CODEGYM

KHOÁ HỌC

AI-Augmented Software Development

Phát triển phần mềm chuyên nghiệp với AI

- 6 buổi × 3 giờ = **18 giờ học trực tiếp**
- Thực hành trên sandbox linehaul

MỤC LỤC

1. Giới thiệu	3
2. Thông tin chung	4
3. Đối tượng & Điều kiện tiên quyết	5
4. Kết quả đầu ra	6
5. Codebase thực hành xuyên suốt	7
6. Lịch học chi tiết (6 Buổi)	8-10
7. Đánh giá	11
8. Đưa kỹ năng vào thực tế sau khóa học	12
9. Tài nguyên đi kèm	13

1. GIỚI THIỆU

Chương trình đào tạo AI-Augmented Software Development dành cho đội ngũ lập trình viên đang sử dụng AI trong công việc hằng ngày, với mục tiêu nâng cao năng suất và chất lượng phát triển phần mềm thông qua việc khai thác AI một cách bài bản.

Trong vài năm trở lại đây, AI coding agent đã trở thành một phần thường trực trong workflow của nhiều lập trình viên. Hầu hết các đội phát triển phần mềm hiện nay đang sử dụng Claude Code, Cursor, GitHub Copilot hoặc các công cụ tương tự, phần lớn ở mức tự học và chưa qua đào tạo bài bản. Tuy vậy, "có sử dụng AI" và "khai thác AI hiệu quả" là hai khoảng cách rất khác nhau. Khi thiếu phương pháp, đội phát triển dễ rơi vào các anti-pattern phổ biến: prompt thiếu chuẩn bị, không quản lý ngữ cảnh, không kiểm chứng output, không có quy trình spec-driven. Hệ quả là năng suất chỉ tăng ở bề mặt, trong khi nợ kỹ thuật ngầm tích luỹ và khó phát hiện.

Chương trình được thiết kế dành riêng cho các bạn sinh viên CNTT năm cuối đã từng tham gia ít nhất một dự án thực tế từ 2 tháng trở lên, các bạn đang thực tập doanh nghiệp, hoặc fresher mới đi làm, đã quen với Git, code review, viết test và đã bắt đầu dùng AI coding tool trong công việc hằng ngày. Đây không phải là một chương trình giới thiệu cơ bản về AI; đây là chương trình rà soát thói quen hiện tại, nhận diện các anti-pattern và đưa kỹ năng của học viên lên cấp độ chuyên sâu thông qua sáu năng lực then chốt: hiểu đúng bản chất AI, thiết kế context engineering, áp dụng workflow spec-driven, tích hợp AI vào toàn bộ vòng đời phát triển phần mềm, cá nhân hóa môi trường AI cho team, và sử dụng AI có trách nhiệm trong môi trường sản xuất.

Với 6 buổi học, khoá học kết hợp lý thuyết ngắn gọn, demo trực tiếp, thực hành theo nhóm và phân tích tình huống thực tế. Mỗi buổi khép lại bằng một sản phẩm cụ thể, từ bản tự đánh giá workflow, CLAUDE.md cho codebase, spec hoàn chỉnh và commit do AI implement, cho đến subagent, slash command và bộ metric đo lường tác động.

Khoá học được thiết kế phù hợp với bối cảnh thực tế của các đội phát triển phần mềm, từ công ty Product đến Outsourcing, nơi các lập trình viên làm việc theo nhóm dự án và phải tuân thủ nghiêm ngặt các quy định về bảo mật mã nguồn cũng như thỏa thuận NDA. Toàn bộ lab và bài tập diễn ra trên một codebase sandbox được chuẩn bị riêng, mô phỏng các đặc điểm thường gặp của một dự án thực tế: legacy code, technical debt, tài liệu chưa đầy đủ, backlog ticket với chất lượng không đồng đều. Học viên không cần và không được khuyến khích mang code khách hàng hay sản phẩm công ty vào lớp dưới bất kỳ hình thức nào. Sau khóa học, học viên có thể áp dụng các kỹ thuật đã học vào công việc thực tế trong khi vẫn tuân thủ ràng buộc bảo mật của khách hàng và công ty.

