

Formation Claude Code avancé : industrialiser vos workflows de développement agentique

Réf. SF34748

Passez d'un usage individuel de Claude Code à des workflows d'équipe fiables, partagés et automatisés, du poste du développeur jusqu'à la CI/CD.

2 j
Durée

1 700 € HT
Inter / participant

0
Sessions à venir

Objectifs pédagogiques

- ✓ Concevoir l'architecture de contexte d'un projet (CLAUDE.md hiérarchisé, règles ciblées, mémoire) pour fiabiliser les résultats et réduire la consommation de tokens
- ✓ Appliquer des patterns de prompting avancé (plan préalable, Best-of-N, auto-critique) pour fiabiliser les résultats sur des tâches complexes
- ✓ Développer des skills et des plugins pour capitaliser les pratiques d'une équipe et les distribuer
- ✓ Configurer des hooks déterministes pour garantir la qualité et la sécurité des actions de l'agent
- ✓ Intégrer des serveurs MCP authentifiés et développer un serveur MCP local simple
- ✓ Orchestrer des subagents spécialisés pour déléguer et paralléliser des tâches complexes
- ✓ Automatiser des workflows Claude Code en CI/CD et avec l'Agent SDK, en maîtrisant la sécurité et les coûts

Public visé & prérequis

Public visé Développeurs, Ingénieurs DevOps

Prérequis- Avoir suivi la formation « Claude Code : automatiser vos tâches de développement » ou avoir une pratique régulière équivalente (CLAUDE.md, commandes, permissions, workflow de développement assisté) - Pratique de Python ou de TypeScript - Maîtrise de Git et de GitHub ou GitLab - Notions de CI/CD

Programme détaillé

Architecturer le contexte d'un projet

- Grille de lecture : cartographier ses usages de Claude Code et repérer ce qui consomme du contexte sans apporter de qualité
- Diagnostic d'un CLAUDE.md qui dérive : fichier monolithique, consignes contradictoires, informations obsolètes
- Hiérarchie de mémoire : organisation, projet, utilisateur, local et sous-dossiers, ordre de chargement
- Imports et documentation par préoccupation : CLAUDE.md comme index, détail chargé à la demande
- Règles ciblées par chemin : instructions limitées à certaines parties du code (`.claude/rules/`)
- Mémoire automatique : ce que l'agent retient seul, comment la relire et la maîtriser
- Mesure de l'empreinte du contexte : identifier ce qui consomme, comparer avant et après restructuration

Activités pratiques :

- Restructuration du CLAUDE.md de son dépôt en index et règles ciblées, avec mesure de l'empreinte avant et après
- Création de règles propres à un dossier (tests, front-end, migrations) et vérification de leur chargement

Fiabiliser les résultats : prompting avancé et patterns de qualité

- Gabarits de demande : rôle, contexte, contraintes, format de sortie, critère de fin
- Exemples guidés : montrer le résultat attendu pour stabiliser le style et la structure
- Décomposition et enchaînement : découper une tâche complexe en étapes vérifiables
- Plan d'abord : exiger un plan avant toute exécution, le challenger et le faire réviser
- Raisonnement étendu : quand demander plus de réflexion, et à quel coût
- Best-of-N et auto-critique : générer plusieurs solutions, les comparer, faire relire le résultat par un second passage ou un second agent

- Diagnostic d'un prompt qui dérive et capitalisation : transformer une formulation éprouvée en skill

Activités pratiques :

- Réécriture guidée d'un prompt complexe et comparaison des résultats
- Best-of-N sur une décision de conception, avec grille de comparaison des solutions

Skills, commandes et plugins d'équipe

- Skills et commandes personnalisées : un format unifié, les commandes historiques restant compatibles
- Conception d'un skill : description qui le déclenche au bon moment, arguments, invocation manuelle ou automatique
- Skills outillés : scripts et ressources embarqués, outils autorisés, exécution isolée dans un subagent
- Bibliothèque de skills : portée projet ou utilisateur, conventions de nommage, maintenance
- Plugins : regrouper skills, agents, hooks et serveurs MCP dans un paquet installable
- Marketplace d'équipe : publication, installation, mises à jour et versions
- Déploiement en entreprise : paramètres gérés par l'organisation, plugins et serveurs autorisés
- Audit d'un plugin ou d'un skill tiers : ce qu'il exécute, ce à quoi il accède, risques

Activités pratiques :

- Création d'un skill outillé (script embarqué) pour une tâche récurrente de l'équipe
- Packaging de skills en plugin, publication dans une marketplace d'équipe et installation par un autre participant
- Audit d'un plugin tiers avant son adoption

