AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Product Requirements and Documentation

Writing clear requirements and documents that engineers can build from and teams can trace, test and release.

Chapter 05 of 26 · 20 topics · Edition 2026 · skillnavora.in

CHAPTER 05

Product Requirements and Documentation

Writing clear requirements and documents that engineers can build from and teams can trace, test and release.

Product Requirement Document (PRD)

The shared document describing the problem, users, goals, scope, requirements, metrics and launch plan.

A PRD describes the problem, target users, goals, scope, requirements, success metrics and launch plan. It is a shared reference for teams, not a contract. Keep it short and link to research and designs rather than copying them in.

Business Requirement Document (BRD)

Business-level needs, objectives, scope and constraints, usually for stakeholders and sponsors.

A BRD captures business needs, objectives, scope and constraints, usually for sponsors and finance. It answers why the business should invest. The PRD then explains what the product will do to deliver that business case.

Functional Requirements

What the system must do: behaviours and features.

Functional requirements describe what the system must do, such as "a user can reset a password by email". Write each one so it can be tested. Vague phrases such as "easy to use" are not functional requirements.

Non-Functional Requirements

How well it must do it: performance, security, reliability, scalability and compliance.

These describe how well the system must perform: response time, availability, security, scalability and compliance. Quantify them, for example "99.9% uptime per month" or "search returns in under 300 ms for 95% of queries".

Technical Requirements

Constraints and needs for architecture, integrations, data, platforms and infrastructure.

Technical requirements cover architecture, integrations, data formats, platforms and infrastructure. Engineers should review them early, because they often change cost and timeline. Include constraints such as data residency rules.

Product Specifications

Detailed descriptions of behaviour, rules, data and interfaces sufficient for build and test.

Specifications describe behaviour, rules, data and interfaces in enough detail to build and test the feature. Include edge cases, validation rules and examples. A good spec reduces questions during development.

Epics, Features and User Stories

Epics are large goals, features are deliverable capabilities, stories are small user-centred slices a team can finish in a sprint.

An epic is a large goal, a feature is a deliverable capability, and a user story is a small, user-centred slice that fits in one sprint. Example: "Faster payouts" as an epic, "Instant payout request" as a feature, and "As a seller, I can request a payout from my dashboard" as a story.

User Story Mapping

Arranging stories along the user journey to see the whole experience and slice releases.

Story mapping arranges stories along the user journey, with the most important path at the top. Slice releases horizontally so each release delivers a usable end-to-end path rather than half of many features.

Acceptance Criteria

Specific, testable conditions that must be true for a story to be accepted, often in Given/When/Then form.

Acceptance criteria are specific, testable conditions a story must meet. The Given/When/Then format is common: given a verified account, when the user requests a payout above the limit, then the system shows the limit message.

Definition of Ready

The checklist a story must meet before the team starts it, such as clear value, criteria and dependencies.

The checklist a story must meet before the team starts work: clear value, acceptance criteria, dependencies identified and design available. It prevents half-understood work entering a sprint.

Definition of Done

The shared quality bar every item must meet to count as finished: tested, reviewed, documented, deployable.

The shared quality bar every item must meet: tested, reviewed, documented, monitored and deployable. Without it, items are called done before they are safe to release.

Use Cases

Step-by-step descriptions of how an actor achieves a goal with the system, including alternatives.

A use case describes how an actor achieves a goal with the system, including alternative paths and failures. Use cases are useful for complex workflows such as loan approval, where many roles and rules interact.

Edge Cases

Rare or extreme inputs and states that can break flows, found by asking what happens when things are empty, huge, slow or invalid.

Edge cases are rare or extreme inputs and states, such as an empty cart, a very large file or a timeout during payment. Ask what happens when inputs are empty, huge, slow or invalid. Many production bugs live in these cases.

Error Scenarios

How the product behaves when things fail, with clear messages, recovery paths and logging.

Error scenarios describe how the product behaves when something fails, including the message shown, the recovery path and what is logged. Good error handling tells the user what to do next.

Product Backlog Creation

A prioritised list of everything that might be built, each item with value and clear description.

The backlog is a prioritised list of everything that might be built. Each item needs a clear description and an expected value. Keep the backlog honest by removing items that no longer matter.

Backlog Refinement

Regularly clarifying, splitting, estimating and ordering backlog items so the next sprints are ready.

Refinement regularly clarifies, splits, estimates and orders backlog items so the next sprints are ready. Refine two to three sprints ahead. Items should be small enough to finish within one sprint.

Release Documentation

Release notes, change logs, runbooks and support guides that explain what changed and how to operate it.

Release notes, change logs, runbooks and support guides explain what changed and how to operate it. Support teams need these before launch, not after the first customer complaint.

Product Requirement Traceability

Linking each requirement to its goal, design, tests and release to prove coverage and manage change.

Traceability links each requirement to its goal, design, tests and release. It shows that every requirement is covered and helps assess the impact of a change. Use a simple table or tool, and keep it current.

Technical Documentation

Architecture notes, API docs and runbooks that let others build, integrate and maintain.

Architecture notes, API references and runbooks let others build, integrate and support the product. Keep them next to the code so they are updated with it.