Kết thúc khoá học, người học có trong tay một bộ công cụ và quy trình áp dụng AI bài bản: CLAUDE.md và conventions cho codebase, subagent và slash command cá nhân hóa, ý tưởng tích hợp MCP với công cụ nội bộ và dịch vụ ngoài (Figma, GitHub, Jira, Sentry và tương đương), bản dự thảo policy AI cho team, và bộ metric đo lường tác động. Quan trọng hơn các sản phẩm cụ thể, người học hình thành được mô hình tư duy đúng đắn về AI và thói quen sử dụng AI có ý thức — nền tảng cần thiết để team tiếp tục cải tiến trong các năm tiếp theo, khi công nghệ AI tiếp tục phát triển nhanh chóng.

2. THÔNG TIN CHUNG

Thời lượng	6 buổi × 3 giờ = 18 giờ học trực tiếp.
Hình thức	Lý thuyết ngắn, Live Demo, Thực hành theo cặp/nhóm trên Sandbox codebase, và Q&A giải quyết tình huống thực tế.
Hỗ trợ học tập	Giảng viên đồng hành giải đáp kỹ thuật, cài đặt môi trường giữa các buổi học.
Ngôn ngữ	Tiếng Việt; giữ nguyên các thuật ngữ kỹ thuật tiếng Anh.

3. ĐỐI TƯỢNG & ĐIỀU KIỆN TIÊN QUYẾT

KHÓA HỌC HƯỚNG TỚI 2 NHÓM ĐỐI TƯỢNG TRỌNG TÂM:

- **Lập trình viên:** Muốn bứt phá năng suất, chuyển từ gõ code thủ công sang điều phối AI, làm chủ quy trình phát triển phần mềm chuẩn mực để nhanh chóng vươn lên cấp độ Mid/Senior.
- **Sinh viên hoặc thực tập sinh CNTT:** Đã có trải nghiệm ứng dụng AI vào học tập/đồ án, muốn trang bị workflow chuẩn doanh nghiệp để tạo lợi thế cạnh tranh vượt trội khi đi phỏng vấn tuyển dụng.

ĐIỀU KIỆN TIÊN QUYẾT:

Kỹ năng lập trình & Công cụ:

- Đã nắm vững kiến thức lập trình cơ bản và sử dụng thành thạo **Git** (clone, commit, branch, pull request).
- Đã từng tự tay phát triển ít nhất một dự án/module phần mềm cụ thể (đồ án môn học, đồ án tốt nghiệp, project cá nhân, hoặc dự án thực tế tại công ty).
- **Trải nghiệm với AI:** Đã từng tiếp xúc hoặc sử dụng ít nhất một công cụ AI (ChatGPT, Claude, Cursor, Copilot...). Khóa học hoan nghênh các bạn chưa có workflow hoàn chỉnh nhưng khao khát xây dựng quy trình chuẩn ngay từ đầu.
- **Quy trình:** Nắm khái niệm cơ bản về Agile/Scrum (sprint, backlog, task). Khoá học không dành thời gian dạy lại các kiến thức nhập môn này.

KHÓA HỌC KHÔNG PHÙ HỢP VỚI:

- Người chưa từng viết code, muốn học lập trình từ con số 0.
- Người chưa từng thao tác làm việc nhóm với Git hoặc chưa từng hoàn thành dự án/đồ án phần mềm nào.

