

AUTOMOTIVE SPICE® CONSULTING · ASSESSMENTS · TRAINING

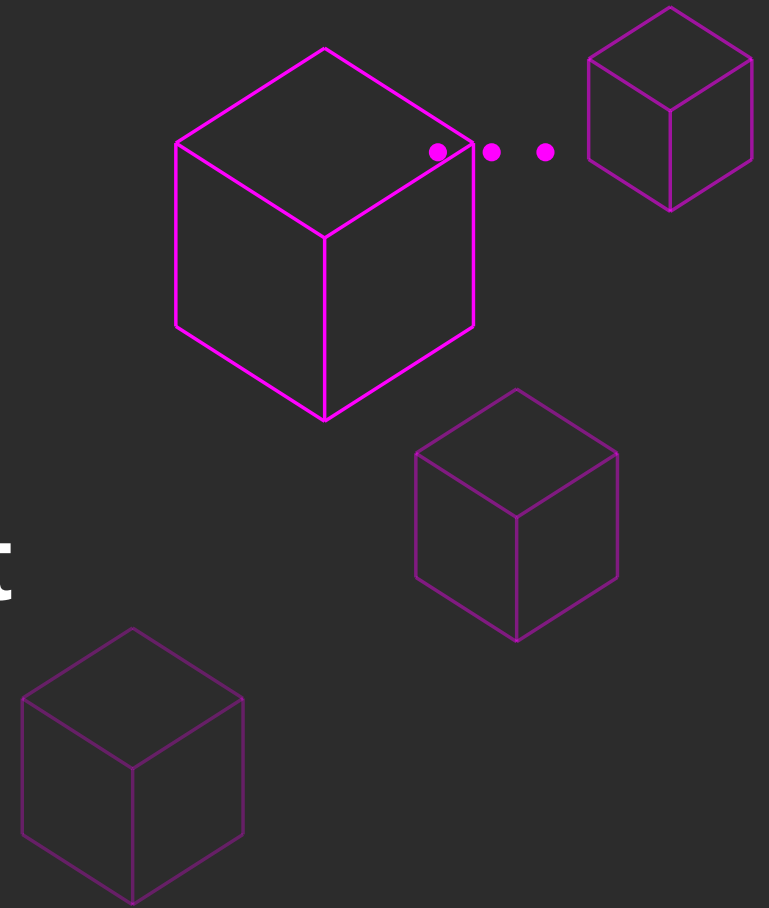
Professional Tool Development with Claude Code

5-Day Intensive Training for Automotive SPICE® Assessors

Format: 5 days, on-site or remote

Group: max. 8 participants

As of: August 2026



Training Agenda

Foundation

01

Set up your toolkit, sharpen your mental model, define your own project

Specification

02

Requirements and architecture an agent can actually implement

Development Loop

03

Build, verify, discard — and keep ownership

Professionalization

04

Automate, integrate, secure, and ship

Evidence & Demo Day

05

Document, present, and evaluate with an assessor's eye

CONTEXT

Why This Training

- You'll soon be assessing projects that AI helped build — you can only judge what you've mastered yourself.
- AI projects rarely fail on the model; they fail on specification, verification, and configuration.
- Home turf: you already know what good process looks like — here, you run it yourself.
- The outcome of the week isn't a slide deck — it's your own working tool.

Learning Objectives

After this training, participants will be able to:

01

Specify projects an agent can act on

Frame context, requirements, and acceptance criteria so an agent can implement them

02

Confidently steer the development loop

Master Explore–Plan–Code–Verify, review results, stop bad turns early

03

Professionalize the working environment

Use CLAUDE.md, slash commands, hooks, subagents, MCP, and version control productively

04

Evaluate AI-assisted development

Establish traceability and review discipline — in their own project and in assessments

Familiar Process, New Tool

- Requirements (SWE.1): CLAUDE.md and a spec file replace the offhand prompt to the model.
- Architecture (SWE.2): plan first, then build — decisions live as ADRs in the repository.
- Verification (SWE.4–6): tests are the gate, not a good first impression of the output.
- Configuration (SUP.8 / SUP.10): branching, commits, and review make AI work traceable.

