

Updated on 10/02/2026

Sign Up

AI Development Training: Superpowers, BMAD, and OpenSpec

3 days (21 hours)

Overview

AI-assisted development requires a clear framework to transform an intention into reliable software. Using BMAD for scoping and design, OpenSpec for formalizing changes, and Superpowers for organizing their implementation and verification, this training offers a structured approach based on explicit deliverables and human validation.

Our AI Development Training: Superpowers, BMAD, and OpenSpec will enable you to move beyond ad-hoc code generation to build a coherent, traceable development process tailored to your team's needs.

You'll learn how to clarify requirements, document architectural choices, write verifiable specifications, and prepare an implementation plan broken down into manageable tasks.

By the end of the training, you'll be able to manage a development assistant, practice TDD, review proposed changes, and gather the necessary evidence before validating a feature.

This training also covers context management, code reviews, and confidentiality. The workshops allow you to test interactions between tools without assuming native integration or activating their commands simultaneously without supervision.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Define a requirement and document an architecture using BMAD
- Formalize a change and its acceptance criteria with OpenSpec
- Organize implementation and verification using Superpowers' capabilities
- Apply TDD and conduct an AI-assisted code review
- Test context transfers between tools and resolve conflicting instructions
- Verify code compliance with specifications before delivery

Target Audience

- Software developer
- Full-stack developer
- Tech Lead
- Software Architect
- QA Engineer

Requirements

- Experience in application development and the ability to independently review existing code
- Proficiency in common Git operations: commits, branches, and merges
- Knowledge of automated testing and code review principles
- Basic proficiency with a programming assistant that uses a language model
- Comfort working with the command line and writing documents in Markdown

Technical Requirements

- Computer running macOS, Linux, or Windows with WSL 2; 16 GB of RAM recommended
- Environment including Git, Node.js 20.19.0 or higher, npm, and uv for the necessary installations and scripts
- Operational access to Claude Code with an authorized account and sufficient quota for the workshops
- Installations of BMAD, OpenSpec, and Superpowers according to the versions and procedures approved prior to the session
- Stable internet connection and network permissions required for downloads and the AI, with a training repository that does not contain confidential data

Agenda for our AI Development Training: Superpowers, BMAD, and OpenSpec

[Day 1 - Morning]

Project Scope and Work Environment

- Principles of AI-assisted development: human responsibilities, model limitations, and verifiable deliverables

- Educational Approach to BMAD, OpenSpec, and Superpowers: Functional Overlaps and the Absence of Presumed Automatic Integration
- Installation and management of versions, access rights, and available capabilities in Claude Code
- Session isolation and control of active instructions: permissions, context, and data protection
- Scoping with BMAD: user problem, scope, constraints, and expected results
- Hands-on workshop: preparing the educational repository, verifying the environment, and producing a feature brief with BMAD.

[Day 1 - Afternoon]

Design and decision-making with BMAD

- Selecting a BMAD approach proportionate to the scale of the change and the project context
- Analysis of existing code: observed behaviors, dependencies, and technical constraints
- Formalizing requirements: use cases, business rules, non-functional requirements, and exclusions
- Architecture design: components, interfaces, trade-offs, and risks
- Human validation of deliverables and preparation of a handover package for OpenSpec
- Hands-on workshop: supplement the overarching project brief with requirements and an architectural proposal, then have the decisions validated.

[Day 2 - Morning]

Formalizing changes with OpenSpec

- Initializing OpenSpec and distinguishing between terminal commands and wizard commands
- Organizing reference specifications and change requests
- Adapting BMAD deliverables into proposals, specifications, designs, and tasks, without relying on automatic conversion
- Drafting requirements and scenarios: observable behaviors, edge cases, and Acceptance Criteria
- Description of specification changes: additions, modifications, and deletions, followed by a consistency check
- Hands-on workshop: converting the BMAD document into an OpenSpec change and verifying traceability between requirements, scenarios, and tasks.

[Day 2 - Afternoon]

Controlled planning with Superpowers

- How Superpowers skills work: triggering, context, and validation points
- Transferring OpenSpec artifacts and comparing the validated design with Superpowers' design questions
- Preventing instruction conflicts: selecting a single execution process and suspending Concurrent automation systems

- Creating the implementation plan: small tasks, relevant files, dependencies, and expected evidence
- Preparing for TDD: failure tests, minimal implementation, and refactoring
- Hands-on workshop: testing the OpenSpec transmission to Superpowers in a controlled session, producing a validated plan, and writing the first failing test.

[Day 3 - Morning]

Implementation, debugging, and reviews

- Step-by-step execution of the Superpowers plan with scope control and intermediate validations
- Choosing between sequential execution and delegation to subagents based on the capabilities of the environment
- Isolating changes using Git branches and worktrees
- Application of TDD and systematic debugging: reproduction, assumptions, and verification of fixes
- Double review: compliance with specifications and code quality, with controlled updates to OpenSpec tasks
- Hands-on workshop: implementing the core project feature, fixing a bug, and addressing code review comments.

[Day 3 - Afternoon] Verification

and traceable delivery

- Principle of verification before declaring success: executed commands, known results, and known limitations
- Verification of acceptance criteria, non-regression tests, and discrepancies between code and specifications
- Controlled feedback to OpenSpec: task reconciliation, specification synchronization, and change archiving
- Finalization of the development branch: human review, merge request, and automated checks
- Team adoption: traceability, confidentiality, token consumption, and tracking of transitions between tools
- Hands-on workshop: gathering compliance evidence, preparing for delivery, archiving changes OpenSpec and document the limits of the tested integration.

Target Audience

This training is intended for both individuals and companies—large or small—seeking to train their teams in new, advanced IT technologies or to acquire specific domain knowledge or modern methodologies.

Assessment at the Start of Training

The pre-training assessment complies with Qualiopi quality standards. Upon

final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency with various types of technologies, as well as their expectations and personal goals for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Hands-On Training: 60% Practical, 40% Theoretical. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the instructor—supported by examples and reflection sessions—and group work.

Assessment

At the end of the session, a multiple-choice quiz is administered to verify that participants have successfully acquired the skills.

Certification

A certificate will be issued to each trainee who has completed the entire training program.