
Build, Buy, or Fix What You Already Have?

A practical decision guide for lawyers and law firm teams.

Start with one recurring operational problem, then compare three options: improve software you already own, buy a new subscription, or build a small custom tool.

Your outcome is a shortlist and one next step to investigate. A custom build is one option, not a promised replacement for every subscription.

Use generic task descriptions. Keep client names, privileged material and matter details out of this planning worksheet.

1. Describe the problem before the tool

Task or handoff that causes friction: _____

Who does the work? _____

How often does it repeat? _____

Minutes per occurrence, including review: _____

What starts the task, and what is the finished result? _____

Record the current process and the specific issue: repeated entry, missing information, hard-to-find status, inconsistent documents or follow-up that is difficult to track. These are prompts for your own observations, not claims about every firm.

Baseline monthly effort = occurrences per month x minutes per occurrence / 60.

Use your own measured or clearly labeled estimated inputs. This baseline is not a prediction of savings.

A useful first problem has a clear owner, a repeatable workflow and a result someone can check.

2. Check tools you already own

List the software used in this workflow and the relevant subscription tier. Ask the vendor or administrator about templates, forms, integrations, exports and automation features before buying or building something else.

Existing tool or feature to investigate: _____

Does your current plan include it? _____

What setup or upgrade is required? _____

Who can configure and maintain it? _____

Identify gaps in the workflow rather than assuming a missing feature requires a new application. A better template, a changed handoff or a supported integration may be enough.

Separate confirmed capabilities from questions you still need answered. Keep a source or contact for each estimate.

3. Define the smallest useful version

Write up to five must-have functions. For each, record the input, output, user and person who checks the result.

Possible examples to investigate: a consultation-preparation form, an internal task handoff, an approved document assembly workflow, or a simple status view. These are ideas, not delivered features or performance promises.

Must-have functions: _____

Nice-to-have functions to defer: _____

Required data source and supported integration: _____

Users and access levels: _____

Owner for updates and support: _____

Check export options, backups, data retention, vendor access and what happens if a dependency fails. Your firm reviews the requirements for the intended use.

Avoid recreating a large specialist system when a narrow tool or improvement would solve the actual problem.

4. Compare cost and responsibility

Use the companion worksheet to compare: improve existing software; new subscription; custom build. Enter quotes or your own labeled estimates. Leave unknown costs blank and record the question to resolve.

Direct first-year cost = one-time setup + one-time integration + annual maintenance/support + 12 x (monthly subscription or hosting + monthly usage charges). Include each cost once; if a quote bundles support, do not add it again. Add other quoted annual costs separately.

Track staff time separately: monthly setup, operation, review and correction effort, with the same assumptions for all options. If you value that effort in dollars, use a firm-selected hourly rate and label it as an estimate, not an invoice or savings promise.

Also compare ownership, updates, integration limits, portability and exit costs. A lower subscription fee alone does not establish that a custom tool is cheaper.

Quote source and date: _____

Unresolved cost or responsibility: _____

5. Choose a small pilot and next step

Option to investigate first, and why: _____

Pilot task and owner: _____

Reviewer and approved test inputs: _____

Start date, review date and spending limit: _____

What useful output should look like: _____

Fallback or rollback owner: _____

Test with synthetic or approved information. Compare the complete task, including preparation, review and corrections. Record errors, extra effort, costs and whether users can act on the result. Keep the current process available while testing.

At review, decide: continue, adjust, choose a different option, or stop. No invented readiness score or guaranteed savings is needed.

Send Chris a short description of the recurring task you want to improve:

<https://www.legalsearchmarketing.com/contact?service=custom-tools>

Read the article and watch the explanation:

<https://www.legalsearchmarketing.com/blog/custom-tools-lawyers-build-or-buy>

Tool development, integration and ongoing maintenance are scoped for the chosen solution.