WealCome
Expertise en conception logicielle
BDD / TDD / DDD / Clean Architecture

PROGRAMME DE FORMATION

Document généré le 23/11/2026

Agents IA et Software Craft : la vitesse sans la dette — Monde Node/Typescript

Type d'action : **Action de formation** Durée totale : **14h**

Informations de session

Lieu : Visio par Zoom

Formateur : Michaël AZERHAD - Président de S.A.S.U WealCome et architecte / expert technique depuis plus de 20 ans

Dates et horaires

Date	Début	Fin	Durée
16/12/2026	09:15	12:15	3h
16/12/2026	13:00	17:00	4h
17/12/2026	09:15	12:15	3h
17/12/2026	13:00	17:00	4h

Description

Tout le monde produit vite avec l'IA. Peu de gens produisent vite, et longtemps.

Sur un backend Node.js / NestJS, l'IA livre en quelques minutes des controllers, des entités d'ORM et des services entiers. Sans Software Craft, ce qu'elle produit ne tient pas la distance :

- une logique métier diluée dans les services et couplée aux entités de l'ORM ;
- des tests écrits après coup, qui rassurent sans réellement protéger ;
- du code en trop, des hallucinations, des fonctionnalités que personne n'a demandées ;
- un contexte qui sature et des tokens qui explosent en allers-retours ;
- du code relu ligne à ligne, sans savoir vraiment le juger ;
- un projet qui bascule en legacy en quelques mois au lieu de quelques années.

Cette formation prend le problème à la racine.

En deux jours, vous menez un vrai projet de bout en bout : une spécification brute, mal ficelée, telle qu'un fondateur vous la donnerait, jusqu'à plusieurs Bounded Contexts qui tournent ensemble dans un monolithe modulaire NestJS.

Ce sont les agents IA qui produisent ; c'est vous qui décidez.

Technologies :

- TypeScript
- Node.js (version LTS courante)
- Monorepo pnpm workspaces
- NestJS / API REST
- Effect-ts
- TypeORM, Prisma ou Knex
- PostgreSQL
- Jest
- Testcontainers (Docker)
- Stryker (mutation testing)
- Kafka ou RabbitMQ
- Claude Code : agents et skills conçus par Michaël AZERHAD

Objectifs de la formation

À l'issue de cette formation, les participants auront acquis les connaissances et méthodes nécessaires pour :

- Comprendre et maîtriser les principes du Software Craft : TDD, BDD, DDD et Clean Architecture.
- Appréhender les approches de conception métier fondées sur l'Event Storming, les Bounded Contexts et le langage ubiquitaire.
- Comprendre les stratégies de tests automatisés, de refactoring et de mutation testing pour garantir la qualité du logiciel.
- Maîtriser les principes de conception d'une architecture modulaire, hexagonale et évolutive dans un environnement Node.js / TypeScript.
- Comprendre les patterns avancés de conception : CQRS, agrégats, cohérence transactionnelle et communication entre contextes métiers.
- Découvrir et maîtriser les principes d'orchestration d'agents IA appliqués au cycle de développement logiciel.
- Comprendre comment encadrer la génération de code par l'IA à travers les spécifications, les tests et les contraintes architecturales.
- Développer une approche critique permettant d'évaluer les décisions techniques proposées par les agents IA et de limiter la dette technique.

Prérequis

- Solide expérience du développement (au moins 5 ans recommandés)
- Bonne maîtrise de TypeScript
- Notions de Node.js / NestJS / ORM appréciées
- Bonnes connaissances en programmation orientée objet
- Capacité à écrire un simple test unitaire avec Jest
- Aucune expérience préalable des agents IA requise. « Niveau expert » désigne le niveau de pratique visé, pas un prérequis.

Public visé

Particuliers et professionnels :

- CTO / Technical Leader
- Développeur Backend
- Développeur Full Stack
- Architecte technique

Méthodes pédagogiques

- Live coding et orchestration d'agents en direct par le formateur

Mob programming : les participants codent et orchestrent ensemble

- Apports théoriques émergeant de la pratique, sur un cas métier réaliste
- Comparaisons entre code écrit à la main et code produit par les agents
- Challenges proposés sur le TDD, le DDD, la Clean Architecture et le refactoring

Les agents Claude Code du formateur sont utilisés tout au long des deux jours, et non cantonnés à une démonstration.

