re:cinq

re:cinq

CodeGenAI Developer Training

Table of contents

Goal

AI-enhanced software development coaching program for an entire software engineering department. This program aims to accelerate software delivery, reduce manual effort, and improve code quality without increasing headcount.

The program should equip engineers with practical skills in four AI coding strategies: prototyping with AI, AI-assisted development (e.g., Cursor), agentic development (e.g., Claude Code), and spec-driven development (e.g., Kiro). The core of the program will be cohort-based learning tracks delivered to groups of 20-25 engineers, which will include live sessions on-site, real-world examples, and effective prompting techniques.

Key to the program's success is long-term tool adoption, as is reflected in the need for success metrics to be defined, and a core 'train-the-trainer' offering to ensure that the training efforts are not a one-off event.

re:cinq

Company overview

re:cinq is a European AI-native consultancy established by veterans of the prominent cloud-native transformation company Container Solutions. Its CEO, Pini Reznik, authored the O'Reilly book *Cloud Native Transformation*, and he and CTO Michael Müller will soon publish *From Cloud Native to AI Native: Catching the Next Wave of Innovation*.

A successful technology transformation is a multi-year journey that hinges on deep investment in training and fundamental behavioral change, a methodology we have proven with some of the wWorld's largest global enterprises. Our approach moves beyond simply teaching a new tool; it's about rewiring how an organization thinks and operates. We begin by empowering a small, dedicated core team to research the new paradigm and build a high-quality Minimum Viable Product (MVP). This team becomes the seed of the new culture and technical expertise. **MVP is** the set of **working practices, behaviours and institutional knowledge**.

A prime example is our work at a past customer, where the initial scaling was catalyzed by an intensive, hands-on "water tower offsite." In just three days, this immersive training program equipped the first wave of teams to deliver their applications to production on the new Cloud Native platform. This successful model, which combines deep technical education with new approaches to team structure, architecture, and product design, was then rolled out in "water tower workshops" across the organization. This strategy of gradual, expert-led training ensures that change is not just adopted, but deeply embedded, creating a sustainable culture of innovation and a genuine competitive advantage.

re:cinq's trainers

CTO Michael Müller designed and oversaw the **multi-year engineer training programme for *a large fashion company*** that enabled their organisation-wide cloud native transformation. As a leading member of the early cloud native community, Michael was part of the group that **defined the *Cloud Native Computing Foundation*'s certifications and syllabi** for the Kubernetes administration and application development. Michael was a **lecturer for Lucerne University**, teaching DevOps principles and practices for six years.

Daniel Jones is UK Partner at re:cinq, and has authored many training courses in his career, **creating education programmes for *The Linux Foundation***, *vmWare* and more across a variety of technologies. *EngineerBetter*, Daniel's prior consultancy, pioneered successful mob-programming immersive training experiences, maximising hands-on time and collaborative learning. Daniel has delivered courses to hundreds of trainees, including a **week-long 'train-the-trainers' session to *General Electric*** on behalf of *The Linux Foundation*. Outside of the IT industry, Daniel is a jiu-jitsu black belt who has **taught martial arts and self defence** to both adults and teenagers, with some sessions featuring over 200 participants.

Michael has overseen the adoption and rollout of CodeGenAI tools across re:cinq's engineering teams, and both Daniel and Michael use Cursor and Claude Code regularly. As part of his responsibilities hosting re:cinq's *Waves Of Innovation* podcast, Daniel speaks to technology leaders rolling out CodeGenAI

practices and developing CodeGenAI tools. More information can be found below:

- re:cinq blog
- Waves Of Innovation podcast

re:cinq's training philosophy

Our approach leverages our experience of both training and organisational transformation, and is informed by cognitive psychology and neuroscience. We incorporate learnings and techniques from these fields to improve the ability and motivation of trainees to put their newly-gained knowledge into practice.

Slideware should be kept to the minimum required to enable more meaningful experiential learning. Slides should establish context and provide a scaffold on which trainees can fix knowledge learned in exercises. Trainers should enrich the course with anecdotes and different perspectives. The bulk of the learning should be the hands-on experiences of trainees. Trainees should discuss and share experiences, verbally retracing the steps taken in exercises.

