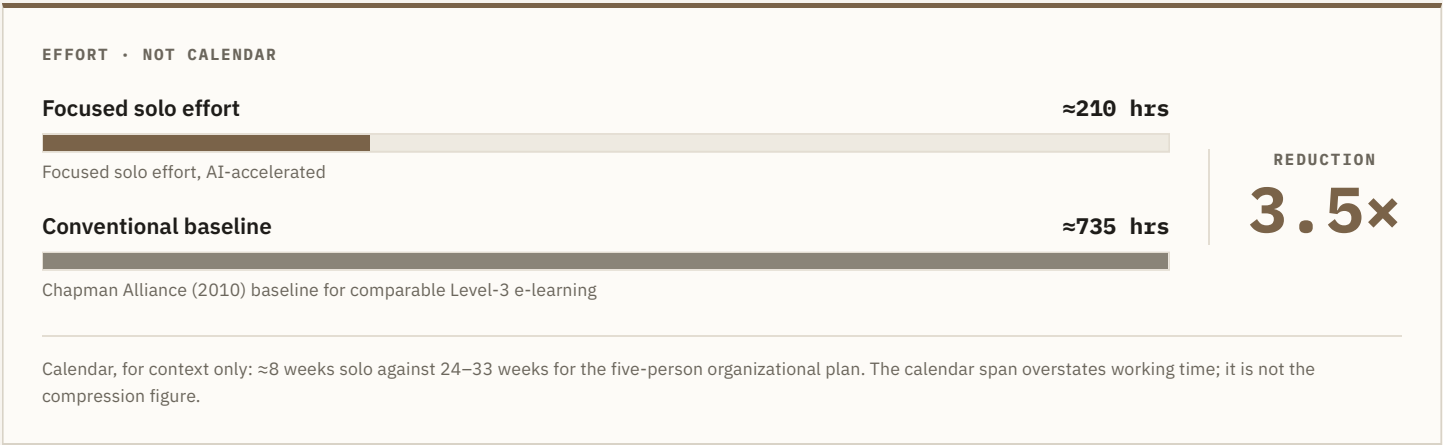


Project Timeline

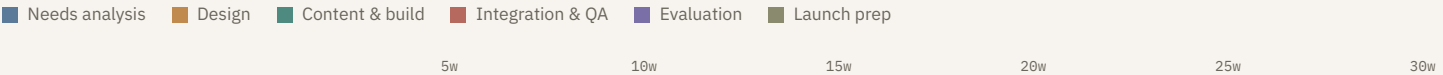
AI Literacy for the Modern Workforce

Two timelines cover the same deliverable set. The first is the retrospective record of a solo, AI-accelerated build, reconstructed from a dated process journal. The second estimates how a five-person L&D team might schedule the same scope with conventional methods. Organizations vary widely in workflow and staffing, so treat the five-person structure as one simple illustration rather than a template. The comparison shows what AI-augmented development compresses and what it does not; it is not an argument for replacing teams with AI.

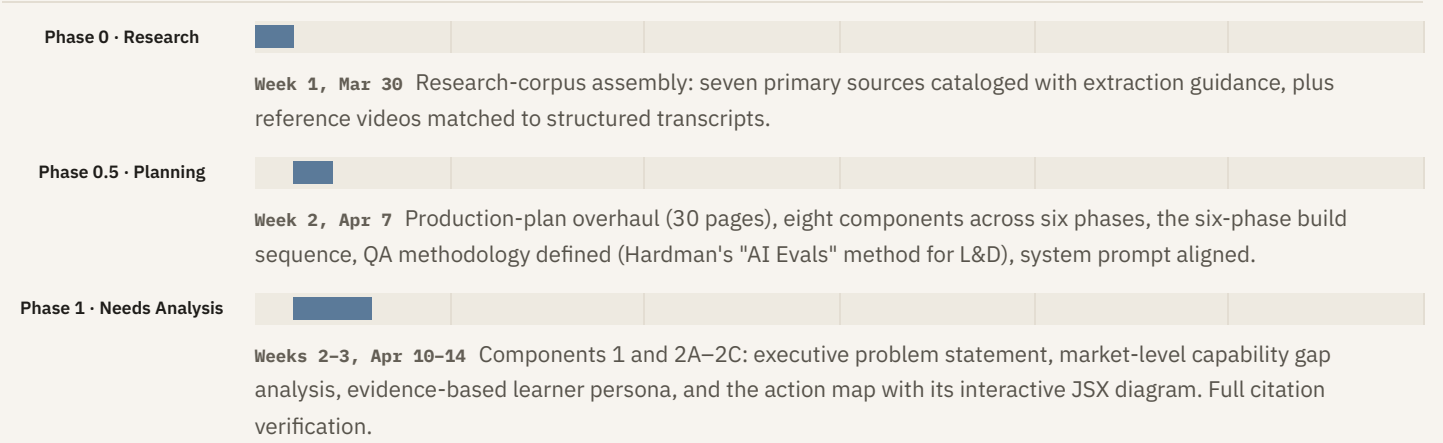
The meaningful figure is roughly 210 focused hours against a ~735-hour industry baseline for comparable Level-3 e-learning¹, a 3.5× reduction. The figure includes iterative review and improvement passes beyond the launch build. The baseline does not measure that work, which keeps the comparison conservative. The eight-week calendar span to launch overstates the working time: the early phases were part-time, and illness plus a physical injury near the start cost roughly two to three weeks of calendar time without adding to the hours.



