



Writing a brief that holds up

The questions to settle before asking for a quote — and the ones better left open.

Who it is for: You are about to approach providers and don't know how far to spell out what you need.

Describe a problem, not a solution

The most common reflex is to write “I want a mobile app with a dashboard.” That is already a solution, and it closes the door on every other one — including some that cost a tenth as much.

Write instead what is going wrong today: who loses time, on what, how many times a week, and what that produces at month end. A provider reading that can propose something other than what you had in mind. That is precisely what you are buying.

Keep the solution-shaped version for the end of the document, under an honest heading: “what we had in mind.” It belongs there — it reveals your implicit constraints — but it should not open the text.

What must be settled before you consult

These are not execution details: they change the architecture, and therefore the price. Leaving them open guarantees a quote that means nothing, followed by a renegotiation.

- ✓ Who will use the tool, and on what hardware? Phone use in the field has nothing in common with desk use.
- ✓ How many people at once, at peak times?
- ✓ What data comes in, where from, and does a file or system already hold it?
- ✓ Who is allowed to see what? Access levels are one of the main sources of complexity.
- ✓ Is there a deadline imposed from outside — a legal obligation, a commercial season, an event?
- ✓ What happens if the tool is unavailable for half a day? The answer sets the level of rigour, and its cost.

What is better left open

Technologies, unless there is a real constraint. Imposing a language because you have heard of it rules out capable providers and improves nothing. That said, if your own IT team will have to take over the code, say so: that is a real constraint, and one sentence justifies it.

Screen-by-screen detail. A brief that describes every button freezes decisions before anyone has watched a user try it. Describe what the person must be able to accomplish, not where to click.

The number of days. Estimating that is the provider's job; imposing it means choosing the answer before asking the question.

The section almost nobody writes

What is out of scope. A brief says what must be done; it rarely says what must not, and that is where most end-of-project disagreements come from.

Write down in plain terms what you are not expecting: migrating historical data, training staff, hosting, maintenance after delivery, writing the content. Each of those lines costs money. Naming them, even to exclude them, stops them from turning up mid-project as things that were obvious — to you only.

Length

A few pages are enough for most projects. A fifty-page document is not more precise: it is merely longer to read, and nobody will reread it when a question actually comes up.

The right test is not volume but this: can a provider who does not know you ask three relevant questions after reading it? If their questions are about things you thought you had written, the document is not too short — it is badly ordered.