

Why Accounting Finally Gets Good Software

Agentic AI in your workday — and behind the tools being built for you

Who's talking to you

William LeFew, Ph.D.

Twenty years building data, machine learning, and AI products for startups, regulated enterprises, and government long before any of it was called AI.

Most recently, GM and VP of AI for the Tax vertical at Aiwyn: led a zero-to-one organization from vision to market-facing product, alongside tax professionals, to accuracy and audit standards the profession requires.

Before that, Senior Director of Data Science at Metabolon: drove generative AI adoption across a regulated R&D environment, eliminating thousands of hours of manual work.

Now co-founder & CTO of FirmPro. Our first product, DepreciationPro, is being used by CPA firms large and small.

WHAT WE'LL COVER TODAY

Why accounting never got great software

Agentic development

How AI reshapes a company

Live demo of DepreciationPro

PART 1

The software industry failed accounting.

Why the modern, specialized tools this profession needs rarely got built

Why deep vertical software never got built

The market is small

~50,000 CPA firms in the U.S.,
~86% of them small.

A venture-scale software company needs millions of seats. A fixed-asset tool for tax practitioners is a rounding error to them so it got a rounding-error budget.

The domain is enormous

Every feature is a corner of the Code plus regs plus state conformity plus form instructions.

Each corner needs an engineer and a CPA in a room for weeks. Vendors ration those weeks.

So you got good enough

Software written once in the 1990s, wrapped in a browser if you're lucky. More often you're logging into a remote server to reach something archaic on the other end.

So you learned workarounds.

How much tax law is there, really?

~2.4M

words in the Code

~10M

words with Treasury regulations

50

state conformity regimes

Depreciation alone: Pub 946 runs over 100 pages before you touch §168(k), §179, §280F, §1245/§1250, QIP, the mid-quarter test, or a single state decoupling statute. And it changed again in July 2025 when OBBBA made 100% bonus permanent.

One edge case, the old way

Example: an asset sold mid-year under the mid-quarter convention

1	2	3	4	5
Read	Ask	Spec	Build	Test
Engineer reads §168(d)(4)(C), Reg. §1.168(d)-1, Pub 946 tables	Books a CPA for an hour. Reschedules twice.	Writes a spec. Gets the 12.5 / 37.5 / 62.5 / 87.5% factors backwards.	Codes it, plus Form 4562, Form 4797, and the export file.	Hand-builds test cases. Misses the same-year buy-and-sell rule.

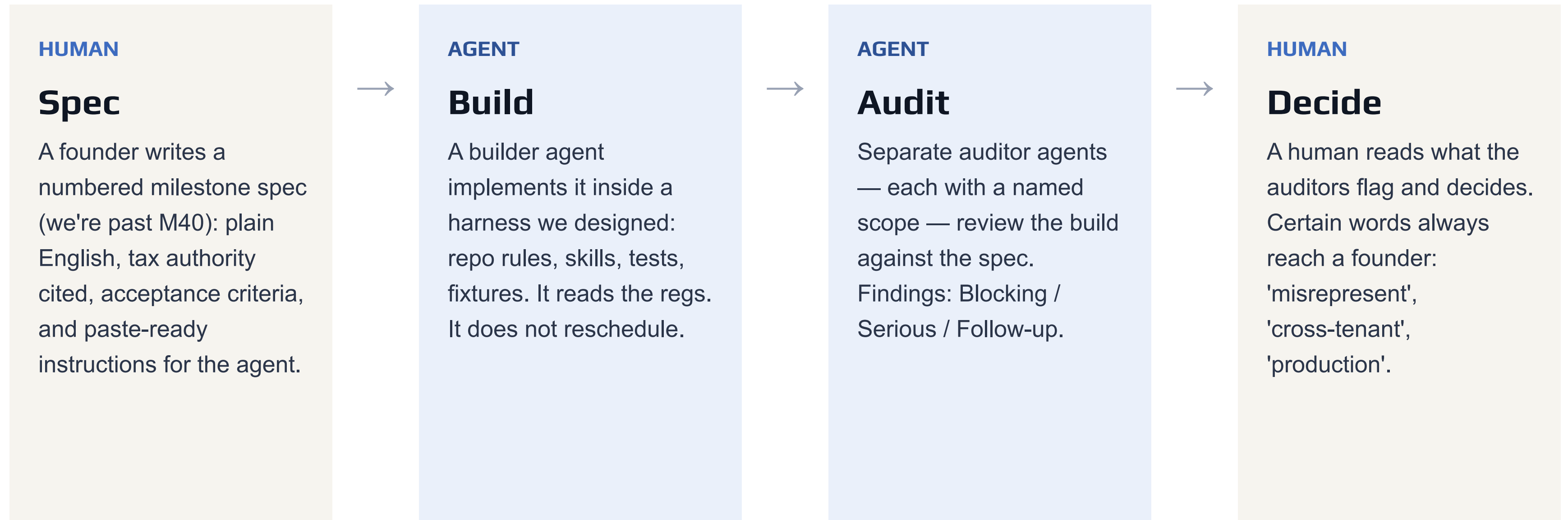
Elapsed: 2–4 engineer-weeks per edge case. Multiply by every corner of §168 and you understand why your vendor's roadmap says 'next year' every year.

