

Updated on 05/15/2026

Sign up

Superpowers Training

3 days (21 hours)

Overview

Superpowers is a skills framework for AI development agents, designed to structure the way code assistants specify, plan, test, review, and validate their implementations.

Our Superpowers training will enable you to master a comprehensive methodology for managing the use of AI programming agents in your software projects.

You will learn how to install and use Superpowers with environments such as Claude Code, Codex, Gemini CLI, Cursor, OpenCode, or GitHub Copilot CLI.

You will be able to transform a vague request into a clear specification, produce a detailed implementation plan, apply TDD, conduct a structured code review, and verify results before delivery.

Through a software engineering-oriented approach, you will learn to avoid common pitfalls of AI agents: over-engineering, lack of testing, superficial fixes, misunderstanding of requirements, or validation without proof.

By the end of this training, you will be able to integrate Superpowers into your development workflows to produce more reliable, better-tested, and more easily maintainable code.

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

Objectives

- Understand the role of Superpowers in AI-assisted development workflows.
- Install and configure Superpowers with an AI programming agent.
- Structure the brainstorming, specification, and planning phases.
- Apply TDD, code reviews, and systematic verification.
- Scale the use of AI development agents within a team.

Target Audience

- Software developers
- AI engineers
- DevOps engineers
- Tech leads
- Software architects
- Teams using AI development agents

Prerequisites

- Solid foundation in software development
- Basic knowledge of Git
- Understanding of development and code review workflows

Technical prerequisites

- Laptop with at least 8 GB of RAM and administrator privileges.
- Stable internet connection to access AI agents, Git repositories, and development tools.
- Code editor installed, such as Visual Studio Code or Cursor.
- Git client installed and configured.
- Access to at least one compatible AI agent: Claude Code, Codex, Gemini CLI, Cursor, OpenCode, or GitHub Copilot CLI.

Superpowers Training Program

[Day 1 - Morning]

Understanding Superpowers and AI development agents

- Understand the role of Superpowers as a methodology for AI programming agents
- Distinguish Superpowers from an AI model, an IDE, or a simple collection of prompts
- Identify compatible platforms: Claude Code, Codex, Gemini CLI, Cursor, OpenCode, and GitHub Copilot CLI
- Understand the classic limitations of AI agents: over-engineering, lack of testing, and superficial fixes
- Positioning Superpowers within an AI-assisted software development workflow
- Hands-on workshop: Analyzing an unstructured AI agent workflow and identifying risks.

[Day 1 - Afternoon]

Installation, configuration, and getting started

- Installing Superpowers in a compatible environment
- Understanding the organization of skills and initial instructions
- Configure Superpowers for a development AI agent
- Verify that the agent correctly triggers the expected procedures
- Adapting the work environment to the tool being used
- Hands-on workshop: Installing Superpowers and validating skill triggers.

Brainstorming, clarification, and functional scoping

- Use the brainstorming workflow to clarify a request before coding
- Transform a vague idea into a requirement that an AI agent can use
- Identify necessary questions before design
- Minimizing misinterpretations and implicit assumptions
- Structure discussions to arrive at a stable request
- Hands-on workshop: Defining a feature based on an imprecise user request.

[Day 2 - Morning]

Design, Specification, and Implementation Plan

- Formalizing a request into clear and verifiable specifications
- Breaking down a feature into coherent deliverables
- Create a detailed implementation plan before development
- Apply the YAGNI and DRY principles in planning
- Prepare small, testable tasks that can be monitored by the AI agent
- Hands-on workshop: Writing a specification and an implementation plan for a mini-project.

[Day 2 - Afternoon]

TDD-guided development with AI agents

- Understanding the Test-Driven Development approach applied to AI agents
- Force the agent to write a test that fails before the code
- Limit the code produced to the bare minimum required to pass the test
- Refactor after validation while maintaining test coverage
- Avoiding premature implementations and unverified behavior
- Hands-on workshop: Implementing a feature using the red, green, refactor cycle.

Systematic debugging and root-cause correction

- Using a systematic debugging methodology
- Avoid superficial fixes based on symptoms

- Gather evidence before modifying the code
- Identify the root cause of a bug with the help of the AI agent
- Validate the fix through reproducible tests and observations
- Hands-on workshop: Diagnosing a bug, identifying the root cause, and verifying the fix.

[Day 3 - Morning]

Sub-agents, batch execution, and advanced workflows

- Understanding the use of sub-agents in AI-assisted development
- Breaking down work into independent and verifiable tasks
- Organizing batch execution without losing control
- Using the right skills for each situation: planning, implementation, review, feedback
- Minimizing the risk of discrepancies between the plan and the produced code
- Hands-on workshop: Completing a development project through guided tasks with interim checks.

[Day 3 - Afternoon]

Code review, verification, and branch closure

- Implement a structured code review with Superpowers
- Compare the implementation to the initial plan and acceptance criteria
- Identify blocking issues before marking a task as complete
- Run tests and collect proof of proper functioning
- Manage branch closure: merge, pull request, retention, or deletion
- Hands-on workshop: Comprehensive branch review, correction of discrepancies, and final validation.

Standardizing the use of Superpowers within the team

- Define standards for using Superpowers within a development team
- Structure AI agent workflows within a project context
- Establishing rules for quality, verification, and documentation
- Train teams in the disciplined use of AI programming agents
- Measuring the benefits, limitations, and risks associated with AI-assisted development
- Hands-on workshop: Creating a team workflow to specify, plan, code, test, and review with Superpowers.

Target Audience

This training is intended for both individuals and companies, large or small, wishing to train their teams in new advanced IT technologies or to acquire specific business knowledge or modern methods.

Assessment upon enrollment

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

final registration, the learner receives a self-assessment questionnaire that allows us to assess their estimated proficiency in various types of technologies, as well as their expectations and personal goals regarding 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 the smooth running of the training session.

Teaching Methods

Practical Course: 60% Practical, 40% Theory. Training materials distributed in digital format to all participants.

Organization

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

Assessment

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been properly acquired.

Certification

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