4. KẾT QUẢ ĐẦU RA

Sau khi hoàn thành khóa học, học viên có khả năng:

- **Đánh giá & Phản biện:** Nhận diện các anti-pattern phổ biến khi dùng AI, ra quyết định chính xác khi nào nên dùng AI và khi nào phải tự xử lý thủ công.
- **Master Context Engineering:** Thiết kế và lưu trữ ngữ cảnh chuẩn hóa cho codebase (sử dụng CLAUDE.md, .cursorrules, hoặc Copilot Instructions).
- **Ứng dụng Spec-Driven Development:** Chuyển đổi ý tưởng/story thành spec chi tiết trước khi sinh code, áp dụng các tư duy từ framework BMad/OpenSpec.
- **Tích hợp trọn vẹn vào SDLC:** Đưa AI vào từ bước phân tích yêu cầu (ADR), viết code, review code hai chiều, sinh automated test đến refactor và viết tài liệu.
- **Mở rộng Agentic Workflow:** Tùy biến Subagent, Slash Commands và kết nối MCP (Model Context Protocol) với các công cụ ngoài (Figma, GitHub, Jira).
- **Làm chủ Chi phí & An toàn:** Phản biện lỗi ngầm (security/bugs), lựa chọn Model linh hoạt (Opus/Sonnet/Haiku) để tối ưu chi phí token, và phòng chống rò rỉ mã nguồn.
- **Xây dựng Portfolio ấn tượng (Dành riêng cho sinh viên hoặc thực tập sinh):** Đóng gói sản phẩm chuẩn AI-Augmented để chứng minh tư duy làm việc chuyên nghiệp với nhà tuyển dụng.

5. CODEBASE THỰC HÀNH XUYÊN SUỐT

Khóa học sử dụng một codebase sandbox có tên **linehaul** (dự án mô phỏng hệ thống quản lý đơn hàng logistics) làm môi trường thực hành. Codebase này chứa sẵn các đặc tính dự án thực tế: legacy code, technical debt, tài liệu chưa đầy đủ, backlog ticket rải rác.

Toàn bộ bài tập và demo diễn ra trên sandbox để đảm bảo an toàn thông tin và không vi phạm bảo mật/NDA.

Hỗ trợ Kỹ thuật: Cung cấp sẵn file cấu hình DevContainer / Dockerfile giúp học viên setup môi trường local dễ dàng chỉ bằng 1-click, tránh mất thời gian xử lý lỗi môi trường.

6. LỊCH HỌC CHI TIẾT (6 BUỔI)

Tất cả 6 buổi được trình bày chi tiết ở các trang tiếp theo.

Buổi 1	Đánh giá thói quen & Xây dựng tư duy làm chủ AI
Buổi 2	Context Engineering: Thiết kế ngữ cảnh đa nền tảng
Buổi 3	Spec-Driven Development: Phát triển Feature theo định hướng Spec
Buổi 4	AI trong vòng đời SDLC & Phản biện Output
Buổi 5	Khai thác Agentic Workflow & Tích hợp công cụ (MCP)
Buổi 6	Kiểm soát rủi ro, Tối ưu chi phí & Đo lường tác động

6. LỊCH HỌC CHI TIẾT (6 BUỔI)

Buổi 1 · Đánh giá thói quen & Xây dựng tư duy làm chủ AI

Mục tiêu: Rà soát workflow hiện tại, nhận diện anti-pattern và xây dựng mô hình tư duy chính xác về bản chất của AI Agent.

- **Brainstorming nhóm:** Khảo sát bức tranh sử dụng AI đa dạng của lớp học.
- **Bản chất kỹ thuật của AI coding agent:** Xác suất, Token, Context Window, phân biệt "Reasoning" và "Prediction".
- Phân tích 7 anti-pattern phổ biến (prompt thiếu chuẩn bị, xả sạch context, bỏ qua verify, tin tưởng AI mù quáng...).
- **Demo so sánh:** Tính hiệu quả giữa code tự phát và code có context & spec.

Sản phẩm buổi học: Bản tự đánh giá cá nhân (Baseline) & Nhận diện anti-pattern cần loại bỏ.