Division of Labor: Tool and Responsibility

✓ What Claude Code Handles

- Scaffolding, refactors, and test harnesses at high speed
- Research across unfamiliar code and libraries
- Documentation kept consistent with the source
- Recurring workflows as commands, hooks, or skills

! What Stays With You

- The target vision, requirements, and acceptance criteria
- Architecture decisions and their rationale
- Review of every change — unreviewed means unbuilt
- Licensing, privacy, and security questions

Prerequisites for Participation

✓ Hardware and Setup

- Your own laptop running Windows, macOS, or Linux with a stable internet connection
- Local administrator rights — installing software is part of the training
- No device management that blocks the terminal or package managers
- Programming experience is welcome but not required

! Accounts and Budget

- A Claude Pro account, set up and tested before the training starts
- A credit card for small expenses during the week
- A GitHub account for version control and collaboration
- Your own project idea — small, real, achievable in five days

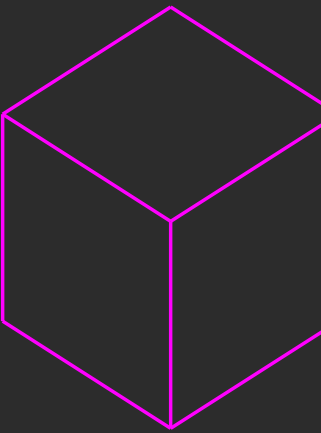
Costs Beyond the Training Fee

- Claude Pro costs \$20/month, cancel anytime — mandatory for all participants.
- Pro has session and weekly limits; anyone who hits them during the intensive week can temporarily upgrade to Max.
- API access isn't included in Pro and is billed separately: roughly €5–20 depending on the project.
- An optional small server costs from about €5/month. Budget €50–80 in incidental costs overall.

DAY 01

Foundation

Set up your toolkit, sharpen your mental model, and define your own project.



Day 1 Schedule

Time	Topic	Method
9:00–9:40 AM	Welcome and introductions: experience, expectations, project sketch	Round-robin
9:40–10:30 AM	Keynote excerpt on prompt structure and its evaluation	Video & discussion
10:45–11:30 AM	Exercise: tell the AI who you are — in the browser	Hands-on exercise
11:30 AM–12:30 PM	Hallucination and privacy: spotting errors and ground rules	Video & exercise
1:30–2:30 PM	Installation, terminal, and Git — everyone at the same starting point	Guided exercise
2:30–4:00 PM	How a coding agent works: from prompt to task for the tool	Talk & demo
4:00–5:30 PM	Sharpening the project idea, creating the repository, first commit	Pitch & individual work

Four Minutes, Four Points

- Who you are: your role, industry, and how long you've been assessing.
- What you bring: experience with AI tools — and, honestly, the gaps too.
- What you expect: what would tell you by Friday that the week was worth it.
- What you want to build: your project idea in two sentences, still rough.

An AI Keynote as Our Shared Starting Point

✓ Excerpts from the Keynote

- Prof. Dr. Christian Stummeyer, CSP — Regensburg, November 2025
- 12:24 — why models respond differently for different audiences
- 13:43–16:56 — anatomy of a prompt and three rules of thumb
- About five minutes total — youtu.be/CIRRT8m3kH8

! Your Observation Task

- Which process steps you'd want to see in an assessment show up here?
- Which are missing — and would you accept that in a real project?
- Where does enthusiasm end and engineering begin?
- Your answers set the thread for the remaining four days

Six Things to Specify Before You Ask for a Line of Code

✓ What Belongs in the Brief

- Context — who you are, your experience, and honestly your gaps
- Role — who Claude should be and what it needs to know
- Task — what should result, with a checkable acceptance criterion
- Audience — who the tool is built for, not just the text
- Tone — how terse or explanatory Claude should be with you
- Format — what comes back: plan first, then code, language, style

! What That Sounds Like in Practice

"I'm an Automotive SPICE assessor, fifteen years of process experience, barely any Python.

