

Updated on 05/15/2026

Sign up

## GSD Training

3 days (21 hours)

### Overview

GSD, short for Get Shit Done, is a workflow system for AI development agents designed to structure the specification, planning, execution, verification, and delivery of code.

Our GSD training will enable you to master a comprehensive methodology for managing the work of AI programming agents and avoiding issues related to context drift.

You will learn to use key GSD commands such as `/gsd-new-project`, `/gsd-discuss-phase`, `/gsd-plan-phase`, `/gsd-execute-phase`, `/gsd-verify-work`, and `/gsd-ship`.

You will be able to transform a vague request into clear requirements, produce a roadmap, and structure context files such as `PROJECT.md`, `REQUIREMENTS.md`, `ROADMAP.md`, `STATE.md`, and `CONTEXT.md`.

Using a delivery-oriented approach, you will learn how to have AI agents work within specific contexts, limit hallucinations, verify results, and prepare branches or pull requests ready for review.

By the end of this training, you will be able to integrate GSD into your development workflows to specify, plan, execute, verify, and deliver more efficiently with AI agents.

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 GSD in AI-assisted development workflows.
- Structure a project using GSD commands and context files.
- Transform a request into requirements, a roadmap, and an implementation plan.
- Reduce context drift using subagents and controlled contexts.
- Verify, document, and deliver the work produced by an AI agent.

## Target Audience

- Software developers
- AI engineers
- Tech Leads
- DevOps engineers
- Software architects

## Prerequisites

- Solid foundation in software development
- Basic knowledge of Git
- Understanding of development workflows and pull requests
- Previous experience with an AI code assistant is a plus

## Technical requirements

- 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, Cursor, or Windsurf.
- Git client installed and configured.
- Access to at least one compatible AI agent: Claude Code, Codex, Cursor, Copilot, OpenCode, Gemini CLI, or Windsurf.

## Our GSD (Get Shit Done) Training Program

[Day 1 - Morning]

### Understanding GSD and AI agent workflows

- Understand GSD's role as a workflow system for development AI agents
- Distinguish GSD from an AI model, an IDE, or a simple system prompt
- Identify compatible platforms: Claude Code, Codex, Cursor, Copilot, OpenCode, Gemini CLI, and Windsurf
- Understand the limitations of unstructured AI agents: loss of context, omissions, hallucinations, and implementation drift
- Position GSD within an AI-assisted software development workflow
- Hands-on workshop: Analysis of an unstructured AI workflow and identification of context drift risks.

[Day 1 - Afternoon]

## Installation, configuration, and context files

- Installing GSD in an environment compatible with a code AI agent
- Understanding the main commands: /gsd-new-project, /gsd-discuss-phase, /gsd-plan-phase, /gsd-execute-phase, /gsd-verify-work, and /gsd-ship
- Structure project metadata files: PROJECT.md, REQUIREMENTS.md, ROADMAP.md, STATE.md, and CONTEXT.md
- Set up a clean working environment with Git, branches, and project documentation
- Understand the role of the persistent context between sessions
- Hands-on workshop: Initialize a GSD project and generate the first context files.

## Project clarification and discussion phase

- Use /gsd-new-project to define a new request
- Transform a vague idea into a project that can be executed by an AI agent
- Use /gsd-discuss-phase to address ambiguities before implementation
- Identify assumptions, constraints, dependencies, and acceptance criteria
- Minimize errors caused by a misunderstanding of the requirement
- Hands-on workshop: Clarify a user request and produce an initial structured project scope.

[Day 2 - Morning]

## Requirements, roadmap, and structured planning

- Document functional and technical requirements in REQUIREMENTS.md
- Build a coherent roadmap with milestones, priorities, and deliverables
- Use /gsd-plan-phase to produce a detailed implementation plan
- Break down the work into short, verifiable, and traceable tasks
- Prevent over-engineering and out-of-scope implementations
- Hands-on workshop: Draft the requirements, roadmap, and implementation plan for a mini-project.

[Day 2 - Afternoon]

## Controlled execution with AI agents

- Use /gsd-execute-phase to advance the implementation step by step
- Maintain alignment between the plan, product code, and tracking files
- Manage project status with STATE.md and decision traces
- Execute specific tasks without overwhelming the main context
- Review changes made by the AI agent before proceeding
- Hands-on workshop: Implement an initial feature following the GSD plan.

## Sub-agents, fresh contexts, and reducing context rot

- Understanding the phenomenon of context rot in long AI sessions
- Use subagents with their own contexts to isolate complex tasks
- Pass only the information relevant to each step
- Reducing hallucinations, omissions, and contradictions in technical decisions
- Maintaining a reliable project memory using structured files
- Hands-on workshop: Breaking down development into subtasks executed with controlled context.

## [Day 3 - Morning]

### Verification, quality, and validation of work

- Use /gsd-verify-work to verify the implementation before delivery
- Compare the result to the requirements, the roadmap, and the initial plan
- Identify discrepancies, omissions, regressions, and unverified behaviors
- Run tests, analyze results, and collect evidence
- Avoid declarative validations without actual verification
- Hands-on workshop: Verify a feature developed by an AI agent and correct any detected discrepancies.

## [Day 3 - Afternoon]

### Delivery, branch, PR, and ship cycle

- Use /gsd-ship to prepare the delivery of completed work
- Clean up the branch, review commits, and prepare a pull request
- Document changes, decisions, and remaining limitations
- Choose between merging, keeping, reworking, or deleting a branch
- Structure a clean end-of-cycle process to avoid hidden technical debt
- Hands-on workshop: Prepare a complete release with a summary, checks, and a pull request.

### Scale GSD within a development team

- Define GSD usage standards for product and engineering teams
- Establish conventions for files, branches, prompts, and validation criteria
- Integrate GSD into Git workflows, code reviews, CI/CD, and documentation
- Measure gains in productivity, quality, and traceability
- Identify the limitations, risks, and best practices for enterprise adoption
- Hands-on workshop: Build a comprehensive team workflow to specify, plan, execute, verify, and deliver with GSD.

## Target Audience

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

## 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 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

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.