Buổi 2 · Context Engineering: Thiết kế ngữ cảnh đa nền tảng

Mục tiêu: Chuyển dịch tư duy từ "viết prompt hay" sang "thiết kế ngữ cảnh hệ thống". Áp dụng được trên đa dạng công cụ (Claude Code, Cursor, Copilot).

- Phân biệt Prompt Engineering vs. Context Engineering.
- Cấu trúc tiêu chuẩn của một file cấu hình ngữ cảnh (CLAUDE.md, .cursorrules, Copilot Instructions): Overview, Conventions, Do/Don't, References.
- **Kỹ thuật quản lý Context Window:** /clear, /compact, Subagent isolation, Prompt caching và cấu trúc thư mục tối ưu.
- **Thực hành theo cặp:** Thiết kế file ngữ cảnh v1 cho sandbox linehaul và kiểm thử chéo bằng bài test coding.

Sản phẩm buổi học: File ngữ cảnh v1 cho sandbox đã qua kiểm thử chéo.

Buổi 3 · Spec-Driven Development: Phát triển Feature theo định hướng Spec

Mục tiêu: Nắm vững quy trình Explore → Plan → Code → Commit để sinh code chính xác, tránh tạo thêm technical debt.

- Vì sao Spec-Driven giải quyết triệt để vấn đề code rác do AI tạo ra.
- Live demo luồng làm việc Explore → Plan → Code → Commit.
- Giới thiệu tư duy cốt lõi từ BMad Framework và OpenSpec.
- **Thực hành:** Bốc ticket từ backlog sandbox, tự viết Spec và điều khiển AI triển khai feature theo đúng Spec.

Sản phẩm buổi học: 1 bản Spec hoàn chỉnh và 1 Git Commit chất lượng do AI thực thi trên sandbox.

6. LỊCH HỌC CHI TIẾT (6 BUỔI)

Buổi 4 · AI trong vòng đời SDLC & Phản biện Output

Mục tiêu: Ứng dụng AI vào toàn bộ các khâu phần mềm và rèn luyện kỹ năng đọc/review code AI có tư duy phản biện.

- **AI trong giai đoạn phân tích & thiết kế:** ADR (Architecture Decision Record) with AI.
- **Pair-programming với AI:** Chọn lựa giữa Autocomplete mode và Agent mode.
- **Thực hành Code Review hai chiều:** Tìm các lỗi ngầm (security, logic, performance) do giảng viên cài cắm sẵn trong PR do AI sinh ra.
- **AI trong Testing & Refactoring:** TDD with AI, sinh Unit Test, Property-based test, refactor legacy code.

Sản phẩm buổi học: Nhật ký áp dụng ít nhất 3 hoạt động ngoài coding (ADR, test generation, refactor documentation) trên sandbox.

Buổi 5 · Khai thác Agentic Workflow & Tích hợp công cụ (MCP)

Mục tiêu: Cá nhân hóa môi trường AI thành các trợ lý chuyên biệt và kết nối với các công cụ trong hệ sinh thái dự án.

- **Kiến trúc AI Agent:** Agent loop, tool use, permission model.
- **Thực hành Subagent:** Viết Custom Subagent có system prompt và tool whitelist riêng (Code Reviewer, Security Auditor, Test Writer).
- **Thực hành Slash Commands & Hooks:** Tự động hóa các tác vụ lặp đi lặp lại.
- **Model Context Protocol (MCP):** Kết nối AI với các công cụ bên ngoài (Figma để đọc design spec, GitHub/Jira để thao tác ticket, Sentry đọc log lỗi).

Sản phẩm buổi học: Bộ công cụ cá nhân hóa gồm 1 Subagent, 1 Slash Command và 1 Kịch bản tích hợp MCP.

Buổi 6 · Kiểm soát rủi ro, Tối ưu chi phí & Đo lường tác động

Mục tiêu: Hoàn thiện tư duy kỹ sư: đảm bảo an toàn thông tin, tối ưu ngân sách AI và đo lường được hiệu quả công việc.