You're an experienced Python developer who knows the pitfalls of Excel files.

Task: a script that turns assessment results from Excel into a short report.
Done when it processes my file without errors.

Audience: assessors who should be able to use it without onboarding.

Answer concisely, but explain every library in one sentence.

Show me the plan first, then the code — in English, with type annotations."

Rules of Thumb, Folklore, and What You Can Verify Yourself

✓ What Holds Up to Scrutiny

- Concrete context beats claimed experience: “knows the pitfalls of Excel” beats “20 years of consulting”
- Plan first, then build — as a step in the tool, not a magic phrase in the prompt
- Blunt, rude prompts measurably worsen results — neutral is enough
- Checkable acceptance criteria: the only thing you can verify afterward

! What's Folklore

- “Politeness helps” — studies disagree, effects are model- and language-dependent
- “You are an expert with 20 years of experience” — provably doesn't improve factual accuracy
- “Take your time” — mostly ineffective on models that already plan internally
- If someone claims an effect, they should show it: run two variants, don't just retell it

Tell the AI Who You Are

Following the discussion — 45 minutes at your own computer:

Describe yourself in five sentences

01

Role, professional experience, specialty — and honestly, what you can't do either

State what kind of answer you need

02

Depth, length, jargon vs. plain explanation — and how you'd recognize a wrong answer

Ask the same question twice

03

Once without context, once with. Put both answers side by side and compare

Save what worked

04

These sentences move into your CLAUDE.md tomorrow and then apply automatically in every session

Find the Errors in Your Own Field

Excerpt 16:56–19:25, then 30 minutes at the computer:

Ask about your own field

01

Have a PAM process explained to you, with base practices, numbers, and mapping

Demand sources

02

Chapter, number, page. This is exactly where models fabricate most reliably — and most convincingly

Mark every error

03

Go through the text line by line. Here, you are the reference, not the model

Note what you almost missed

04

The better your prompt, the smoother the text — and the harder an error is to spot

What Doesn't Go In — and What That Means for Your Project

✓ Stays Out

- Personal data: names, ratings, anything from real assessments
- Your clients' trade secrets: architectures, defect lists, contracts
- Credentials, keys, and tokens — not even briefly, just to try something
- Anything covered by a non-disclosure agreement

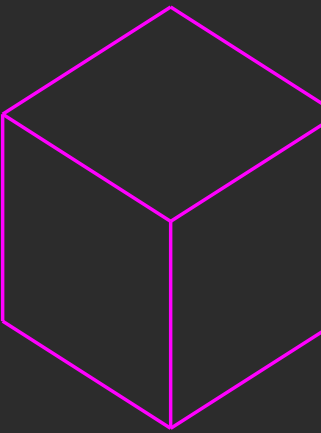
! Consequences for Your Project

- You work with fabricated test data — Claude generates it for you in minutes
- Structure over content: your file's format isn't sensitive, its contents are
- Check beforehand whether your employer has its own rules on AI tools
- Your tool still turns out real — you feed it real data yourself, later

DAY 02

Specification

From wish to a requirement an agent can reliably implement.



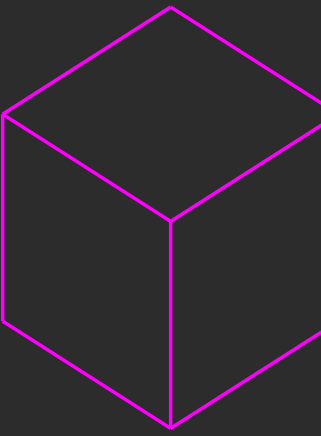
Day 2 Schedule

Time	Topic	Method
9:00–10:30 AM	CLAUDE.md: the project handbook for humans and agents	Talk & exercise
10:45 AM–12:30 PM	Requirements with acceptance criteria instead of wish lists	Individual work
1:30–3:00 PM	Plan Mode: designing architecture before code exists	Guided exercise
3:15–4:45 PM	Justifying decisions and filing them as ADRs in the repository	Individual work
4:45–5:30 PM	Peer-reviewing each other's specs with an assessor's eye	Peer review

DAY 03

Development Loop

Build, verify, discard — and keep ownership throughout.



