TEMPLATE · FREE RESOURCE

Decision Memo and Prototype Brief

Two short templates that replace a long PRD. A one to two page decision memo that gets a clear yes or no, and a prototype brief that scopes 3 to 5 days of work around the questions that actually need answers.

By **Ratomir Jovanovic** · ratomir.com/resources/decision-memo-and-prototype-brief

---

These templates are for product leaders, founders and PMs who are tired of writing twenty-page requirement documents that nobody reads past the first section, and that are out of date by the time engineering starts. They are especially useful now that a working prototype can be built in days, which changes what a document needs to do. The document no longer has to describe the product in full. It has to get the right decision made and point the prototype at the right questions.

The outcome is two short documents. The **decision memo** fits on one or two pages and ends with a decision someone with authority can say yes or no to. The **prototype brief** scopes three to five days of building around specific questions, says what the prototype is not allowed to prove, and sets the rules for what happens when it is handed to engineering. Both are in the Word document that comes with this page.

How to use this

1. Write the decision memo first, before anyone builds anything. If you cannot fill in "decision requested" in one sentence, you are not ready for a prototype yet.
2. Share the memo with the decision owner and two or three people who will be affected, and give them 48 hours to comment in writing.
3. Hold a short meeting only if the comments disagree on something material. Record the decision at the top of the memo.
4. If the decision is to explore, write the prototype brief. Copy the open questions from the memo into it.
5. Build for three to five days, test with users using the plan in the brief, and write the results directly under each question.
6. Use the handoff rules in the brief before any prototype code or design reaches engineering.
7. Keep both documents. They are the record of why you built what you built, which matters more a year later than the original PRD would have.

The PDF guide contains both templates with field-by-field notes. The Word document contains the blank templates ready to copy.

Part 1: The decision memo

A decision memo exists to get one decision made by one person or group. Everything in it should either help that decision or be cut. The six fields below are the whole template.

Problem

Describe the problem in the customer's terms, with the evidence you have and how strong it is. Say what happens today and what it costs the customer or the business.

Weak: "Users find onboarding confusing and we need to improve the experience."

Strong: "New business customers stall at bank account verification. In the last quarter's support tickets, verification was the most common onboarding topic, and in five of eight recent sales calls the buyer mentioned it unprompted. Customers who stall here usually wait for a sales rep to unblock them, which adds days to activation."

The strong version names a step, cites its evidence and admits the evidence is a mix of tickets and calls, not a controlled study.

Who has it

Name the segment precisely and say who does not have the problem. This is where most PRDs go vague, and it is the field that most often changes the decision.

Weak: "Our customers."

Strong: "Finance admins at companies with 20 to 200 employees who sign up without a sales rep. Enterprise accounts are onboarded by our team and do not see this step."

Decision requested

One sentence, phrased so that yes and no are both possible answers, with the name of the person deciding and the date you need it by. If you are asking for several things, list them separately so each can be approved or rejected.

Weak: "Alignment on the onboarding direction."

Strong: "Approve one designer and one engineer for five days to prototype instant verification, with a go or no-go on a full build at the end of the month. Decision owner: VP Product."

Constraints

List the limits that are not up for debate: budget, deadlines, regulation, contracts, technology you must or must not use, and team capacity. Constraints are not risks. A risk might happen; a constraint is already true.

Weak: "Must be secure and scalable."

Strong: "Must use our existing verification provider under the current contract. No change to the KYC data we store. Must ship before the annual price change in Q3 or wait until Q1."

What we stop doing

Every yes is a no to something else. Name it. If the answer is "nothing", the memo is asking people to absorb extra work silently, which is how roadmaps slip without anyone deciding they should.

Weak: "We will reprioritise as needed."

Strong: "We pause the reporting export redesign for six weeks. The two customers waiting on it will be told directly by their account managers."

How we will know we were wrong

Write the signal that would make you reverse or stop, and when you will check it. This is the field people skip, and it is the one that makes the memo honest. Pick a measure you can actually observe, a threshold and a date.

Weak: "We will monitor engagement."

Strong: "If, eight weeks after release, the share of self-serve signups that complete verification within a day has not improved over the prior eight weeks, we stop further work and revisit the problem statement."

Part 2: The prototype brief

A prototype brief turns open questions into a short, bounded piece of building. It protects the team from two failures: prototypes that answer nothing because they tried to do everything, and prototypes that get treated as finished products because they looked convincing in a demo.

