

retengr

Développement et Ingénierie logicielle assistés par l'IA : Du Spec-Driven Development à l'orchestration d'agents

Durée : 2 jours (14 heures)



Méthode pédagogique

Composée à 50% de pratique, cette formation utilise des exercices illustrés et didactiques, une mise en situation sur des cas pratiques réels.

Chaque module théorique est suivi d'un lab ou d'une démonstration sur des cas réels. Les stagiaires manipulent les outils, écrivent des spécifications, orchestrent des agents et implémentent un serveur MCP (ils ne sont pas uniquement en mode observateur).

Une évaluation régulière de l'acquisition des connaissances de chaque chapitre est effectuée auprès de chaque participant.

Une synthèse est proposée en fin de formation.

Une évaluation à chaud sera proposée au stagiaire à la fin du cours.

Un support de cours sera remis à chaque participant comprenant les slides sur la théorie, les exercices et leurs corrigés.

Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

Un suivi et un échange avec les participants seront mis en place quelques jours après la formation.

Cette formation peut être animée en présentiel ou en distanciel (classe virtuelle avec formateur).

Elle peut être organisée en journées non consécutives, en laissant du temps entre chaque jour de formation (quelques jours/semaines/mois de pratique terrain) pour une meilleure montée en compétences.

Présentation

Cette formation de 2 jours couvre l'intégralité du spectre de l'ingénierie logicielle assistée par l'IA : de la rédaction de spécifications à l'orchestration d'agents, en passant par la sécurité, la gestion du contexte, l'intégration CI/CD et l'intégration de l'IA sur des projets existants. Vos



équipes repartiront avec des méthodes, et outils directement applicables à leurs projets.

C'est une approche ancrée dans le Spec-Driven Development, mais qui va bien au-delà. Agents autonomes, orchestration multi-agents, MCP, CI/CD augmentée, gestion de la dette technique, modèles locaux : chaque dimension de l'ingénierie logicielle moderne est couverte, avec pour fil rouge constant la question de la qualité et de la maîtrise.

Est inclus dans ce cours un module unique sur les modèles locaux et la confidentialité, pensé pour les environnements soumis à des contraintes réglementaires fortes : médical, défense, juridique, finance...etc

Une clôture lucide sur la productivité réelle de l'IA, avec des données indépendantes et un cadrage honnête sur les risques (atrophie des compétences, dette technique, hallucinations) pour que vos équipes restent les pilotes de leur code.



Objectifs

- Maîtriser le workflow Spec-Driven Development. De l'idée initiale à la mise en production, en passant par la spécification, la planification et l'implémentation assistée.
- Produire des spécifications exploitables par les agents. Formats BDD, EARS et ADR pour des specs structurées, testables et directement consommables par un agent IA.
- Tester avec et grâce à l'IA. TDD assisté, génération de suites de tests depuis des specs BDD, mutation testing pour valider la solidité des tests.
- Instruire l'IA sur l'architecture de vos projets. Encoder les décisions architecturales, définir une constitution, contraindre l'agent à respecter la structure cible.
- Gouverner l'IA au niveau de l'équipe. Fichiers de règles persistants, standards partagés, onboarding accéléré des nouveaux développeurs.
- Identifier et maîtriser les risques. Vulnérabilités du code IA, hallucinations, slopsquatting, propriété intellectuelle.
- Choisir les bons outils et modèles. Panorama du marché, matrice outil par cas d'usage, coûts réels et arbitrages selon votre contexte.

Audience

Développeurs, Tech Lead, Architectes, DevOps



Le formateur

Un regard d'expert qui va au-delà de l'IA.

Nos formateurs sont des praticiens seniors en développement, architecture et conception logicielle. Les stagiaires bénéficient de conseils concrets sur la qualité du code, les choix architecturaux et les bonnes pratiques, indépendamment de l'IA.

Pré-requis

Pas de prérequis demandé.

