Faysal Aziz

Steer the Machine: Harness Engineering for Beginners

Build fun, working apps with AI coding agents, and stay in control of every step.

FA-0838 · 3 days

Agentic & Generative AI · Software Development

Build and deploy small apps with AI coding agents while practising specifications, project instructions, tests, human review and recovery.

Introduction

An AI coding agent can help build a small game, but adding the next feature can expose a familiar problem: earlier decisions are undone, one fix breaks another, and the developer loses track of how the code works. Clear specifications, feedback and review help keep that work on course.

Harness engineering is the practice of building the structure around an agent: the instructions it reads, its tools and execution boundaries, the checks that test its output, and the points where a human approves or redirects it. Thinking of an agent as a model plus its harness is a useful starting point. Written instructions guide behaviour; permissions, isolation and checks provide controls beyond the prompt.

This course teaches harness engineering from the ground up to developers who are new to AI-assisted work. Participants learn how AI models and coding agents work, set up a real workbench, turn ideas into clear specifications and review the changes the agent makes. Small, fun applications provide practice with a single-page app, a database and a model API, followed by supervised deployment to a training server.

Learning outcomes

By the end of this course, participants will be able to:

- Explain how large language models generate output, and why that output can be wrong.
- Describe what makes an AI coding agent different from a chat assistant.
- Explain the harness concept and the role of instructions, rules, tools, checks and human approval.

- Set up a working environment with a terminal, Git, GitHub, a coding agent and a remote server.
- Write project instruction files and rules that guide a coding agent.
- Turn an idea into a specification with clear scope, acceptance criteria and tasks.
- Build small applications with a coding agent working from a specification.
- Review agent changes, use a second agent as a reviewer, and roll back safely.
- Deploy an application to a Linux server and keep secrets out of the code.
- Recognise the main risks of AI coding agents and apply simple controls against them.

Prerequisites

- Ability to write and run small programs independently in at least one programming language.
- A laptop with a current browser, a code editor, Git and an SSH client, with rights to install software.
- A GitHub account.
- An account for an AI coding agent such as OpenAI Codex or Claude Code.
- Internet access permitted for the selected training tools and services.
- Authorised SSH access to the designated remote training servers, with outbound SSH permitted by the device and network.

Training outline

1. AI Foundations for Developers

1. AI, machine learning and generative AI
2. How large language models generate answers
3. Tokens, context windows and model limits
4. Hallucinations and non-deterministic output
5. Prompts, context and instructions
6. What makes an AI system an agent

2. Meet the AI Coding Agent

1. Chat assistants versus coding agents
2. The agent loop: read, plan, act, check
3. Tools, file access and terminal commands
4. Permissions, approvals and sandboxes
5. Agent equals model plus harness

6. Vibe coding and its limits

3. Setting Up the Workbench

1. Terminal and command-line essentials
2. Git commits, branches and history
3. GitHub repositories and pull requests
4. Installing and signing in to a coding agent
5. SSH access to a remote server
6. A first agent-built app

4. Building a First Harness

1. Project instruction files: AGENTS.md and CLAUDE.md
2. Project rules, permission boundaries and checks that enforce critical constraints
3. Coding conventions and project structure
4. Reusable agent skills
5. Tests and automatic checks as feedback
6. Secrets and safe configuration

5. From Idea to Specification

1. Describing the problem and the user
2. Scope, exclusions and the minimum viable product
3. Acceptance criteria and completion checks
4. Planning and breaking work into tasks
5. Spec-driven development with GitHub Spec Kit
6. Keeping specifications and code in step

6. Human in the Loop

1. Reading and understanding agent changes
2. Approving, rejecting and redirecting the agent
3. A second agent as a reviewer, with human judgement and test evidence
4. Build reports and review checklists
5. Checkpoints, rollback and recovery

7. Building Fun Apps with AI

1. A single-page web app from a specification
2. A data-backed app with a database
3. AI-powered features through a model API
4. Adding features to an existing app
5. Debugging and fixing with the agent

8. Shipping to a Server

1. Running an app on a Linux server
2. Environment variables and API keys
3. Packaging an app with Docker
4. Supervised deployment by the agent
5. Logs, updates and rollback

9. Working Smart with AI Agents

1. Choosing models and controlling cost
2. Keeping context focused
3. Prompt injection and unsafe agent actions
4. A personal harness checklist
5. The path to advanced harness engineering

Disclaimer

This course outline serves as a guideline for training planning and delivery. The trainer reserves the right to amend, reorder, expand, reduce, or otherwise modify the content and delivery approach as he considers appropriate, taking into account participant experience, organisational requirements, technical constraints, and available time, without prior notice.

© 2026 Faysal Aziz · Kuala Lumpur