Selon le curseur IA choisi par le groupe, les participants observent, puis orchestrent eux-mêmes les agents : spécification, tests, code, validation par les tests et refactoring.

Capacités développées

À l'issue de la formation, les participants seront capables de :

Axe 1 — Software Craft

- Identifier un périmètre produit pertinent et découper les besoins métier en itérations courtes selon l'approche NoEstimates.
- Réaliser un Event Storming et identifier les Bounded Contexts ainsi que leurs interactions.
- Rédiger des spécifications BDD précises, exploitables et orientées comportements métier.
- Pratiquer le TDD Outside-In pour développer progressivement les fonctionnalités d'une application Node.js / TypeScript.
- Concevoir et exécuter des tests unitaires, d'intégration et end-to-end avec Jest et Testcontainers.
- Mettre en œuvre le mutation testing avec Stryker afin d'évaluer la pertinence et la robustesse des tests.
- Refactorer le code de production et les tests tout en préservant les comportements attendus.
- Concevoir des agrégats métier et appliquer les principes du DDD tactique.
- Structurer une application selon les principes de la Clean / Hexagonal Architecture en séparant le domaine métier des dépendances techniques.
- Organiser un monolithe modulaire et mettre en œuvre le CQRS dans un environnement NestJS.
- Évaluer la pertinence de patterns avancés comme l'Event Sourcing, les Sagas, l'Outbox Pattern et la communication asynchrone, selon les options abordées pendant la formation.

Axe 2 — Développement agentique

- Configurer et orchestrer plusieurs agents IA spécialisés pour automatiser les différentes phases du développement logiciel.
- Transformer un besoin métier brut en spécifications BDD structurées et directement exploitables par des agents IA.
- Superviser des agents réalisant le développement en TDD, depuis les tests jusqu'à l'implémentation et au refactoring.
- Contrôler la qualité du code généré par l'IA à l'aide des tests automatisés, du mutation testing et des règles architecturales.
- Détecter et corriger les hallucinations, les sur-implémentations et les décisions de conception inadaptées proposées par l'IA.
- Paramétrer les agents, les commandes spécialisées et les hooks afin de fiabiliser les workflows de développement.
- Utiliser les agents IA pour générer et configurer des composants techniques : environnements, données de test, adaptateurs et API.
- Superviser un workflow agentique complet tout en conservant la maîtrise des décisions fonctionnelles et architecturales.

Modalités d'évaluation

- QCM en fin de formation
- Mise en pratique : les participants pilotent eux-mêmes les agents sur le projet fil rouge
- Échanges et discussions

Documents et supports pour les apprenants

- Supports de cours en markdown très complets
- Claude Code : Agents et Skills personnels du formateur optimisés pour du TDD efficace en autonomie
- Projet Git du live coding avec X branches
- Partage de ressources (articles/blogs/discussions de forums Tech)
- Discussion privées sur le canal Slack dédié au mini-groupe post-formation

Déroulé de la formation

Jour 1 — Concevoir et livrer un premier Bounded Context

Matin — Du besoin brut au produit spécifié

- Accueil des participants, tour de table, présentation des parcours, recueil des attentes.
- Quiz de positionnement et présentation du projet fil rouge.
- Fondamentaux de l'agilité : approche produit, itérations courtes et NoEstimates.
- Analyse d'une spécification métier brute, remise en question des pratiques courantes et identification des besoins réels.
- Définition du premier produit avec des agents agilistes : exploration d'au moins cinq orientations business, priorisation et choix humain du périmètre.
- Event Storming assisté par IA (modes facilité et guidé) : workflow, macro-architecture, diagramme et identification des Bounded Contexts.
- DDD stratégique : langage ubiquitaire, frontières métier, duplication de données, messaging, transactions atomiques et eventual consistency.
- Introduction au CQRS et à l'Event Sourcing à partir des artefacts de l'Event Storming ; choix collectif sur l'Event Sourcing (option).
- Atelier BDD avec agent IA : clarification par QCM, comportements métier, scénarios par Bounded Context et spécifications exécutables catégorisées, séquencées, numérotées et suivies par statut.

