Software Project Brief Template

A practical workbook to define your requirements, brief software developers and compare quotes on a like-for-like basis.

Use this before you request quotes.
Complete what you can, write “Not sure” where you need advice, and send the same brief to every development company. The goal is to help each provider price the same problem and scope.

What this workbook helps you do

- Explain the business problem you want the software to solve.
- Describe users, workflows and the features that matter most.
- Identify integrations, data, privacy and security considerations.
- Set an indicative budget, timeline and measurable acceptance criteria.
- Ask each developer for a comparable proposal and assess total cost—not just the headline price.

Project details

Business name	
Project name	
Prepared by / role	
Date prepared	
Preferred contact details	

How to use: This is a working template, not a technical specification. You do not need to know the technology or have every answer. A suitable development partner can help clarify requirements during discovery.

1. Project overview

Give a developer a quick picture of your business, the project and the outcome you want. Keep this section concise; you can add detail in the sections that follow.

Industry / business type	
What does your business do?	
What are you planning to build?	
Who will use it?	
Why are you considering this project now?	

2. The business problem

Describe what happens today and where the process breaks down. Start with the problem rather than assuming a particular solution is required.

What is not working today?	
How do you handle this process now?	
Who is affected by the problem?	
How often does it happen?	
What does it cost in time, errors, delays or lost work?	
What happens if nothing changes?	

Helpful prompt: Instead of “We need a CRM”, explain “Our team tracks enquiries in spreadsheets and email. Follow-ups are missed, and managers cannot see which opportunities are close to closing.”

3. Goals and success measures

Write outcomes that can be checked. Add a current baseline if you know it; your target can be refined with the developer.

Goal / outcome	Current baseline	Target / measure	How will it be measured?
Example: reduce manual quoting time	2 hours per quote	Reduce time per quote	Time tracking / system report

4. Users and roles

List each user group and the actions they need to perform. Include approximate numbers and whether users work in the office, remotely or in the field.

User group / role	Approx. number	What must they do?	Access / restrictions
Administrator			
Manager / supervisor			
Staff / operator			
Customer / partner			
Other			

- Some users will access the system on a phone or tablet.
- Users may need different permissions or approval levels.
- Customers, suppliers or partners may need their own login.
- Accessibility requirements or standards need to be considered.

5. Key workflows

Describe three to five important processes in the order they happen. For each one, explain who starts it, what happens next, what decisions are made and how it ends.

Workflow 1 — name and steps	
Workflow 2 — name and steps	
Workflow 3 — name and steps	

6. Features and priorities

Separate what the first release must do from features that can wait. If everything is marked essential, it becomes harder to control cost and delivery time.

Priority	Feature / requirement	Why is it needed?	Notes / dependencies
Must-have			
Must-have			
Must-have			
Should-have			
Should-have			
Later			
Later			

MVP reminder: A Minimum Viable Product (MVP) is the smallest useful release that can deliver value and be tested with real users. It should be fit for purpose—not simply a low-quality or incomplete build.

7. Existing systems and integrations

Name the actual products you use. For each integration, explain what information should move, in which direction and when.

System / product	What data needs to move?	When / trigger	API access or notes
Accounting			
CRM / sales			
Payments			
Inventory / operations			
Other			

Existing data to migrate (type and approximate volume)	
Do you have API documentation or vendor contacts?	
Any systems that must remain in place?	

8. Data, privacy and security

Tell developers what information the software will handle and any known obligations or restrictions. If you are unsure, say so and ask the provider to identify what needs further review.