DUAL-TRACK GANTT



SOLO BUILD ≈8 weeks



Phase 2 · Design



Weeks 3–4, Apr 18–21 Program overview (17 performance objectives), the learning-architecture diagram, the platform architecture decision document and design system, six high-fidelity mockups, five component specs, and six triple-validated JSON data files.

Phase 3 · Content + Build



Weeks 4–6, Apr 25–May 9 Four module documents (37 sections) and the custom platform: Vite + React + TypeScript, Tailwind, Recharts, 12 practice activities, 16 knowledge checks, the pre/post assessment, the analytics dashboard, three-mode navigation (later consolidated into the two-mode learner/portfolio system), and dark mode. Deployed to Cloudflare Pages.

Phase 4 · Evaluation



Weeks 7–8, May 16–19 Kirkpatrick Levels 1–4 evaluation designs, a multi-module QA audit and remediation, and the Component 7 project-management artifacts.

ORGANIZATIONAL DEPLOYMENT 24–33 weeks

Needs Analysis



3–4 weeks Research synthesis, stakeholder interviews, learner-persona development, gap analysis. Depends on SME availability, stakeholder scheduling, and data access.

Design



4–5 weeks Action map, program structure, performance objectives, design system, and component specifications, including a 1–2 week stakeholder design-review and approval cycle within the phase.

Content Development



6–8 weeks Module authoring (4 modules, 37 sections), knowledge-check and assessment items, data-file creation and validation, and 2–3 SME content-review rounds per module.

Platform Build



8–10 weeks Custom React platform: 12 interactive components, data dashboards, analytics panel, responsive design, dark mode, three-mode navigation (later consolidated to two modes), and QA cycles. Runs parallel with content development for the first ~4 weeks.

Integration + QA



3–4 weeks Content integration, cross-module testing, accessibility review, browser compatibility, a near-final stakeholder review, and bug fixes and polish.

Evaluation Design



2–3 weeks Level 1–4 evaluation instruments, the ROI model, and administration protocols. Can partially overlap the Integration + QA phase.

Launch Prep



2–3 weeks Deployment to hosting, manager orientation materials, learner communication, pilot-group selection, and baseline data collection for the Level-4 KPIs.

Both tracks run the same lifecycle. The solo build merges content development and platform build into one block and folds integration, QA, and launch prep into adjacent phases without skipping them. The organizational schedule is illustrative, derived from typical phase durations; real timings vary by team and scope.

Where the compression comes from

Four factors account for most of the difference. Approval and review cycles were eliminated: the designer and the decision-maker were the same person, so there were no phase gates. Subject-matter expertise was in-house: AI literacy was the topic of my Master's degree, so the solo build also filled the SME role a team assigns to a separate person, with no availability or review-round dependency. AI-accelerated code generation compressed the single largest workstream: the AI-assisted build took two to three weeks of active development, where a corporate build of the same platform would run considerably longer, largely because of the additional corporate audits and review gates it would have to pass. And context continuity replaced the meetings and handoffs a distributed team needs: the process journal and system prompt carried all project context.

What the team timeline buys

A team produces things a solo build cannot: multiple perspectives that challenge design decisions, distributed risk (the solo build lost roughly two to three weeks to illness and injury with no one to absorb it), and the stakeholder alignment that phase-gate approvals create. A practical approach combines the two: use AI to compress implementation and documentation, and keep analysis, design, and review human-led.

Notes

1. Bryan Chapman, "How Long Does It Take to Create Learning? [Research Study]," Chapman Alliance, September 2010, <https://www.slideshare.net/slideshow/how-long-does-it-take-to-create-learning/5198860>

© Michael Ritchot. Code licensed under the MIT License. Instructional content licensed under CC BY-NC-SA 4.0.