Spaced repetition, social importance and episodic memory should be leveraged to improve learning outcomes. Recall should be actively exercised to ensure that trainees can retrieve knowledge on demand.

Teaching students is not enough: all opportunities should be taken to increase the likelihood of behaviour change. This requires an empathetic and considered approach to material delivery.

Program Structure

Based on our extensive experience of authoring and delivering training, and effecting broad-scale technology transformations at large organisations, we believe that the following structure presents the most effective means to drive lasting adoption of CodeGenAI tools and practices.

Overview

All content is merged into one track comprising:

- **5 units** of education, plus final recap
- **3.5 days** of instructor-led training per trainee
- additional **2 days** for **trainee trainers**.

Phase 1 - discovery

The discovery phase remains as requested. **Value-stream maps** are created to highlight bottlenecks in the delivery process.

Phase 2 - pilot

The curriculum (detailed below) is delivered to a pilot cohort. The foundation and first modules are compressed into one day, enabled by the lower headcount.

Phase 3 - roll-out

Roll-out the programme to remaining cohorts.

To maximise engagement, **all remaining cohorts** attend an exciting **one-day** on-site event of instructor-led **foundational training**, featuring a large-scale interactive workshop using Liberating Structures to surface any concerns, anxieties and reservations.

The **train-the-trainers event** is scheduled to take place immediately after the foundation day, and before the remaining cohorts start their first learning

iteration. This gives trainer-trainees one-to-one time with course authors ahead of the rest of their cohorts, and maximises contiguous in-person time. Attendees give supervised mock delivery of units to their peers.

Curriculum

A (bi-)**weekly cadence** starts in which each cohort:

1. Reviews last iteration's progress (see point 5)
2. Receives instructor-led training on one type of AI coding (see syllabus)
3. Performs a curated exercise under supervision
4. Discusses the exercise (including "what did we just do," to improve recall)
5. Is set a challenge to use the tools/techniques covered in their daily work during the upcoming iteration
6. Disbands until the next session

Cohorts each have a **dedicated Slack channel** for ad-hoc requests for help, coaching, and discussion.

Rationale

The progression of autonomy, power, and guard rails in CodeGenAI tools can be seen below:

1. Using an LLM chat interface to ask coding questions
2. Integrating that chat window into the IDE, so it can automatically provide context (early versions of Copilot and Cursor)
3. Agentic capabilities in an IDE (modern Cursor), to achieve outcomes rather than give granular instructions
4. Agentic capabilities in external tools (Claude Code, Cursor CLI) to de-emphasise the coding experience, and focus on approval and guidance
5. Spec-driven AI code generation (Kiro, Humbug), because keeping agentic code generation 'on-track' is difficult
6. Unsupervised agents responding to GitHub Issues, PRs, etcetera (out of scope).

By learning about and experiencing increasingly sophisticated usage sequentially, trainees will gain a better understanding of the motivations of each innovation - it's easier to grasp why Kiro's spec-driven approach is valuable if you've experienced Cursor going off-topic in agentic mode.

All-cohorts one-day kick-off

We envision, if possible, a one-day all-cohort kick-off session that combines instructor-led training, and crucially a workshop with all participants.

An all-participants workshop will allow trainees to voice concerns about the upcoming programme. Some may have concerns about CodeGenAI removing the work they enjoy most, or even leading to redundancies. By exposing these concerns we can address them. By being invited to discuss the upcoming programme, attendees will feel listened to and consulted - reducing reactance, the phenomenon by which people resist change that they believe is forced upon them.

The one-day event will have more of the feel of an internal tech conference, increasing engagement and excitement of those involved, again increasing the likelihood of lasting behaviour change.

Gathering all cohorts together for the initial kick-off session may (depending on logistics) make for a more cost-effective use of time.

Hands-on labs

We believe that hands-on labs are an essential part of successful learning, and therefore an integral part of the instructor-led training sessions. These are performed under supervision to ensure all trainees have a minimum level of competence with tools. By undertaking and discussing these exercises in the group sessions, trainees can benefit from the insights of others and also form social, episodic memories which are more persistent.

Learning cadence

By introducing content at regular intervals at a time, we avoid overloading trainees with too much information. By reviewing prior content before moving on to the next, we leverage spaced repetition, a technique proven to improve learning outcomes.

