</b> et d'intégrer l'IA à chaque étape du cycle de vie logiciel (SDLC), de l'analyse des besoins au déploiement.</p> <p>Le programme introduit la méthodologie du <b>vibe coding</b>, où vous apprendrez à générer, réviser et assembler du code professionnel en supervisant l'IA. Vous manipulerez les outils de pointe, tels que <b>Microsoft Copilot, Windsurf, Lovable ou Google Antigravity</b> et explorerez le potentiel des <b>serveurs MCP</b> (Model Context Protocol) pour connecter vos modèles à des systèmes réels.</p> <p>Au-delà de la génération de code, la formation insiste sur l'excellence opérationnelle : <b>clean code, sécurité</b> (DevSecOps augmenté) et gouvernance des prompts. Le parcours se conclut par la réalisation d'un projet complet "from scratch", incluant architecture, tests et pipeline CI/CD générés par IA, vous rendant opérationnel pour le <b>SDLC 2.0</b>.</p>
Objectifs
<p>À l'issue de cette formation, vous atteindrez les objectifs de compétences suivants :</p> <ul style="list-style-type: none"><li>• comprendre l'impact de l'IA générative sur l'ingénierie logicielle et son intégration dans le SDLC ;</li><li>• pratiquer le vibe coding pour orchestrer la production de code et superviser la qualité technique ;</li><li>• maîtriser les outils d'assistance (GitHub Copilot, Windsurf, MCP) pour accélérer le développement ;</li><li>• appliquer les bonnes pratiques de clean code, de sécurité et de gouvernance dans un contexte d'IA ;</li><li>• concevoir et déployer un projet complet augmenté par IA, incluant tests et pipelines CI/CD.</li></ul>
Points forts
<ul style="list-style-type: none"><li>• <b>Méthodologie avant-gardiste</b> : vous adopterez le vibe coding, la nouvelle posture du développeur orchestrateur.</li><li>• <b>Tech stack moderne</b> : vous manipulerez les derniers outils, comme Windsurf, Lovable et le protocole MCP.</li><li>• <b>Approche DevSecOps</b> : vous ne sacrifierez pas la qualité au profit de la vitesse, en intégrant sécurité et clean code.</li><li>• <b>Projet intégral</b> : vous réaliserez de A à Z une application augmentée, validant concrètement vos nouvelles compétences.</li></ul>
Modalités d'évaluation

Travaux Pratiques
Pré-requis
<p><i>Suivre cette formation nécessite les prérequis suivants :</i></p> <ul style="list-style-type: none"> <li>• <b>Expérience technique</b> : une maîtrise d'au moins un langage de programmation moderne (JS, Python, Java ; etc.).</li> <li>• <b>Outils</b> : une connaissance de Git, des workflows de développement et de bonnes connaissances avec les IDE/terminaux.</li> <li>• <b>Concepts</b> : une compréhension générale du cycle de développement logiciel (SDLC).</li> </ul>
Public
<p><i>Cette formation s'adresse aux profils techniques expérimentés. Le public inclut notamment :</i></p> <ul style="list-style-type: none"> <li>• <b>les développeurs confirmés, Tech Leads et architectes logiciels</b> qui souhaitent évoluer vers un rôle d'orchestrateur et maîtriser le vibe coding ;</li> <li>• <b>les ingénieurs DevOps et SRE</b> qui désirent intégrer l'IA dans les pipelines CI/CD et automatiser les tâches d'infrastructure ;</li> <li>• <b>les ingénieurs QA et testeurs</b> qui veulent automatiser la génération de tests et de scénarios de validation ;</li> <li>• <b>les chefs de projet techniques</b> qui cherchent à comprendre les impacts de l'IA sur le cycle de vie logiciel pour mieux piloter les équipes.</li> </ul>
Programme
<p><b>Jour 1 : s'approprier les fondamentaux et transformer le SDLC</b></p> <ul style="list-style-type: none"> <li>• L'impact de l'IA sur l'ingénierie logicielle : assistance vs génération autonome.</li> <li>• L'évolution du SDLC classique vers un modèle augmenté (requirements, design, tests).</li> <li>• Les bénéfices attendus : vélocité, sécurité et réduction de la dette technique.</li> </ul> <p><b>Travaux pratiques</b></p> <ul style="list-style-type: none"> <li>• Générer des user stories et critères d'acceptation à partir d'un besoin flou.</li> <li>• Utiliser l'IA pour détecter des incohérences et générer un diagramme d'architecture (UML/C4).</li> </ul> <p><b>Jour 2 : maîtriser le vibe coding et le prompting avancé</b></p> <ul style="list-style-type: none"> <li>• Le concept de vibe coding : le développeur comme orchestrateur d'intentions.</li> <li>• Les techniques de prompting pour développeurs : intention, technique, audit et tests.</li> <li>• La méthodologie complète : définition, génération guidée, revue et assemblage.</li> </ul> <p><b>Travaux pratiques</b></p> <ul style="list-style-type: none"> <li>• Générer une API complète avec tests unitaires et scénarios QA.</li> <li>• Réaliser un refactoring automatisé et ajouter de la gestion d'erreurs via l'IA.</li> </ul> <p><b>Jour 3 : automatiser le développement avec le protocole MCP</b></p> <ul style="list-style-type: none"> <li>• Le fonctionnement des serveurs MCP (Model Context Protocol) et leur architecture.</li> <li>• Le workflow IA + MCP : interaction avec fichiers, repos Git et bases de données.</li> </ul>

- Les cas d'usage : analyse de code, documentation automatique et pipelines CI/CD.

### **Travaux pratiques**

- Manipuler un projet réel via un serveur MCP (création, modification).
- Générer et auditer un workflow CI/CD complet et automatiser le changelog.

## **Jour 4 : sécuriser les pratiques et finaliser le projet**

- Le clean code à l'ère de l'IA : imposer un style et documenter l'intention.
- Le DevSecOps augmenté : scans de sécurité et réduction de la dette technique par l'IA.
- La gouvernance : gestion des prompts, traçabilité et conformité.

### **Travaux pratiques**

- Projet final : concevoir et développer une mini-application opérationnelle complète (architecture, code, tests, documentation, CI/CD) en utilisant le vibe coding et les outils étudiés.

*Les noms de marques et logos éventuellement cités dans cette fiche (ex. Microsoft Copilot, GitHub, Windsurf, Lovable) sont la propriété de leurs détenteurs respectifs.*

*Leur mention à des fins pédagogiques ne constitue ni un engagement ni un partenariat.*