A Real Guide to Legal Automation: When to DIY, When to Pay

For solo practitioners, small firm lawyers, and anyone who doesn't have an IT department

The Core Question

You're doing something repetitive and annoying. Should you:

- Buy software
- Build something yourself (with AI help)
- Pay someone to build it

Here's the framework to decide.

The Five-Minute Decision Process

Question 1: How often?

Rarely (less than monthly)?

→ Just do it manually. Don't optimize things you barely do.

Regularly (weekly or more)?

→ Keep reading.

Question 2: Does it already exist?

Google: `[your task] software for lawyers`

Found something that looks close?

→ Try it. If it works for your actual cases, use it. Done.

Nothing exists?

→ Keep reading.

Question 3: Can you explain it simply?

Write one sentence: "Take [this] and turn it into [that]."

Can't write this clearly?

→ Not ready to automate. Do it manually 10 more times until you understand your own process.

Can explain it?

→ Keep reading.

Question 4: The Critical Question Everyone Forgets

Will anyone else need to use this?

If yes → **Complexity just increased 10x.**

This is not an exaggeration. Building something just for yourself is fundamentally different from building something others will use.

Just for you:

- Can be ugly
- Can break sometimes (you'll fix it)
- Can have weird quirks (you'll remember them)
- Can assume you know what you're doing
- No documentation needed
- No support needed

Question 5: The Complexity Check

How many of these apply?

- Files are messy (scans, handwriting, weird formats)
- Involves deadlines, court rules, or jurisdictional differences
- Need to pull data from multiple different systems
- If it breaks, real harm could occur
- Can't easily tell if output is correct

Three or more?

→ Buy software or hire someone.

Fewer than three?

→ You might build it yourself.

If You Decide to Build

First principle: You should not be writing code.

I mean it. You're a lawyer. Writing raw code from scratch is almost never the right answer.

Instead, use tools that do the coding for you:

AI Coding Assistants

These tools let you describe what you want in plain English and they write the code:

- **Claude** (this) + artifacts for simple tools
- **Cursor** - AI pair programmer
- **Lovable** - describe a web app, get a web app
- **Claude Code** - build command-line tools
- **Bolt.new** - similar to Lovable

If You Decide to Buy

How to Choose

- Try no more than three options
- Test with real work, not examples
- Use each for at least a few days
- Pick whichever annoys you least

"Least annoying" beats "most features."

Red Flags

- Can't try it free first
- Requires "implementation consulting"
- Includes tons of features you'll never use

If You Decide to Hire Someone

When Hiring Makes Sense

- DIY got 60-70% there but you're stuck
- Specific enough that no product exists
- You'll use it many times
- Just for you (remember: building for others is 10x harder)

How Not to Get Burned

Do this:

1. Share your one-sentence description
2. Give 3 real example files
3. Ask them to solve just those 3 examples first

The Real Decision Matrix

Situation	Do This	Not This
Rarely do this task	Manual	Anything else
Common task, product exists	Buy it	Build it
Unique task, just for you	Try AI-assisted build	Raw coding
Unique task, others will use	Buy or hire	Build yourself
Messy data or high stakes	Buy professional tool	DIY
DIY failed in 90 minutes	Stop and buy/hire	Keep trying

Real Examples

Example 1: Contract Assembly

Task: Generate contracts from intake data

Wrong: Learn Python, build system, spend weeks

Right: Used Clio + integration, done in hours

Example 2: Deadline Calculator

Task: Calculate all deadlines from one triggering date

Wrong: Buy complex case management software

Right: Google Sheet with formulas, took an evening

Example 3: Brief Formatting

Task: Convert memos to court-formatted briefs

Wrong: Try to write a Word macro

Right: Asked Claude to make a script, refined it, works perfectly, just for me

Example 4: Client Portal

Task: Share documents with clients

Wrong: Built custom portal for clients to log in

Right: Spent months fighting bugs, clients confused, finally bought Clio Manage

Lesson: Building for others ≠ building for yourself

Common Mistakes

Mistake 1: "I'll learn to code properly"

You won't. You don't need to. Use AI assistants to translate your ideas into code.

Mistake 2: "This tool does everything"

Focused tools beat Swiss Army knives.

Mistake 3: "I'll build it future-proof"

Build for today's problem. Expand later if needed.

Mistake 4: "It needs to be perfect"

80% automated is better than 100% manual.

Mistake 5: Forgetting the deployment question

Three Principles That Matter

1. Start with the simplest possible solution

Is this already solved by:

- A feature in software you already use?
- A basic spreadsheet?
- A text expander?
- A simple email filter?

If yes, use that. Don't over-engineer.

2. Time-box everything

Don't let exploration become infinite.

- Trying a tool? Give it a few days max.

Your Action Plan

Today:

1. Pick one annoying repetitive task
2. Time yourself doing it manually once
3. Decide: buy, build, or hire?

This Week:

- If buying: try 2-3 options
- If building: 90-minute test with AI assistant
- If hiring: find 3 candidates, give test project

This Month:

- Actually implement one thing

Final Thoughts

Most lawyers overthink this decision.

The framework is simple:

1. Does a product exist? Try it.
2. If not, can you describe it clearly?
3. Will others use it? If yes → buy or hire.
4. If just for you → try AI-assisted build for 90 minutes.
5. If that fails → buy or hire.

The goal isn't to become technical. The goal is to stop doing annoying work.

You don't need to automate everything. You need to automate the right things.

Start small. Start today. Pick one task and eliminate it this week.

Tools Worth Knowing About

For building (AI-assisted):

- Claude (for simple scripts/tools)
- Cursor (for slightly more complex projects)
- Lovable / Bolt.new (for web apps)

For buying:

- Check Lawyerist.com for honest reviews
- Ask other lawyers in your practice area what they use
- Try free trials with your actual work

For hiring:

- Upwork for simple projects