The cadence could be weekly or every two weeks, to align with trainees' other responsibilities and internal delivery schedule.

Much like practices such as TDD, using CodeGenAI is more about experience than remembering facts - by broaching topics incrementally, we allow space for regular challenges that give that experience.

Regular challenges

To ensure effective learning, it is crucial to apply tools and techniques in real-world scenarios rather than only through contrived exercises. By setting challenges as part of each session we give trainees more opportunities to learn, more realistic experiences to learn from, and crucially also discover any real-world barriers to adoption.

Provisional syllabus

The syllabus is expected to be refined as part of the discovery process.

- Unit 1: Foundation
    - LLMs, non-determinism and context
    - Data, privacy, intellectual property
    - Prompt engineering
    - The CodeGenAI ladder
    - Common CodeGenAI pitfalls
- Unit 2: Using an LLM as a better search engine
    - LLMs to understand legacy code
    - LLMs to generate documentation
- Unit 3: Assisting CodeGenAI

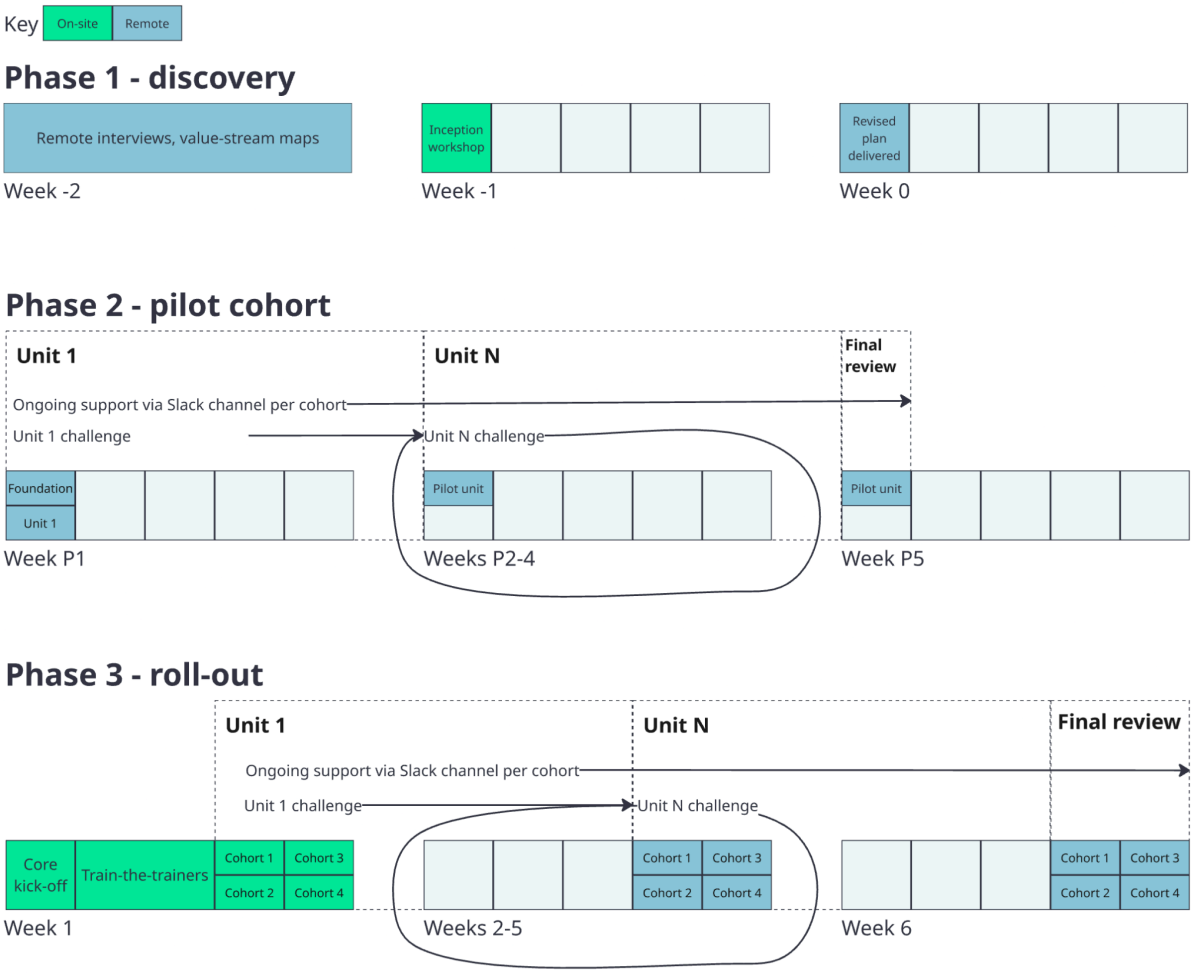
    - Tools: Copilot, Cursor (non-agentic)
    - Context engineering
- Unit 4: Agentic CodeGenAI
    - Tools: Cursor (agentic), Claude Code
    - Techniques for different tasks
    - Rules and guardrails
- Unit 5: Spec-driven CodeGenAI
    - Tools: Kiro (pending general release)
    - Source code is the new compiled code
    - Version control and reproducibility
- Unit 6: Final review