PART 2

What agentic development actually looks like

Not the demo. The Tuesday.

The loop we run every day



Humans own the ends of the loop, what to build and whether it's right. The middle runs at machine speed, inside a structure humans built.

This is not vibe coding

Same model, two completely different practices

VIBE CODING

Type a wish into a chat box

Agent starts from a blank page

'Looks right' is the test

One agent does everything

Nobody remembers why

Fine for a weekend prototype

HARNESSED AGENTIC ENGINEERING

Write a spec with the authority cited and acceptance criteria

Agent starts from standing repo rules, skills, and shared context

Golden fixtures, tie-outs, and end-to-end tests are the test

Named roles: builder, auditors, housekeeper — each with a scope

Every decision written down once; drift gets audited nightly

Required for software a CPA signs a return with

The harness we built around AI

01

Standing rules per repo

Every codebase carries a rules file the agent reads before it touches anything: conventions, locked decisions, what never to do.

02

Skills

Reusable, tested procedures the agent invokes — how we build a tax-software export file, how we tie out a schedule, how we write a release note. Written once, run hundreds of times.

03

Shared context repo

One source of truth across every agent session: milestones, decisions, open questions. Nobody — human or agent — works from memory.

04

Milestone specs

Numbered, sequential, phase-gated. Each ends with paste-ready instructions so the builder starts from the spec, not a chat.

05

Auditor roles

Contract-truth, firm-scoping, positioning, retention. Each a standing prompt with a defined scope and a defined severity scale.

06

Nightly housekeeping

An agent audits the context repo every night and posts what drifted to Slack before anyone wakes up.

Where the humans are

Less typing. More deciding.

Architecting the harness

Deciding what rules, skills, and auditors exist at all. This is the CTO job now.

Writing the spec

Reading §168 and Pub 946 well enough to state what 'correct' means, with the authority cited.

Supplying the reference

A CPA hand-works the schedule the agent's 400 test cases must tie to. Scarce, valuable, human.

Deciding on the findings

Reading Blocking and Serious. Owning the call on anything that touches 'misrepresent' or 'cross-tenant'.

The same edge case, now

v64.1 — Aug 19, 2026

- Mid-quarter assets take the statutory 12.5 / 37.5 / 62.5 / 87.5% for the quarter sold
- Form 4562 carries the prorated deduction
- Bought-and-sold same year: no depreciation, no bonus, §179 election stands
- Final-recovery-year sales finish correctly
- Form 4797 gain/loss and the §1245 split follow — on screen, PDF, and Excel

v64.2 — Aug 20, 2026

- Add a mid-life asset with prior depreciation blank; the app reconstructs the history and the recapture agrees with it

~1 day

spec to production, including audit

[N]

test cases generated and tied out against
hand-worked schedules

Release velocity

v35 → v64

major versions of DepreciationPro

111 days

May 1 – Aug 20, 2026

2

human founders. Zero outsourced dev.

May 1

API, webhooks,
OpenAPI spec

Jun 3

Automated end-to-end
test pyramid

Jun 22

Vehicle §179 / §280F
edge-case review

Jul 22

GL journal entries +
QuickBooks / Intacct

Jul 25

Born-digital PDF
schedule import

Aug 20

Mid-life assets &
disposal recapture

Fast is worthless if it's wrong



Golden-fixture tie-outs

Every computation ties to a hand-worked reference schedule before it can ship. Same idea as your tickmarks.



Automated test pyramid

Every release runs log in → import → depreciate → report → export end to end. Broken pages never reach a firm. (Shipped v40, June 2026.)



Document totals must agree

A PDF asset schedule import ties to the document's own printed totals before you can commit it.



Nothing posts without a human

API and PDF imports land in a review queue. An automated import never changes a return without sign-off.



Locked decisions, written down

Rounding rule, 'zero is never omitted', API-is-truth — decided once, enforced by the auditors on every build.

PART 3

How AI reshapes the company and the new problem it creates

Two people, forty milestones, and a productivity hazard

A two-person software company

TEAM

CEO — customers, firms, partners
CTO — specs, escalations, taste
Neither of us writes most of the code.

