

Construction Software Buyer's Checklist

10 questions to ask before
hiring a software team

For contractors, builders and construction operators comparing custom software teams.

Made for your phone. Scroll down: one question per screen.

Ask out loud. To any vendor.

Choosing a software team is a bet on people, not only on technology. Most bad outcomes come from the same few causes: a team that did not understand the work, a scope nobody pinned down, or terms nobody read until there was a problem.

These ten questions are meant to be asked out loud, to any vendor, including us. Good teams will welcome them. Teams that avoid them are telling you something.

HOW TO USE IT

Four simple steps

- 1** Ask every question to each team you are considering, and take notes on the answers.
- 2** Ask for answers in writing where it matters: scope, ownership, costs and exit terms.
- 3** Use the "good answer" and "red flag" notes as prompts, not a pass or fail test.
- 4** Score each answer from 0 to 2 on the scorecard and compare vendors side by side.

SMALL OR MID-SIZE BUILDER?

You do not need every enterprise feature. You need software that fits how you work and a team that stays reachable. Weigh the answers with that in mind.

01

Have you built software for construction workflows, and can we talk to those clients?

WHY IT MATTERS

Construction is not generic business software. Change orders, progress billing, retainage, subcontractor payments and field updates all have their own logic. A team that has to learn this on your budget will be slow and will ask you to explain the basics.

A GOOD ANSWER

Specific examples of construction or similar project-based work, and a willingness to connect you with a past client. Ask what they got wrong the first time.

RED FLAGS

Only generic portfolio screenshots, no references, or "we can build anything".

02

Who will actually do the work, and how will I see progress?

WHY IT MATTERS

Many agencies sell with senior people and deliver with whoever is free. You should know who is building your software, and you should never have to wonder what is happening. Regular, visible progress is the best early warning against a project going off track.

A GOOD ANSWER

Named people and roles, one contact who can answer technical questions, a fixed update rhythm, a working version you can click through every few weeks, and a time zone overlap that works for you.

RED FLAGS

Heavy subcontracting you were not told about, a changing project lead, long silences followed by a big reveal, or progress reported only as percentages.

03

How will you learn our process before writing code?

WHY IT MATTERS

Most failed software projects solve the wrong problem well. Discovery is where a team proves it listens: calls with your project managers and foremen, and a look at your current spreadsheets and paperwork.

A GOOD ANSWER

A short, clearly scoped discovery phase that ends with a written description of your workflow and what the first version will and will not do.

RED FLAGS

A quote delivered after one call. A fixed price with no discovery. No questions about how your crews really work.

04

What exactly will version one include, and what is left out?

WHY IT MATTERS

Scope is where budgets break. You need to know what you get on day one and what counts as a change. A smaller first release that people actually use beats a large one that arrives late.

A GOOD ANSWER

A written feature list for the first release, a plan to ship something usable early, and a clear process for adding or swapping items.

RED FLAGS

"Unlimited revisions", a scope described only in broad terms, or no explanation of how changes are priced.

05

Who owns the code, the data and the accounts?

WHY IT MATTERS

If you pay for custom software, you should be able to keep running it if the relationship ends. Ask this before signing, not after a dispute.

A GOOD ANSWER

Written terms covering code ownership or a usable license, data export in a standard format, and hosting and domain accounts registered in your company's name.

RED FLAGS

The vendor keeps the only copy, holds the hosting accounts, or charges to hand over your own data.

06

What will it cost in total, including what happens after launch?

WHY IT MATTERS

The build price is only part of the cost. Hosting, maintenance, support, fixes, and changes continue for as long as you use the software.

A GOOD ANSWER

A separate line for setup and for recurring costs, a description of what support includes and how fast issues get a reply, and a clear price for extra work.

RED FLAGS

A single number with no breakdown, support that is "best effort", or recurring fees that can change without notice.

07

How will our people in the office and in the field actually use it?

WHY IT MATTERS

Software that crews will not open on a phone on site does not get used. Poor connectivity, gloves, sunlight and short attention spans are normal conditions.

A GOOD ANSWER

Mobile-friendly screens designed for quick updates, an approach to weak signal, and a plan to test with real users from your company before launch.

RED FLAGS

A desktop-only design, no user testing, or training treated as an afterthought.

08

How does it connect to the tools we already use?

WHY IT MATTERS

You probably have accounting software, spreadsheets, email and messaging tools that will not disappear. Double entry between systems is how adoption fails.

A GOOD ANSWER

A list of the integrations that matter, an honest view of which are easy and which are hard, and a way to import your existing data.

RED FLAGS

"We can integrate with anything" with no details, or a plan that requires everyone to re-enter their history by hand.

09

How do you handle security, access and backups?

WHY IT MATTERS

Your software will hold contracts, pricing, client details and payment information. Different roles need to see different things. A lost database is a serious problem.

A GOOD ANSWER

Role-based access, regular automated backups with a tested restore, encrypted connections, and a clear answer on where your data is stored and who can reach it.

RED FLAGS

No backup plan, shared logins, or an inability to explain who has access to your production data.

10

What happens if it does not work out?

WHY IT MATTERS

Good partners are comfortable discussing the exit. How a team talks about ending a project tells you how they will treat you during it.

A GOOD ANSWER

Contract terms that let you stop at a milestone, a handover process with documentation, and the ability for another developer to take over.

RED FLAGS

Long lock-in terms, penalties for leaving, undocumented code, or discomfort when you raise the topic.

Questions you can ask us

SolSoft is a software studio specialized in construction, based in Asunción, Paraguay. We build custom CRMs for construction companies, covering clients, proposals, progress billing, project tracking and payment follow-up, set up around how each company already works.

We would rather you ask these ten questions of us than skip them. If you want to compare notes on your own situation, tell us about your company and what you are trying to fix, and we will get back to you.

If it is not a fit, we will say so.

Score vendors interactively:

solsoft.dev/resources/checklist/

**Use the contact form
at solsoft.dev**

This checklist is general guidance and not legal or financial advice. Review contract terms with a qualified professional.