MARKING MY OWN HOMEWORK

I made predictions about AI coding in January. Then I tried it for myself.

Here’s my report card.

16 public claims, re-scored against a real Race Directory application taken to production

1	11	4
reworded	held	reinforced

Report card for: Andy Hickman

Class: 26AI

SUBJECT: Blog 1, Why vibe-coding will rule software

SUBJECT AREA & FEEDBACK	GRADE
Vibe-coded software will dominate production grade development *Right destination. Although the term vibe-coding has morphed a bit. Swap with "vibe-coded" with "AI-assisted" to get A-.*	B
Code 10–100x cheaper; whole systems only 2–10x *True. At the low end of those ranges in my case, but will clearly increase with experience.*	B+
Humans stay critical for intent and design *Understated. Even with latest models I needed to keep the agent on track. When I didn't, I got a 11,000 lines house of cards.*	A
File-level technical debt matters less *50,000 lines unread. "Smells", such as poor abstractions and repeated string literals, matter less and haven't bitten me yet.*	A-

TEACHER'S NOTE

Graded hardest. This is the blog the build was a direct test of.

Report card for: *Andy Hickman* Class: *26AI*

SUBJECT: Blog 3, What your software team will look like

SUBJECT AREA & FEEDBACK	GRADE
Teams shrink by roughly 5x *Will vary, but this is plausible — capped by specifying and owning, not by coding and testing.*	A−
Syntax knowledge matters far less *Held. I spent zero time worrying about syntax and library usage.*	A+
Systems thinking matters far more *The main skill my whole project definitively rewarded.*	A+
Product thinking becomes the bottleneck *The app wasn't complex enough to validate this, but I spent far more time on product thinking than I would have done in the past.*	A−
Testing is sovereign, and agent-written *~2,500 test cases; I wrote none of them and they exceeded quality expectations. Key artefact for keeping agents on brief.*	A

TEACHER'S NOTE

A team of one is thin evidence for some claims.

Report card for: *Andy Hickman* Class: *26AI*

SUBJECT: Blog 4, The future of the software industry

SUBJECT AREA & FEEDBACK	GRADE
Many software companies fail	B+
Companies don’t compete on development cost	A–
The death of the per-seat licence	A-
Software volume explodes	A–
The great reskilling	B+
Job losses stay modest	A–
Build vs buy: the pendulum swings to build *It swings – then a good many quietly swing back to buy as the true cost of software ownership becomes apparent.*	B+

TEACHER’S NOTE

Marked on conviction and industry trends since January, not on the evidence of one application!

WHAT I'D TELL A CTO

1. **Spend the majority of your review effort on design, not implementation.** The errors that cost real money are high level and compound.

2. **Budget for ownership, not just for build.** Generating it got dramatically cheaper. Owning it did not.

3. **Treat prototyping as the easy bit.** Roughly: getting it working is ~20% of the effort; getting it to production is the rest.

Andy Hickman

andy-hickman.co.uk · race-directory.co.uk