STANDING RULES

- ▶ Founder is escalation-only
- ▶ One numbered milestone list, no departments
- ▶ One shared context repo, the source of truth for every agent session
- ▶ Every finding is Blocking, Serious, or Follow-up

AGENTS (STANDING ROLES)

Builder

Implements the milestone spec end to end

Firm-scoping auditor

Can firm A ever see firm B's data?
(Blocking if yes.)

Retention auditor

Do we keep only what a processing layer should keep?

Contract-truth auditor

Does the code do what the spec and the tax authority say?

Positioning auditor

Does UI copy overclaim or misrepresent?

Nightly housekeeper

Audits the context repo, posts drift to Slack

The new problem: hyper-productivity

When building is nearly free, you can build the wrong thing ten times faster.

Spec debt

The agent builds exactly what you wrote. If you wrote it at 11pm, that's what ships.

Review debt

Output arrives faster than any human can read it. Unreviewed work piles up silently.

Surface-area creep

Every 'while you're in there' becomes a feature. Features have maintenance costs forever.

Drift

Twenty agent sessions, each slightly different, and nobody remembers why the rounding rule is what it is.

Dealing with hyper-productivity, productively

Spec debt

Specs are the job now

- Writing the spec — with the authority cited — is where the founder's hours go. A day of spec beats a week of rebuild.

Review debt

Auditors before humans

- Agent auditors triage; humans read only Blocking and Serious. Escalation words route the dangerous stuff automatically.

Surface creep

Numbered milestones, one list

- If it isn't a milestone, it doesn't get built. 'While you're in there' becomes a Follow-up ticket, not a feature.

Drift

Locked decisions, one source of truth

- Rounding, pagination, error format, retention — decided once, written in the shared repo, enforced by the nightly audit.

This is about to happen in your firm

Everything on the last three slides translates directly.

OUR WORLD

Builder agent writes code

Auditor agents review against spec

Humans read only Blocking / Serious

Golden-fixture tie-outs

Spec is the scarce skill

Hyper-productivity → drift

YOUR WORLD

AI drafts the workpaper, the memo, the schedule

AI checks the draft against the return, the prior year, the authority

Reviewers read exceptions, not every line

Tickmarks and tie-outs — you invented this

Framing the question well is the scarce skill

Hyper-productivity → unreviewed work and inconsistent positions

PART 4

What becomes possible

Modern specialized software for accounting

What we built because we could afford to

Depreciation + cost segregation in one workflow

Nobody else does this. Cost seg is usually a separate vendor and a PDF. Here the study feeds the asset ledger directly.

Migration that respects your workpapers

Fixed Assets CS exports, born-digital PDFs, scanned schedules, prior-year 4562s — tied to the document's own totals before you commit.

Tax-software fidelity

CCH Axcess-shaped asset-detail files, method codes, disposals, and a tie-out report so you can prove it matched.

Book-to-GL, not just tax

Balanced journal entries by GL account, disposal entries composed automatically, QuickBooks and Sage Intacct-ready — and a warning before you double-post.

Edge cases as features

Vehicle §179 / §280F review, mid-quarter disposals, mid-life asset reconstruction, state conformity. The 20% path, shipped.

What you should now demand from every vendor

1

Show me the tie-out.

If a number can't be traced to a schedule you can hand-work, it isn't a number.

2

Show me where a human signs.

Automation that changes a return without a review queue is a liability, not a feature.

3

Show me the AI switch.

Firm-wide on/off, with a plain-English disclosure of what AI touches and what it doesn't.

4

Show me the edge cases.

Ask about mid-quarter disposals, §280F, state decoupling. 'Next year' is no longer an acceptable answer.

DEMO

DepreciationPro

Ten minutes, one client, start to finish

1

Import

Drop in a prior-vendor asset schedule (PDF). Watch it tie to the printed totals.

2

Review

The 'Needs review' vehicle. Confirm the GVW, watch §179 update live.

3

Run

Depreciate. Form 4562, Form 4797, state forms — one workspace.

4

Cost seg

Reclassify a building. See the first-year bonus impact in the ledger, not a PDF.

5

Export

Tax-software asset-detail file + journal entries for the GL. Tie-out report.

Watch for the four demands: tie-out, human sign-off, AI switch, edge cases.

DepreciationPro

The tools are finally being built for you.

Ask more of your vendors. Ask more of your AI.
And hold both to the standard you hold yourself to.

Bill LeFew, Ph.D. · bill@firmpro.ai · 815-382-1806

Andy Schrader, CEO · andy@firmpro.ai

FirmPro 

See DepreciationPro for yourself.

Your first client, entity, and return never expire.

