

What a real build takes

I am not a developer, and over 1,500 updates have gone into the software my company runs on, built with Claude. So here is the honest version. AI made writing code cheap. It did not make building software easy. This guide is what actually decides whether custom software works, drawn from our own build.

TIME Fifteen minutes to read

YOU NEED A process you are thinking about turning into software · Fifteen minutes for the readiness check

What AI changed, and what it did not

WHAT AI MADE EASY

- Writing the code
- A first version you can click through
- Screens that look finished
- Changes in minutes instead of weeks

WHAT IT DID NOT

- Knowing what to build
- Who is allowed to see what
- Moving years of your data in safely
- Making sure a change does not break what already worked

A demo that looks finished takes an afternoon now. Software a team can run a business on still takes judgment, testing and time. Most of the cost is in the part you cannot see.

The eight things underneath

Screens are what you see. These are what you are actually paying for, and every one of them came up in our own build.

01

Who can see what

Your salesperson should not see payroll. A client should see their project, not the next client's. Every screen, report and export has to respect that, and it has to be tested.

02

Your existing data

Years of spreadsheets, a CRM, email history. Moving it in without losing or duplicating a record is a project of its own.

03

The tools it talks to

Email, calendar, accounting, your CRM. Two-way connections fail quietly: a record updates in one place and not the other.

04

The ugly cases

The client with two addresses, the refund, the duplicate lead, the person who double-clicks. Software lives or dies on the cases nobody puts in a demo.

05

Security

Two-step sign-in, what happens when someone leaves, where passwords and keys are kept. The day you need it is the wrong day to add it.

06

Phones

Your team uses it in a car, a hallway, on a job site. If it only works on a laptop, it does not work.

07

When it breaks

Backups, alerts, and who fixes it at 9pm on a Friday. Everything breaks eventually. What matters is how fast you know and how fast you recover.

08

Every change after

Each update can quietly break something that worked last week. Checking for that is most of the ongoing work, and it never ends.

From our own build

Real things that happened while building Origin's software. AI wrote the code every time. It caught none of these.

THE FEATURE THAT VANISHED

One update quietly removed an entire feature, and every automatic check still passed, because the change took the feature and its tests out together. Nothing failed and nothing warned us. Now every change is checked against what already worked.

THE DOUBLE CHARGE

A button clicked twice in quick succession could start the same paid job twice. The fix was not a better button. It was a rule in the database that makes a second copy impossible.

THE FORM THAT SAID NO

A security check on our website quietly rejected every new signup. The pages looked fine. We caught it the same day because we watch the numbers, not just the screens.

THE FIRST SIGN-IN

On a brand new account, two setup steps could run at the same moment and create duplicate records, with an error on the very first screen a client saw. It only happened sometimes, which is the hardest kind of problem to find.

None of these show up in a demo. All of them show up in month two.

Are you ready for custom software?

Tick each one that is true today. Six or more, and a build is worth a serious conversation. Fewer, and the best first step is usually fixing the process, which is often faster and cheaper.

☐ I can describe the process step by step, and so can the person who does it.

☐ It happens every week, not every quarter.

- | | |
|---|--|
| ☐ We have tried off-the-shelf tools, and the gap is specific to how we work. | ☐ I know what it costs us today when this goes wrong. |
| ☐ I know who will own it inside the business once it exists. | ☐ The data it needs lives somewhere we can get to. |
| ☐ I can say in one sentence what done looks like. | ☐ We will test it on real work before we trust it. |

Seven questions to ask anyone who builds it

An agency, a freelancer, or someone on your team with Claude. The answers tell you whether they have done this before.

- 1 How will you move my existing data in, and how will I know nothing was lost?
- 2 How do you control who can see what, and how do you test it?
- 3 What happens when an update breaks something that used to work?
- 4 How do you test on a phone, with my real records?
- 5 When it goes down, how fast will I know, and who fixes it?
- 6 What does it cost to run and maintain each month after launch?
- 7 Can I change it later without you?

Bring a brief to the first conversation

Paste this into Claude before any first call. The one-page brief it writes saves hours in the meeting, including one with us.

THE BRIEF PROMPT

I am thinking about building software for my business to handle: [one sentence].

Interview me one question at a time: who will use it and on what device, how the work happens today step by step, where it breaks, what it costs when it goes wrong, what data it needs, and what done looks like.

Then write a one-page brief: who uses it, what it replaces, the steps it covers, what done looks like, what it must never get wrong, and what is out of scope. Plain language. Do not suggest technology.

Want a partner who has already made these mistakes?

The eight things underneath are what we do every day, starting with our own platform. Bring your brief to a call, or take the Origin Diagnostic first.

joinoriginos.com/start