Hooks : garantir un comportement déterministe

- Pourquoi des hooks : une consigne du prompt peut être ignorée, un hook s'exécute toujours
- Cycle de vie : avant et après l'utilisation d'un outil, soumission d'un prompt, début et fin de session, fin de tâche, notifications
- Types de hooks : commandes shell, appels HTTP, contrôles confiés à un modèle
- Blocage avant action : interdire les commandes dangereuses et l'écriture dans les fichiers protégés
- Qualité après édition : formatage, lint et tests lancés automatiquement, erreurs renvoyées à l'agent
- Contrôle de fin de tâche : empêcher l'arrêt tant que les tests ne passent pas
- Contexte et traçabilité : injection d'informations de session, journal d'audit des actions, notifications de fin de tâche vers la messagerie d'équipe
- Partage des hooks : configuration projet versionnée ou plugin

Activités pratiques :

- Hook de blocage des commandes destructrices et de protection des fichiers sensibles
- Chaîne qualité après édition (formatage, lint, tests) avec retour d'erreur à l'agent
- Journal d'audit des actions et notification de fin de tâche

MCP avancé : connecter et étendre

- Portées et partage : configuration locale, configuration projet versionnée dans le dépôt, configuration utilisateur
- Serveurs distants et authentification : OAuth, jetons, renouvellement des accès
- Gouvernance des accès : serveurs autorisés par l'organisation, permissions par outil, moindre privilège
- Coût en contexte : chargement des outils à la demande, nombre de serveurs actifs à limiter
- Cas d'usage en entreprise : dépôt de code, base de données, suivi d'erreurs, gestion de tickets, navigateur
- Architecture d'un serveur MCP : outils, ressources, prompts, transports local et HTTP
- Serveur MCP local simple : exposer un outil interne à Claude Code avec le SDK MCP
- Risques : données remontées dans le contexte, injection de prompt via les réponses d'un serveur

Activités pratiques :

- Connexion d'un serveur MCP authentifié et partage de sa configuration avec l'équipe
- Création d'un petit serveur MCP local exposant un outil interne (API maison, référentiel de données)
- Comparaison de la consommation de contexte selon le nombre de serveurs actifs

Subagents et orchestration multi-agents

- Rôle des subagents : contexte séparé, prompt spécialisé, outils et permissions propres
- Conception d'un subagent : description qui déclenche la délégation, outils restreints, choix du modèle, portée projet ou utilisateur
- Délégation automatique ou explicite : laisser l'agent principal choisir le spécialiste ou l'imposer
- Patterns d'orchestration : recherche en parallèle, chaînage (exploration, implémentation, revue, tests), isolation des opérations volumineuses
- Tâches en arrière-plan : lancer, suivre et récupérer le résultat des traitements longs

- Sessions parallèles avec git worktrees : plusieurs agents sur des branches isolées
- Agent teams : aperçu d'une fonctionnalité expérimentale de coordination entre agents
- Garde-fous : périmètre étroit, moindre privilège, limites de tours et de coût, vérification indépendante

Activités pratiques :

- Création de subagents spécialisés (revue de code, exécution des tests, audit de sécurité) aux outils restreints
- Pipeline sur une fonctionnalité : exploration, implémentation, revue et tests délégués
- Travail en parallèle sur deux branches avec des worktrees

Industrialiser : headless, CI/CD et Agent SDK

- Mode non interactif : prompt en entrée, sortie texte ou JSON structurée, intégration dans des scripts
- Claude Code dans GitHub Actions : revue automatique des pull requests, réponse aux mentions dans les issues et les PR
- Autres plateformes : intégration GitLab CI/CD et pipelines internes
- Sécurisation de l'exécution automatisée : secrets, permissions minimales, outils autorisés, contenus externes non fiables
- Agent SDK : quand sortir de la CLI, ce que le SDK reprend de Claude Code (outils, permissions, gestion du contexte)
- Premier agent avec l'Agent SDK en Python ou TypeScript : réutilisation des outils, hooks et serveurs MCP
- Stratégie de coûts : modèle par tâche, cache de prompt, hygiène du contexte, budgets et suivi en CI
- Assemblage de la configuration d'équipe : bilan des briques mises en place et démonstration croisée

Activités pratiques :

- Workflow GitHub Actions de revue automatique des pull requests
- Premier agent Agent SDK pour une tâche de maintenance (par exemple trier et résumer des issues)
- Mini-projet de synthèse : assemblage de la configuration d'équipe (contexte, plugin, hooks, MCP, subagents, CI) et démonstration croisée

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...

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