BUSINESS DECISION

Software project brief

Decisions before you build. Complete with the person who owns the workflow.

> Type short answers or print; continue longer notes in your project document. Use "unknown" and assign an owner when evidence is missing. Compare configuration, an existing product and custom development.

01 Define the problem and evidence

Project / decision owner

Who can prioritize scope?

Users / release reviewer

Who does the work and accepts it?

Problem in one sentence

Who struggles to do what, and where in the workflow?

Current workflow and evidence

Trigger, key steps and finish; volume, active handling time, exceptions, source and measurement period.

Desired change and measure

Name an observable outcome. Separate turnaround, released staff capacity and actual cash savings.

02 Choose the smallest useful release

First-release must-haves

Up to three workflows and their users.

Explicit exclusions

What can wait until a later release?

Alternatives to a build

Tools or configuration evaluated; remaining gap and evidence for it.

SYSTEMS AND ACCEPTANCE

Define the boundaries

Name the real systems and owners. Exclude passwords, keys and personal customer records.

03 Map the connections and data responsibilities

Connection A

System + owner; authoritative record; data direction / trigger; interface / limits; failure handling.

Connection B

System + owner; authoritative record; data direction / trigger; interface / limits; failure handling.

Data responsibilities

Who approves access, retention, migration and deletion?

Roles and permissions

Who can view, create, approve, edit or export?

04 Describe what accepted means

> Illustrative example - not a client result: retrying an approved request with a reference ID creates one matching destination record and logs the attempt. If unavailable, keep the request visibly pending for its owner. Agree on retry rules and permitted actions.

For each case: starting condition + action + expected result + reviewer / evidence.

Normal workflow

What should happen?

Failure or exception

How is recovery made visible and owned?

Permission boundary

Which action is allowed or refused?

DELIVERY AND OWNERSHIP

Prepare a comparable scope

Use the same first-release assumptions when reviewing proposals and ongoing responsibilities.

05 Agree on launch and ownership

Review and release

Who tests, approves launch and decides to pause or roll back?

Operating responsibilities

Monitoring, triage, backups, updates and support; response arrangements.

Deliverables and access

Agree on code, docs, accounts, licenses and ownership terms in the contract.

Training and adoption

Who trains users and observes use, errors and unresolved work?

06 Compare costs and resolve unknowns

Initial work	Discovery, build, integration, migration, testing, training and internal time.
Recurring work	Hosting, licenses, usage, monitoring, maintenance, support and human review.
Uncertainty	Interfaces, data quality, approvals, assumptions and work to resolve them.
Change process	Who approves changes, updates the estimate and revises deliverables?

Budget constraint / decision date

Record actual constraints and the reason for any deadline.

Open question / owner / next step

Resolve high-impact unknowns before relying on a proposal.

Bring the brief, a sanitized workflow example, the systems inventory and known exceptions to a scoping conversation.

ROI worksheet AI readiness checklist Custom software Discuss a project

Original planning template for project scoping. Not an estimate or contract.