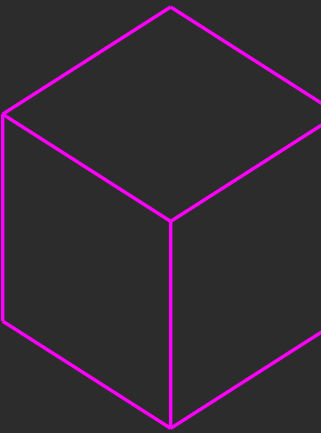
Day 3 Schedule

Time	Topic	Method
9:00–10:30 AM	Explore – Plan – Code – Verify: the controlled loop	Talk & demo
10:45 AM–12:30 PM	Test first: verification as the agent's gate	Individual work
1:30–3:00 PM	Reading diffs, running review, stopping bad turns	Pair exercise
3:15–4:45 PM	Debugging, subagents, and parallel work streams	Guided exercise
4:45–5:30 PM	First working walkthrough of your own project	Individual work

DAY 04

Professionalization

Make it repeatable, integrate it, secure it, and ship it.



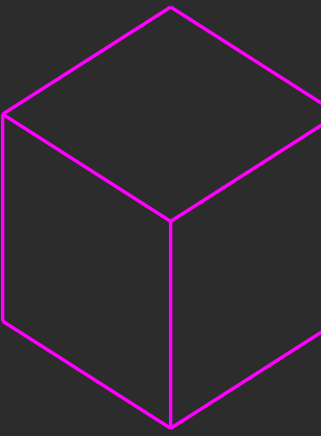
Day 4 Schedule

Time	Topic	Method
9:00–10:30 AM	Slash commands, hooks, and skills: building your own workflows	Talk & exercise
10:45 AM–12:30 PM	Integrating MCP and external interfaces, managing keys	Guided exercise
1:30–3:00 PM	Secrets, permissions, privacy, and licensing	Talk & case study
3:15–4:45 PM	Deployment: from local state to your own server	Individual work
4:45–5:30 PM	Setting up automated checks on every change	Individual work

DAY 05

Evidence & Demo Day

Finish, document, present — and put on the assessor's lens.



Day 5 Schedule

Time	Topic	Method
9:00–10:30 AM	Finishing touches: closing open items, documentation	Individual work
10:45 AM–12:30 PM	Building evidence: traceability and review records in the repository	Exercise & discussion
1:30–2:45 PM	Demo Day: everyone presents their own result	Presentation
3:00–4:15 PM	The assessor's lens: evaluating AI-assisted development	Group work
4:15–5:00 PM	Transfer to everyday work, feedback, and wrap-up	Discussion

Your Project: Choosing and Scoping It

✓ Good Fit

- A tool that regularly saves you work yourself
- Its value fits in one sentence, success is objectively checkable
- Reachable at a first usable state within five days
- For example: analyzing assessment data, checklists, reports

! Better to Avoid

- Projects involving real client or personal data
- Dependencies on approvals or systems outside the week
- Too ambitious — platforms, portals, anything “comprehensive”
- Topics with no personal connection; interest tends to fade by day five without one

OUTCOMES

What's in Place at the End of Each Day

Day	Outcome by End of Day	Artifact	Method	Ties to
1	Working environment is running, project idea is sharpened and locked in	Repository + commit	Pitch	Start
2	Requirements and architecture are documented and reviewed	CLAUDE.md, spec, ADR	Peer review	SWE.1-2
3	First working walkthrough with automated tests	Source code + tests	Code review	SWE.3-4
4	Project is automated, integrated, and published	Commands, hooks, CI	Exercise	SUP.8-10
5	Finished tool, presented and demonstrably developed	Docs + demo	Demo Day	Evidence

Five Days. One Working Tool.

QUESTIONS & CONTACT

Registration and inquiries

kosmas.kopmeier@spicey3.de

+49 179 5306208

www.spicey3.de