- **Phân tích rủi ro:** Prompt injection, rò rỉ Secrets/API keys, bản quyền code AI và data residency.
- **Tối ưu chi phí & Lựa chọn Model:** Chiến lược phối hợp giữa các Model (Opus, Sonnet, Haiku hoặc các model tương đương) theo từng loại task, quản lý budget token.
- **Đo lường tác động:** DORA metrics (Cycle time, Change failure rate) giản lược cho cá nhân/team.
- **Trình bày kết quả khóa học:** Các nhóm báo cáo sản phẩm sandbox / Kế hoạch triển khai.

Sản phẩm buổi học: Kế hoạch hành động 30 ngày cho cá nhân/team & Bộ hồ sơ Portfolio sản phẩm chuẩn AI-Augmented.

7. ĐÁNH GIÁ

Hai hạng mục đo bổ sung cho nhau: bài test cuối khoá đo năng lực ra quyết định và mức độ nắm bắt khái niệm, project xuyên suốt đo năng lực tích hợp nhiều kỹ thuật qua thời gian trên một codebase thật.

Hạng mục	Trọng số	Cách đo
Bài test cuối khoá	**40%**	Trắc nghiệm auto-gradable trên hệ thống examination, khoảng 30 câu trong 40 phút, ở cấp độ apply, phủ 6 learning outcomes.
Project xuyên suốt trên sandbox	**60%**	Rubric đánh giá: chất lượng spec, code quality, test coverage, AI workflow đã áp dụng, documentation.

8. ĐƯA KỸ NĂNG VÀO THỰC TẾ SAU KHÓA HỌC

Khóa học cung cấp nền tảng, bộ công cụ và một số sản phẩm cụ thể (CLAUDE.md, subagent, spec mẫu, bản phác thảo policy AI), đủ để người học áp dụng vào thực tế sau khóa học.

DÀNH CHO LẬP TRÌNH VIÊN ĐANG ĐI LÀM

- **Chuẩn hóa Codebase:** Áp dụng file ngữ cảnh (CLAUDE.md / .cursorrules) cho project thực tế tại công ty.
- **AI Champion:** Đóng vai trò nhân tố kết nối, chia sẻ các AI-patterns hiệu quả trong các buổi Sprint Review/Retro của team.
- **Quy trình Review:** Đưa AI-Checklist vào PR Template của dự án để đảm bảo chất lượng code đầu ra.

DÀNH CHO SINH VIÊN / THỰC TẬP SINH

- **Đóng gói AI-Powered Portfolio:** Đưa các dự án cá nhân/đồ án lên GitHub có đính kèm file Spec, .cursorrules và Prompt Logs chuyên nghiệp.
- **Tạo điểm nhấn CV & Phỏng vấn:** Nhận bản hướng dẫn trình bày kỹ năng làm việc với AI trong CV, giúp chứng minh tư duy làm chủ AI (Spec-Driven, Critical Reviewing) vượt trội so với các ứng viên khác.

9. TÀI NGUYÊN ĐI KÈM

- **Bộ Template chuẩn:** Spec Document (BMad/OpenSpec), ADR-with-AI template.
- **Cấu hình tham chiếu:** CLAUDE.md, .cursorrules, Subagents mẫu, MCP Server mẫu.
- DevContainer & Dockerfile chạy sẵn môi trường sandbox linehaul.
- Tài liệu hướng dẫn sử dụng MCP Specification và Best Practices for Agentic Coding.

CODEGYM

BẠN CÓ CÂU HỎI?
LIÊN HỆ VỚI CODEGYM.

Chúng tôi sẵn sàng giải đáp mọi thắc mắc về nội dung, lịch học và điều kiện tham gia khoá học AI-Augmented Software Development.

www.codegym.vn

Hotline: 0989 534 458

Email: info@codegym.vn

Địa chỉ: Số 23, Lô TT01, KĐT Mon City, Nam Từ Liêm, Hà Nội