Afin de valider les profils, les compétences et le niveau de connaissance de chaque participant, en amont de la formation, le formateur organise un entretien téléphonique ou visio. Il confirme alors que le participant a le niveau nécessaire et que le contenu répond bien à ses attentes.

Programme

Jour 1 — Panorama et fondamentaux [1h]

- Histoire de l'IA pour le code : de l'autocomplétion à l'agent autonome
- Modèles du marché : GPT-5, Claude, Gemini, Mistral et leurs différences clés
- Comprendre les LLM : tokens, fenêtre de contexte, température, limites
- Vibe coding : définition, pourquoi ça séduit, pourquoi ça ne tient pas en production



Jour 1 - Prompt engineering [1h]

- Anatomie d'un bon prompt : rôle, contexte, instruction, format de sortie
- Itérer plutôt qu'espérer : la boucle de raffinement
- Pièges classiques : ambiguïté, sur-spécification, contradictions

Jour 1 - De l'idée à la mise en production [1.5h]

- Le workflow SDD complet : Brainstorm, Spec, Plan, Tasks, Implement, Validate
- Les trois phases : spécifier, planifier, implémenter avec le rôle humain à chaque étape
- Outils de la chaîne : LLM conversationnel, modèle de raisonnement, agent de code
- GitHub Spec Kit : les 6 commandes /specify, /clarify, /plan, /tasks, /implement, /constitution

Jour 1 - Écrire de bonnes spécifications [1.5h]

- Ce qu'une spec doit contenir : formats entrée/sortie, règles métier, cas limites, contraintes d'intégration, sécurité
- Séparer le QUOI du COMMENT : pourquoi c'est non-négociable
- Formats efficaces : Markdown structuré, BDD/Gherkin (Given/When/Then), ADR, notation EARS
- Spec fonctionnelle versus détails techniques : où tracer la ligne

Jour 1 - Tests assistés par IA [1h]

- TDD assisté par IA : le LLM comme partenaire du cycle red/green/refactor
- Specs BDD comme tests exécutables : le lien naturel Given/When/Then vers test code
- Génération de tests unitaires, intégration et e2e : ce que l'IA détecte bien
- Mutation testing assisté : vérifier la solidité des tests générés



Jour 1 - Instruire l'IA sur l'architecture logicielle [1h]

- Encoder les décisions architecturales : ADR comme source de vérité pour l'agent
- La notion de constitution : invariants, patterns imposés, anti-patterns interdits
- Contraindre l'IA à respecter la structure cible : couches, frontières, conventions de nommage
- Cas pratique : clean architecture, hexagonale, microservices

Jour 2 — Conventions, règles persistantes et standards d'équipe [1h]

- Fichiers de règles persistants : CLAUDE.md, .cursor/rules/, AGENTS.md (standard Linux Foundation)
- Les 4 modes d'activation Cursor : Always, Auto, Agent Requested, Manual
- Divulgarion progressive : dire à l'IA comment trouver l'info plutôt que tout inclure
- SKILL.md : encapsuler des capacités réutilisables, distinction règles (contraindre) vs skills (étendre)
- Standards d'équipe : prompts réutilisables, patterns partagés, onboarding accéléré

Jour 2 - Introduction aux agents [1h]

- La boucle agentique : plan, agir, observer, affiner
- Agent unique : configuration, portée des tâches, supervision
- Quand confier une tâche à un agent et quand garder la main
- Horizons temporels : ce qu'on peut attendre d'un agent en quelques minutes

Jour 2 - Gestion du contexte et économie de tokens [1h]

- La fenêtre de contexte : évolution, enjeux, le problème du "lost in the middle"
- Structurer ses fichiers et prompts pour maximiser la pertinence de la réponse



- RAG pour le code : indexation, chunking syntaxique, scoring sémantique
- Économie de tokens : caching, divulgation progressive, découpage des tâches

Jour 2 - Choix des modèles et des outils [1h]