Questions the prototype must answer

Write two to four questions, each answerable by watching people use the prototype. Rank them. If time runs out, the first question must be answered.

Weak: "Will users like the new flow?"

Strong: "Can a first-time finance admin complete verification without help in under five minutes? Do they trust instant verification enough to proceed, or do they look for a manual option?"

What it must not pretend to prove

List the things the prototype cannot tell you, so nobody claims it did. Prototypes routinely get credited with proving demand, technical feasibility at scale, or pricing acceptance, none of which a few days of building and a handful of sessions can show.

Weak: (left empty)

Strong: "This prototype uses mocked bank responses. It does not show that the provider's API meets our latency or success rates, and five sessions do not tell us what share of all customers will adopt it."

Scope for 3 to 5 days

State what is in, what is faked, and what is out. Fake anything that does not affect the questions. Name the people and the days.

In scope	Faked	Out of scope
The verification screens and the decision point between instant and manual	Bank API responses, account data	Error recovery, admin settings, mobile layout

Test plan with users

Say who you will test with, how many, how you will recruit them, what tasks you will give them and what you will record. Five to eight sessions with the right people usually show the main usability problems. Write the task script before building, so the prototype is built for the test and not the other way around.

Weak: "Show it to some customers."

Strong: "Six finance admins from recent self-serve signups, recruited through support. Each gets one task: connect a business account and reach the dashboard. We record time to complete, points of hesitation, and what they say when they reach the trust decision."

Handoff rules to engineering

Define what the prototype is allowed to become. The default rule should be that prototype code is reference material, not production code, unless engineering reviews it and decides otherwise. State which artifacts are handed over (recordings, findings, screens, a list of decisions made) and who owns the next step.

Strong: "Engineering receives the findings, the session recordings and the screens. Prototype code is not merged. The tech lead decides within one week whether to estimate a full build or to return open questions to product."

Worked example

This is a hypothetical example created to show how the two templates fit together. It does not describe a real company.

A B2B payments startup sees self-serve customers stalling at bank account verification. The PM writes a decision memo in an afternoon. The problem section cites support tickets and sales call notes and says plainly that the evidence is qualitative. The segment is self-serve finance admins at smaller companies; enterprise accounts are excluded. The decision requested is five days of one designer and one engineer to prototype instant verification, decided by the VP Product by Friday. The constraints name the existing provider contract and the Q3 pricing change. Under "what we stop doing", the reporting export redesign is paused for six weeks. Under "how we will know we were wrong", the team commits to a specific completion measure eight weeks after release.

The VP approves with one change: the decision on a full build will also need a feasibility check from the provider, which becomes a constraint on the brief.

The prototype brief lists two questions: can admins finish without help in under five minutes, and do they trust instant verification. It states that the prototype cannot prove API performance or adoption rate. Bank responses are faked. Six sessions are booked before building starts.

In testing, five of six admins finish quickly, but four hesitate at the trust decision and look for a way to speak to someone. That finding changes the design more than any requirement in a long PRD would have: the team adds a visible explanation of what instant verification shares and why. Engineering receives the findings and recordings, does not merge the prototype code, and returns a build estimate within a week.

Common mistakes

- **Writing the memo after building.** A memo written to justify a prototype that already exists will not contain a real decision.
- **A decision requested that cannot be refused.** "Alignment" and "awareness" are not decisions.
- **Leaving "what we stop doing" empty.** It is the most important line for everyone outside the team.
- **Too many prototype questions.** More than four and the prototype becomes a product build in disguise.
- **Treating a convincing demo as evidence of demand.** Write down what the prototype cannot prove before anyone sees it.
- **Merging prototype code by default.** It was built to answer questions quickly, not to be maintained.
- **Testing with whoever is available.** Colleagues and friendly customers will tell you the flow is fine.

When not to use these templates

Do not use them for decisions that are cheap to reverse and owned by one team; just make the change and watch the result. They are also the wrong tool for work with heavy regulatory, contractual or integration requirements, where engineering and compliance need a detailed specification. In those cases, use the memo to make the decision and write the full specification afterwards, once you know it is worth writing.

Editable template: Memo and brief templates (Word), available at ratomir.com/resources/decision-memo-and-prototype-brief

© 2026 Ratomir Jovanovic. Free to use inside your team. Please link to ratomir.com/resources/decision-memo-and-prototype-brief when you share it.