Feedback opportunities

Fast feedback and action thereupon is key to assuring useful outcomes. re:cinq defines several feedback points so that amendments can be made as necessary:

- Discovery phase
    - Approach formed by staff interviews, and value-stream mapping exercise to determine where development bottlenecks are.
- Instructor-led training
    - End-of-day retrospective with trainees to capture the experience of each cohort, allowing improvements for subsequent cohorts.
    - Additionally increases trainee satisfaction and improves rapport ahead of multi-week coaching.
- Trainee Slack
    - Anonymous polls after each instructor-led training session, to ensure feedback from less-confident trainees is captured.
    - Ongoing channel membership for direct communications during multi-week coaching.
- Internal 'break-glass'
    - Trainees invited to escalate any concerns that they do not feel comfortable raising with re:cinq to an appointed staff member.

Delivery schedule



- Phase 1 - discovery
  - Interviews, value-stream-mapping (1 week, remote)
  - Inception meeting (1 day, in-person)
    - Goals, anti-goals, risks, assumptions, segmentation, metrics, syllabus refinement
  - Delivery of refined training structure (1 week after inception, remote)

- Phase 2 - single-cohort pilot
  - Foundation module and first module (1 day, remotely)
  - Ongoing education and coaching (regular 1 or 2 week cadence per unit)
    - Weekly remote session (4 hours per unit per cohort)

            - Remote video call with cohort, reviewing their progress of the last unit, interactive debugging, show-and-tell
            - Instructor-led education on the next unit
            - One supervised, curated exercise
            - Discussion and review of exercise
            - Trainees challenged to use the approach in their week's work
        - Ongoing dedicated Slack channel for each cohort

- Phase 3 - delivery roll-out
    - Foundation (1 day, in-person)
        - Instructor-led foundational training (see unit 1 of syllabus)
        - Interactive workshop using Liberating Structures
    - Train-the-trainer (2 days, in-person)
        - All training units are delivered to trainers
        - Trainers complete exercises for all units
        - Mock delivery of material to other trainers

Success criteria

re:cinq recognises the importance of defining and driving towards measurable outcomes.

The following initial metrics are committed to, with further metrics discussed as part of discovery:

- Completion: ≥ 90% trainees complete curriculum
- Developer confidence: ≥ 80% respondents report increase

Metrics for further discovery and discussion

Time savings on repetitive tasks will depend on the types of task. Some engineering leaders anecdotally report an *increase* in repetitive work (which logically follows, because whatever isn't automated becomes repetitive).

Tool adoption is a metric that will need some discussion, as this is highly dependent on internal processes, such as procurement and whitelisting of new tools. Our approach of having trainees use tools during their working weeks will expose impediments to adoption.

Quality improvement may also need further discussion, as this is dependent on the delivery process. As the reader may be aware, the DORA '24 report showed that while perceived productivity increased at the individual level, overall delivery velocity and stability decreased. Recent work from Stanford showed that productivity gains diminish with the complexity, size and maturity of the codebase - in some cases for niche languages, CodeGenAI adoption was shown to decrease productivity. Predicting changes in overall system performance based only on the optimisation of individual parts is intrinsically challenging, and requires care and consideration.