- Panorama : Cursor, Claude Code, GitHub Copilot, Windsurf, Aider, Kiro, Antigravity
- Matrice outil par modèle par cas d'usage : autocomplétion, refactoring, debugging, architecture
- Coûts réels : licences, consommation API, TCO selon profil d'usage
- Quand combiner les outils : par exemple flow et le raisonnement profond

Jour 2 - Sécurité et hallucinations [1.5h]

- Vulnérabilités du code IA : 3 fois plus que le code humain, XSS 2,74 fois plus fréquent
- Hallucinations et slopsquatting : packages fictifs recommandés par les LLM, exploitables par des attaquants
- Rules File Backdoor : injection de code malveillant via fichiers de configuration IA
- Propriété intellectuelle et ce qu'il n'est pas recommandé de déléguer

Jour 2 - Greenfield vs brownfield [1h]

- Adopter le SDD sur un projet existant : audit du codebase, génération du CLAUDE.md initial, migration progressive
- Rendre un projet agentic-coding compliant : structurer pour que l'IA comprenne le contexte
- Plan de migration legacy : de l'analyse à la roadmap de refactoring assisté par agent
- Pièges spécifiques au brownfield : dette existante amplifiée, tests absents, architecture implicite



Jour 2 - Lucidité sur la productivité [0.5h]

- Le paradoxe perception/réalité : +20% ressenti vs -19% réel (étude METR) pour calibrer ses attentes
- Atrophie des compétences : maintenir sa capacité de raisonnement sur le code sans IA
- Éthique et responsabilité : le développeur reste le signataire du code
- Synthèse : du codeur à l'orchestrateur, la compétence clé est de penser clairement ce qu'il faut construire

Modalités et délais d'accès à la formation

Les inscriptions sont possibles jusqu'à 48 heures ouvrées avant le début de la formation, en interentreprises, dans la limite des places disponibles.

Pour les formations organisées en intra entreprise, la liste des participants peut être modifiée jusqu'à 24h ouvrés avant le début de la formation.

