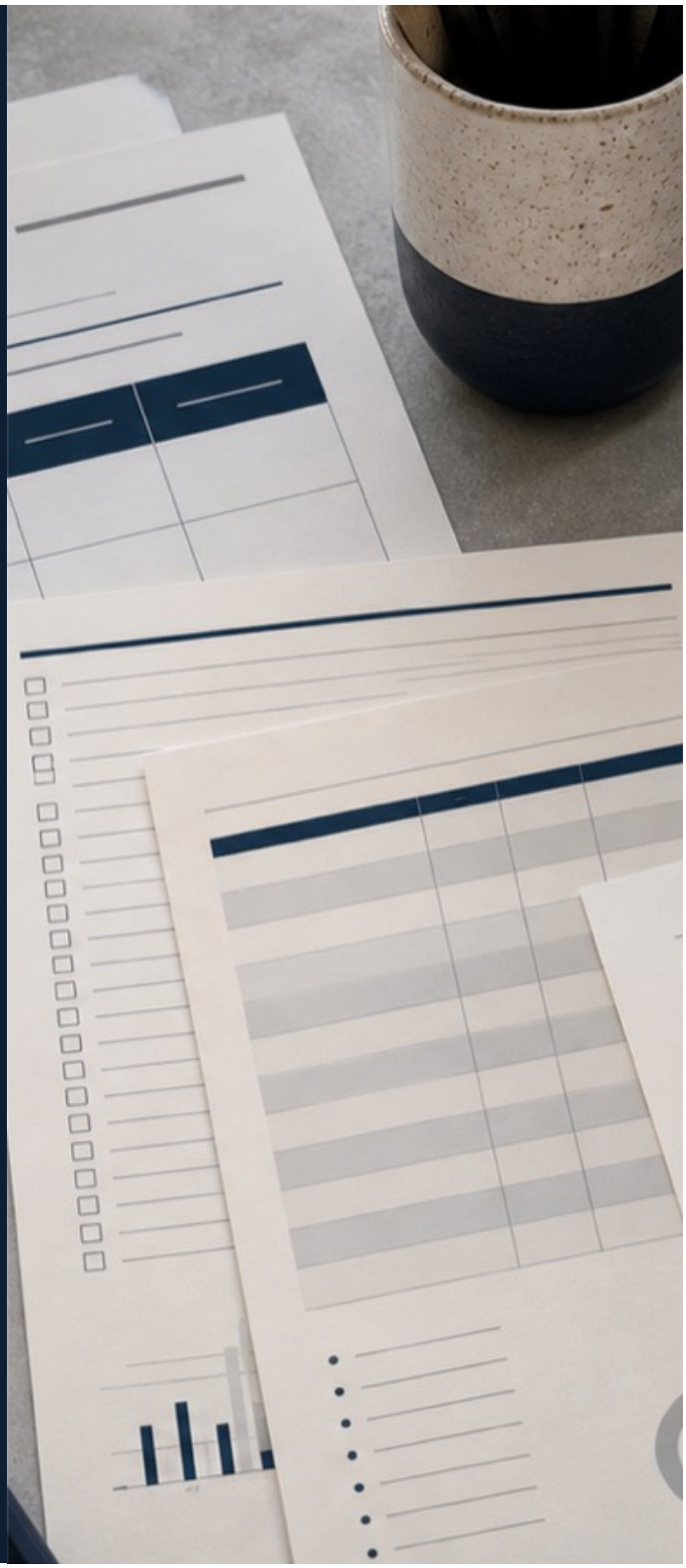
Rangefront Labs

Template · Free resource

Custom Software Project Brief Template

The difference between a quote that's accurate and one that doubles halfway through is almost always the brief. This template walks you through what a developer actually needs to know: describe the problem properly an...



Updated May 2026

https://rangefrontlabs.com.au/resources/software-project-brief-template/

Built in Toowoomba. Working across Australia and internationally.

When a software project goes over budget, the cause is rarely the code. It's that the brief described a *solution* ("build us an app with a dashboard") instead of a *problem* ("our techs waste an hour a day chasing job details over the phone"), so everyone agreed to something different without realising it.

A good brief fixes that before a line is written. It doesn't need to be long or technical. It needs to be clear about the problem, the people, and what success looks like. Get those right and the quotes you receive get sharper, the proposals get more honest, and the build has a much better chance of landing on budget.

This template covers what a developer actually needs from you. Fill in the brackets, and you'll have a brief worth sending.

The template

Replace everything in **[square brackets]**. Short and honest beats long and vague. If you don't know an answer, write "not sure yet". That's useful information too.

Project brief: [working name]

1. In one sentence. [What is this, in plain language? "A booking system for our mobile detailing business."]

2. The problem. [What's wrong today, and what does it cost? Be concrete: time, money, missed work, frustration. Why now?]

3. Who will use it. [The main groups of users and roughly how many. e.g. "8 office staff, 20 field techs, and our customers booking online."]

4. What success looks like. [How will you know it worked? Aim for something measurable: "techs stop ringing the office for job details", "we cut quoting time from a day to an hour".]

5. Must-have features. [The things it can't launch without. Keep this list honest: every item is cost and time.]

6. Nice-to-have features. [Things you'd love but could live without at launch. Separating these from must-haves is the single most valuable thing in this brief.]

7. What it must connect to. [Existing systems and data: accounting (Xero, MYOB), CRM, payment, email, spreadsheets, anything it has to read from or write to.]

8. Practical requirements. [Where does it run: web, phone, both? Who needs to log in? Any security, privacy or compliance obligations? Roughly how many users and how much data?]

9. What exists today. [Current process, tools and any system this replaces or sits beside. Screenshots or a spreadsheet export help enormously.]

10. Budget and timing. [A budget range and any real deadline. "We don't have a number" is fine, but a range lets a good partner shape scope to fit instead of guessing. A real deadline ("before peak season in October") matters; "ASAP" doesn't.]

11. Explicitly out of scope. [What this project is *not* doing. Naming this prevents the slow creep that blows out timelines.]

12. Decisions and open questions. [Who signs off? Who's the day-to-day contact? What are you still unsure about or worried about?]

How to use it

1. **Lead with the problem, not the solution.** Resist describing screens and buttons. Describe what's broken and let the people you're hiring propose the how. That's what you're paying them for.
2. **Be ruthless splitting must-have from nice-to-have.** This one section decides your budget more than anything else. Everything is "important" until it has a price next to it.
3. **Give a budget range.** It feels safer to hide it; it isn't. A range lets a good partner design something that fits, and it filters out anyone who'd rather sell you the maximum.
4. **Attach the messy reality.** A screenshot of your current spreadsheet, a sample invoice, a photo of the whiteboard process: these tell a developer more than three pages of prose.

Common mistakes this avoids

- **Specifying the solution instead of the problem.** You hire builders to solve the problem; tell them what it is.
- **No budget, no deadline.** Without either, every quote is a guess and none of them are comparable.

- **One giant feature list with no priorities.** If everything's essential, nothing can be cut when reality bites, and reality always bites.
- **Hiding the awkward bits.** The legacy system with no API, the data that's a mess, the hard deadline. These don't go away by leaving them out. They just surface later as a change in price.

A clear brief gets you comparable quotes and a partner who understood the job before they started. When you've got yours drafted, send it our way. Reading briefs properly and quoting them honestly is how we build software.

Need this adapted to your organisation, systems or data? Book a discovery call: https://rangefrontlabs.com.au/contact/