Après-midi — Du TDD au Bounded Context opérationnel

- Articulation entre macro-architecture et TDD ; développement orienté comportements plutôt que base de données.
- ATDD / TDD Outside-In avec agents IA : cycle red – clean green but minimal – maybe refactor further, tests sociables et programming by wishful thinking ; prévention des hallucinations et de la sur-implémentation.
- Mutation testing avec Stryker : objectif de 100 % et limitation du code superflu.
- Clean / Hexagonal Architecture et DDD tactique : agrégats, règles métier, conception guidée par les tests et refactoring continu.
- Implémentation du CQRS et, en option, de l'Event Sourcing.
- Stratégie de tests : tests unitaires avec Jest, tests d'intégration et end-to-end avec Testcontainers / Docker.
- Adaptateurs secondaires : PostgreSQL, TypeORM / Prisma / Knex, Event Store éventuel, Snapshot Pattern, séparation métier / persistance et API externes.
- Adaptateur primaire REST avec NestJS ; orchestration multi-agents, supervision des décisions et pilotage à distance via Claude Code.
- Mise en service du premier Bounded Context : environnement Docker, configuration, requêtes API par story, seeds et jeux de données.
- Échanges, questions-réponses, pratique en partage d'écran et remise des supports du jour.

Jour 2 — Passer à l'échelle avec plusieurs Bounded Contexts

Matin — DDD avancé et maîtrise des agents

- Retour sur les acquis du jour 1, questions et clarification des notions.
- DDD tactique avancé : modélisation des agrégats, exercices avec et sans IA, concurrence et validation critique des propositions des agents.
- Conception du deuxième Bounded Context : frontières métier, duplication de données et comparaison des communications HTTP / messaging.
- Context Mapping : présentation et mise en œuvre de patterns adaptés (option).
- Principes de conception : POO, programmation fonctionnelle, Tell Don't Ask et SOLID, appliqués avec discernement par les agents.
- Personnalisation des agents IA : commandes « décorateurs », hooks et règles de cadrage.

Après-midi — Plusieurs Bounded Contexts, un monolithe modulaire

- Atelier pratique et évaluation : réalisation du deuxième Bounded Context par les participants avec leurs agents IA, selon les principes du Software Craft.
- Communication inter-contextes : Kafka / RabbitMQ et Outbox Pattern (option).
- Cohérence distribuée : eventual consistency, Sagas et Process Managers, avec mise en œuvre TDD assistée par IA (option).
- Construction du monolithe modulaire NestJS : monorepo pnpm workspaces, séparation des packages et schémas PostgreSQL par Bounded Context.
- Configuration des environnements dev / prod : substitution des adaptateurs, modules NestJS, agents de configuration et scripts automatisés.
- Échanges sur les enjeux d'entreprise : microservices ou monolithe modulaire, agilité et adoption des pratiques Software Craft face au vibe coding.
- Évaluation finale : QCM, auto-évaluation, échanges et validation des acquis.
- Installation des agents et skills Claude Code, remise des ressources et accès au Slack privé de suivi post-formation.
- Option transversale : Reprise et refactoring d'un code legacy généré par IA, avec les agents, vers une architecture testable et maintenable.

Modalité d'accès

Contactez-nous par mail : contact@wealcomecompany.com

Délais d'accès

Nous pouvons programmer les formations en fonction de vos contraintes et de nos disponibilités.

Moyens d'accompagnement et d'assistance (Formation à distance)

En cas de difficulté d'accès à la plateforme de visioconférence ou pour toute question technique ou pédagogique pendant la session, le stagiaire dispose d'un point de contact unique pour obtenir une assistance immédiate :

Contact d'assistance technique et pédagogique : WealCome

E-mail : contact@wealcomecompany.com

Accessibilité handicap

Nos formations sont accessibles aux personnes en situation de handicap.

Les aspects, l'accessibilité et le type de handicap au regard des modalités pédagogiques sont à évoquer impérativement au cours de l'entretien préalable à toute contractualisation afin de pouvoir orienter ou accompagner au mieux les personnes en situation de handicap.