

L O A D B E A R I N G C O D E

The Business Owner's Software Budget Guide

Four pages on what to clarify before you ask for a price, what the bands actually look like, how to work out whether the build pays for itself, and what to argue about in a quote.

Christopher McGrath, senior developer, 20+ years
Software your business can stand on

Answer these before you ask anyone for a price

A quote is only as good as what it was written against. Every item below that you can answer turns a guess into a number. Every one you cannot is where a project goes over.

☐ **What breaks today**

Describe the problem in operational terms, not software terms. Who does what, how often, and what goes wrong.

☐ **Who uses it**

Count the people and name the roles. A tool for two people and a tool for forty are different projects.

☐ **What it has to connect to**

List every system that holds data this tool needs. Accounting, CRM, email, shipping, the spreadsheet on the shared drive.

☐ **What happens on day one**

Existing data has to come from somewhere. Migration is real work and it belongs in the quote.

☐ **What good looks like**

One number you would use to judge this in six months. Hours saved, errors caught, orders processed per person.

☐ **What is out of scope**

Write down what you are deliberately not building yet. This is the single most useful sentence in any brief.

☐ **Who decides**

Name the person who signs off. Projects stall on approval far more often than on code.

☐ **When it matters**

A real deadline with a reason behind it, or no deadline at all. Invented urgency costs money.

What projects like yours usually cost

US small-business work, fixed price, built and delivered by one senior developer. Bands, not quotes. Use them to sanity-check what comes back, not to budget from.

One workflow, replaced	\$8k - \$18k
Multi-user web application	\$25k - \$60k
Application plus integrations	\$45k - \$120k
Legacy modernisation, per phase	\$15k - \$40k

Work out whether it pays for itself

Fill this in before your first call. Most people have never done the arithmetic, and it usually settles the question on its own.

A	Hours per week your team spends on this today	_____ hrs
B	Fully loaded hourly cost of the people doing it	\$_____
C	Annual cost of the manual process (A x B x 52)	\$_____
D	Money lost per year to errors, rework, missed billing	\$_____
E	Total annual cost of doing nothing (C + D)	\$_____
F	Quoted build cost	\$_____
G	Payback in months (F divided by E, times 12)	_____ mo

Anything under 18 months is a straightforward yes. Eighteen to thirty-six months is a judgement call about growth. Past that, either the scope is too big or the problem is too small, and both are worth saying out loud.

Ten questions to ask before you sign

- 01 What did you decide to leave out of this scope, and why?
- 02 What is the riskiest part of this build?
- 03 Who writes the code? Give me a name.
- 04 What happens if it takes you longer than you thought?
- 05 What does handover include? Documentation, accounts, source code?
- 06 Who owns the code when we are done? Put it in writing.
- 07 What does year two cost me if I never call you again?
- 08 What would you need from us to keep this on schedule?
- 09 Which part of this would you push back on if you were me?
- 10 Show me something you built that is still running five years later.

Red flags in a quote

A number with no scope under it Nothing to hold anyone to later.

Hourly, no cap, no estimate That is an open tab, not a price.

No discovery phase The quote will be rewritten halfway through.

No mention of testing or handover You will pay for both as change requests.

Vague about who does the work Ask for a name, not an agency.

Silent on code ownership If it is not in writing, assume it is not yours.

Tell me what's breaking.

Thirty minutes, no pitch. I'll tell you honestly whether this is a project worth doing, and roughly what it would take.

[LOADBEARINGCODE.COM/CONTACT](https://loadbearingcode.com/contact)