PRD Framework

Delivery & Leadership | Adapted synthesis | Portfolio version 1.0

A decision-oriented product requirements structure connecting problem evidence, outcomes, users, scope, workflow, requirements, analytics, risk, rollout, and open decisions.

Problem it solves

PRDs become large specifications that document features without clarifying the problem, decisions, trade-offs, success measures, or operational readiness.

When to use it

- A cross-functional team needs shared product intent.
- The workflow, scope, dependencies, or risks are complex.
- A release requires measurable acceptance and readiness criteria.
- An AI capability needs evaluation and oversight requirements.

When not to use it

- A tiny reversible change can be explained clearly in a short story.
- The problem and outcome have not been discovered.
- The document will not be maintained or used for decisions.

Inputs

- Problem and evidence
- Product outcome
- Users and jobs
- Workflow and experience
- Technical and operational constraints
- Analytics and risk context

Process

1. Executive summary - State the problem, user, outcome, approach, and decision required.

2. Problem and evidence - Document the workflow problem and evidence that it matters.

3. Users and jobs - Clarify actors, context, needs, responsibilities, and permissions.

4. Scope - Define included, excluded, deferred, and dependent work.

5. Experience and requirements - Describe workflow, behavior, functional and non-functional requirements.

6. Measurement - Define events, success metrics, guardrails, and evaluation.

7. Risk and readiness - Cover dependencies, assumptions, rollout, support, monitoring, and open decisions.

8. Acceptance - Define testable criteria for intended behavior and release readiness.

Decision points

- Is the problem sufficiently clear?
- What is intentionally out of scope?
- Which requirement protects the core outcome?
- How will success and failure be observed?
- What must be true before release?

Outputs

- PRD
- Workflow model
- Requirements and acceptance criteria
- Analytics plan
- Risk and dependency register
- Rollout and readiness plan

Success measures

- Requirement clarity
- Decision resolution time
- Scope stability
- Defect escape
- Release readiness
- Outcome instrumentation

Portfolio application

I use PRDs to translate product opportunities into shared decisions, executable stories, acceptance criteria, evaluation needs, and release-readiness expectations.

Common pitfalls

- Documenting a solution before validating the problem.
- Mixing requirements with implementation instructions.
- Omitting exclusions and open decisions.
- Writing untestable acceptance criteria.
- Ignoring analytics and operations.

Skills demonstrated

- Requirements
- Product writing
- Workflow design
- Acceptance criteria
- Risk management
- Release readiness

Interview discussion prompts

- What makes a good PRD?
- How much detail is enough?
- How do you handle unresolved decisions?
- What changes in an AI PRD?