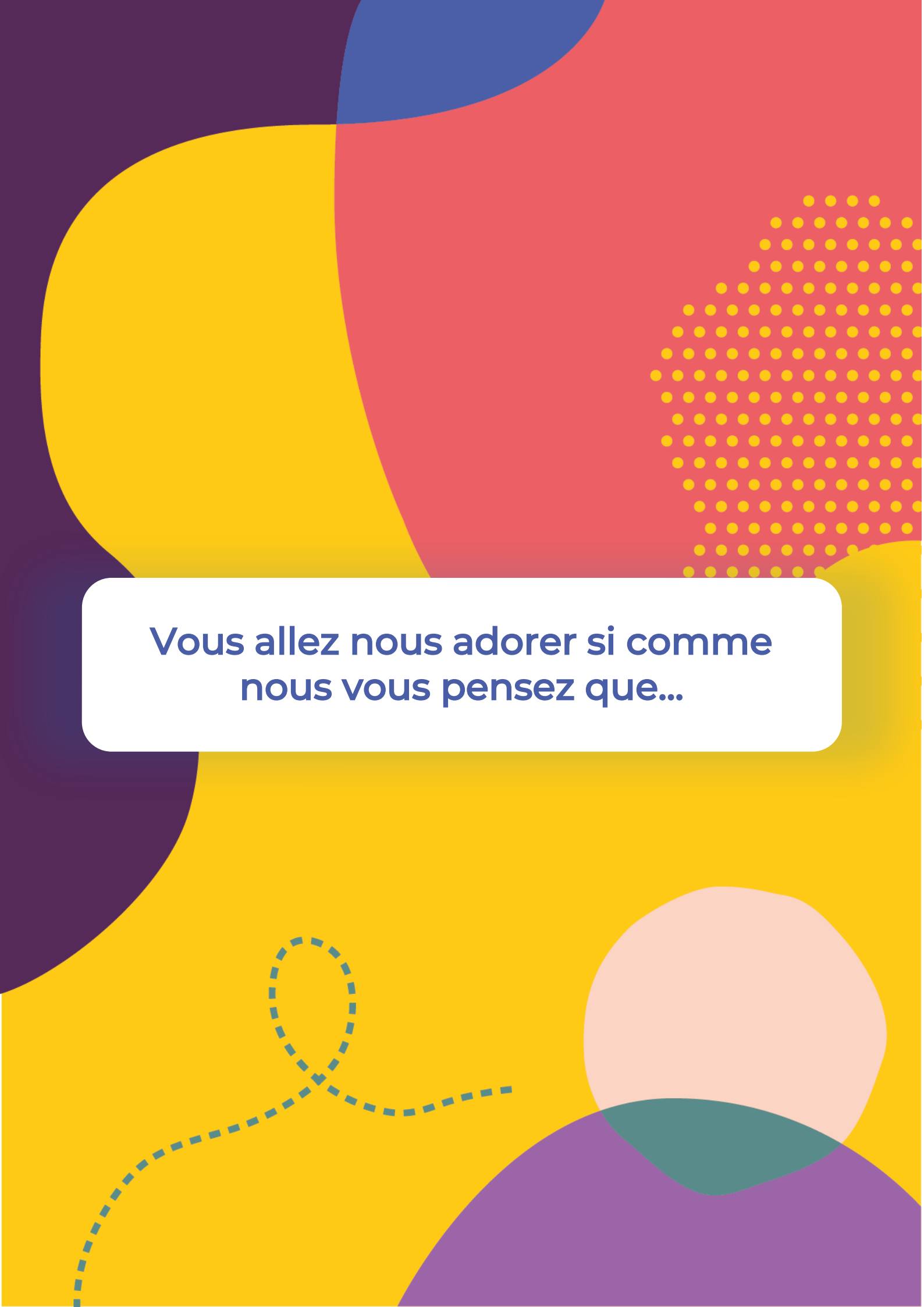
Accessibilité

RETENGR facilite l'accessibilité de ses formations.

Cette formation est accessible aux personnes en situation de handicap.



Si vous avez un besoin d'accès spécifique, contactez Céline BOURREIL (celine.bourreil@retengr.com) qui étudiera avec Handifiel's (notre référent handicap) votre demande et vous proposera les meilleures solutions.

The background is a vibrant abstract composition. It features large, overlapping organic shapes in shades of purple, yellow, and pink. A semi-circular area on the right side is filled with a dense pattern of small yellow dots. In the lower-left quadrant, there is a dashed teal line that forms a loop and then extends outwards. The overall aesthetic is modern and playful.

**Vous allez nous adorer si comme
nous vous pensez que...**

Une formation doit être au service de la performance du collaborateur et de l'entreprise

Ceci nécessite une quête constante d'excellence de la part de l'organisme formateur avec une adaptation systématique aux enjeux de l'entreprise, la mise à jour régulière des supports de cours et une veille technologique indispensables pour toujours être à la pointe du domaine.



L'expertise technique est aussi importante que les qualités pédagogiques



Nos formateurs sont tous des experts de leur domaine. Mais qu'ont-ils de plus que les autres ? Nous les sélectionnons en plus pour leurs qualités de pédagogue et leurs méthodes d'enseignements. Nous plaçons les qualités pédagogiques au même niveau que l'expertise afin que nos stagiaires tirent le meilleur de leurs formations.

L'excellence naît de l'excellence

Beaucoup de nos clients se classent parmi les leaders de leurs industries respectives ou parmi les start-ups les plus prometteuses. Nous savons que former les collaborateurs de telles entreprises nécessite de prêter attention à chaque détail en prodiguant un accompagnement à la hauteur de l'ambition de nos stagiaires. C'est pourquoi nous savons faire des leaders d'aujourd'hui les champions de demain !





retengr

Faire du leader d'aujourd'hui, le
champion de demain