

# Claude Code Architect

CCA | 1 Day



Turn Claude Code into automated pipelines that build, verify, and correct on their own. This one-day course covers loop engineering: defining clear inputs and outputs, using tests as pass/fail gates, and constructing agentic build-verify-correct loops that run with minimal supervision. You'll engineer real loops hands-on and learn to make them reliable, bounded, and observable. It is the capstone of the three-course series.

## Who should take this course?

This course is for developers, leads, and platform engineers who want to automate Claude Code rather than drive it by hand. You should be comfortable orchestrating multiple agents, whether from the Orchestrator course or your own experience. Comfort with a terminal, a code editor, testing, and CI concepts is assumed. Come ready to design and run automated loops throughout the day.

## Course content

This course builds your fluency with loop engineering, from a single build-verify-correct cycle to production pipelines that run on a schedule. You'll work hands-on against real code and real tests. This is the third of three courses; it builds on Developer and Orchestrator. Each module closes with a hands-on lab.

### 1. LOOP ENGINEERING FOUNDATIONS

- From conversation to autonomous loop
- Anatomy of a loop: build, verify, correct
- Defining clear inputs and outputs
- Success criteria the loop can check
- Bounding loops: iteration caps and budgets
- Hands-on lab

### 2. TESTS AS PASS/FAIL GATES

- Why deterministic gates beat model judgment
- Designing tests as the loop's oracle
- Wiring build, lint, and test results in
- Exit codes, structured results, and parsing
- Failing fast and surfacing the cause
- Hands-on lab

### 3. BUILD, VERIFY, CORRECT

- The correction step: feeding failures back
- Self-repair and retry strategies
- Loop-until-green and loop-until-dry patterns
- Detecting non-convergence and bailing out
- Verification you can trust
- Hands-on lab

### 4. PRODUCTION PIPELINES

- Composing loops into pipelines
- Scheduling and triggering automated runs
- Observability: logs, metrics, audit trails
- Guardrails, cost controls, and kill switches
- Operating pipelines over time
- Hands-on lab

## Course Designer

This course was designed by Richard Hundhausen, a Professional Scrum Trainer, co-creator of the Nexus framework, and a longtime developer who now helps delivery teams adopt AI-assisted engineering with tools like Claude Code. To see other Claude Code and AI engineering courses, visit [www.accentient.com](http://www.accentient.com).