- Customer or supplier contact details
- Employee or contractor information
- Payment or financial information
- Health, identity or other sensitive information
- Business-confidential documents or records
- Location, tracking or activity data
- Data processed or generated using AI

Where is the data stored today?	
Who should be able to view or change each type of data?	
Known privacy, industry or contractual obligations	
Hosting or data-location requirements	
Backup, recovery and data-retention expectations	
Third parties that may access or process data	

9. Platforms and constraints

Topic	Your requirement / preference
Product type	■ Web app ■ Mobile app ■ Both ■ Not sure
Devices	■ Desktop ■ Tablet ■ iOS ■ Android
Browsers / operating systems	
Hosting / cloud provider	
Existing or legacy systems to retain	

Topic	Your requirement / preference
Expected users / growth over 2 years	
Other technical or operational constraints	

10. Budget and timeline

Indicative budget range (AUD)	
Preferred project start date	
Target first-release date	
Is the deadline fixed or flexible?	
Reason for any fixed deadline	
Is phased delivery acceptable?	

11. Acceptance criteria and delivery

Explain how you will decide whether the delivered software meets the agreed requirements. Criteria should be observable and testable.

Feature / area	What must the system do?	How will it be tested?	Who signs off?
User registration			
Core workflow			
Notifications			
Reporting / exports			
Permissions / access			
Other			

Who will review and approve milestones?	
Training required for staff or administrators	
Documentation / handover required	
Expected support after launch	

12. What to request in each developer proposal

Send this same request to every provider so you can compare responses fairly.

- Their understanding of the business problem and desired outcomes.
- Recommended solution and why it suits the project.
- Included scope, assumptions and explicit exclusions.
- MVP recommendation and proposed phases.
- Timeline, milestones and dependencies.
- Discovery, design, development and testing costs shown separately.
- Whether user acceptance testing (UAT) is included.

- Hosting, licences, third-party services and ongoing costs.
- Project team, roles and main point of contact.
- Source code, intellectual property, documentation and account ownership.
- Warranty, maintenance, support options and response expectations.
- Payment schedule and change-request process.

Important: Ask every developer to state what is excluded. Confirm in writing who will own the source code, project accounts, documentation and intellectual property before signing.

13. Software development quote comparison

Use one copy of this scorecard for each provider, or compare providers side by side. First check whether each proposal covers the same scope; then consider price and fit.

Provider / company name	
Proposal date / validity	
Main contact	

Comparison area	Provider A	Provider B	Provider C
Understands our business problem			
Includes all must-have requirements			
Scope and exclusions are clear			
Recommended approach is suitable			
Timeline and milestones are realistic			
Testing and UAT are included			
Security and privacy needs addressed			
Source code / IP ownership is clear			
Post-launch support is defined			
Third-party and running costs included			
References / relevant experience checked			

Cost comparison

Cost item	Provider A (AUD)	Provider B (AUD)	Provider C (AUD)
Discovery / planning			

Cost item	Provider A (AUD)	Provider B (AUD)	Provider C (AUD)
Design			
Development			
Testing / UAT			
Hosting / licences (annual)			
Support / maintenance (annual)			
Other / third-party costs			
Estimated total			

Questions / gaps to clarify	
Preferred provider and reasons	

14. Before you send the brief

A final check helps avoid missing information and makes it easier for providers to respond consistently.

- The business problem is explained in plain language.
- The most important users and workflows are included.
- Features are separated into must-have, should-have and later.
- Known integrations and data migration needs are named.
- Data, privacy, security and hosting considerations are noted.
- Budget and timeline expectations are realistic or marked as unknown.
- Acceptance criteria describe what a successful release must do.
- The same version of the brief will go to each provider.
- You have asked providers to list assumptions, exclusions and ongoing costs.

Notes and questions for your development partner

Notes	
Questions to resolve during discovery	

Not sure about every answer? That is normal. Mark unknown items clearly. A discovery phase can help define workflows, test technical assumptions, identify risks and turn an early brief into a more reliable scope and estimate.

Need help shaping your software project?

Cognify Digital works with businesses to plan and build custom software solutions. Use your completed brief as the starting point for a conversation about requirements, priorities and a practical first release.

Website: cognifydigital.co

Important note

This workbook is a general planning aid, not legal, financial, privacy or technical advice. Requirements vary by business, industry and project. Have relevant specialists review legal, security, privacy and regulatory obligations before committing to a build.

© Cognify Digital | Software Project Brief Template