Product Decision Records

Short records of decisions, options considered, reasons and consequences, so context is not lost.

A decision record notes the decision, the options considered, the reason and the expected consequences. Months later, people can see why a choice was made instead of re-arguing it.

Summary of terms

Term	Meaning
Product Requirement Document (PRD)	The shared document describing the problem, users, goals, scope, requirements, metrics and launch plan.
Business Requirement Document (BRD)	Business-level needs, objectives, scope and constraints, usually for stakeholders and sponsors.
Functional Requirements	What the system must do: behaviours and features.
Non-Functional Requirements	How well it must do it: performance, security, reliability, scalability and compliance.
Technical Requirements	Constraints and needs for architecture, integrations, data, platforms and infrastructure.
Product Specifications	Detailed descriptions of behaviour, rules, data and interfaces sufficient for build and test.
Epics, Features and User Stories	Epics are large goals, features are deliverable capabilities, stories are small user-centred slices a team can finish in a sprint.
User Story Mapping	Arranging stories along the user journey to see the whole experience and slice releases.
Acceptance Criteria	Specific, testable conditions that must be true for a story to be accepted, often in Given/When/Then form.
Definition of Ready	The checklist a story must meet before the team starts it, such as clear value, criteria and dependencies.
Definition of Done	The shared quality bar every item must meet to count as finished: tested, reviewed, documented, deployable.
Use Cases	Step-by-step descriptions of how an actor achieves a goal with the system, including alternatives.
Edge Cases	Rare or extreme inputs and states that can break flows, found by asking what happens when things are empty, huge, slow or invalid.
Error Scenarios	How the product behaves when things fail, with clear messages, recovery paths and logging.
Product Backlog Creation	A prioritised list of everything that might be built, each item with value and clear description.
Backlog Refinement	Regularly clarifying, splitting, estimating and ordering backlog items so the next sprints are ready.
Release Documentation	Release notes, change logs, runbooks and support guides that explain what changed and how to operate it.

Product Requirement Traceability	Linking each requirement to its goal, design, tests and release to prove coverage and manage change.
Technical Documentation	Architecture notes, API docs and runbooks that let others build, integrate and maintain.
Product Decision Records	Short records of decisions, options considered, reasons and consequences, so context is not lost.

Practice exercise: Write a two-page PRD with user stories, Given/When/Then acceptance criteria and three edge cases for a feature you choose.

Self-check

1. Explain Product Requirement Document (PRD) in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Definition of Done in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Product Decision Records in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- PRD library : Template, framing guide and 120 PRD ideas.
- AI PRD Writer : Draft a PRD with AI.
- BRD and FRD guide : Business and functional requirement documents.
- AI PM Toolkit : Templates and helpers for PM work.
- AI Tutor : Ask questions on any topic.
- Study Library : Every lesson, class and PDF by programme.

© SkillNavora · skillnavora.in · Facts and tools change; check official sites before relying on a figure.

ORIGINAL CONTENT

Original AI Product Management content

The AI Product Management programme already on SkillNavora, included unchanged so the full course is in one place.

Quick links Study Library (all material) Learning sites for PM AI builder tools (how-to) Webhooks & Playwright MCP Webhook URLs and keys (start here) AI Skills Academy AI Skills Map Context engineering (PDF) AI skills & templates (PDF) Orchestration (PDF) PRD library (120 ideas) AI PRD Writer AI PM Toolkit AI Evals (PDF) Browser automation (PDF) AI second brain (PDF) PM handbook Top 100 PM questions AI product sense Q&A (86) PM practice test Practice Lab Tool tutorials AI Agents & MCP for PMs

PRD documents

PRD library: template, framing guide and 120 PRD ideas

Learn how top PMs frame a PRD, download the free template, and turn any of 120 ideas across 15 industries into a full PRD with AI.

Open the PRD library ✨ AI PRD Writer PRD template (Word) AI Evals for PMs (PDF)

Context window

Context window: what a model can see at once

The context window is the most tokens a model can consider in one request. It covers your prompt, any documents you paste, the conversation so far and the reply. Anything outside it is invisible to the model. For PMs: a longer window fits more material but costs more and can slow responses; put the most important facts near the instructions; use retrieval (RAG) rather than pasting whole libraries; and test answers on long inputs, where quality can drop.

Context engineering course Context windows in the AI PM 2 course

Tokens and context in the Grokbot course

AI-First Product Management

Resources for Product Managers

The PM Master Handbook, AI and product tools with their official websites, interview practice, a frameworks cheat sheet, a PRD template and free courses.

Open the PM handbook library: all 68 PDFs ↗ PM Glossary Handbook Tokens, Pricing and Tokenization Vibe Coding addendum

PM Master Handbook by SkillNavora
23 modules and 620+ pages: 12 Product Core modules, an 11-class AI Tools Lab, frameworks, case studies, projects and interview prep.

Open the interactive handbook ↗ Complete handbook PDF (620+ pages, 23 MB) ↗

Product Core modules

1Fundamentals of Product Management*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
2Product Strategy and Strategic Thinking*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
3Problem Discovery and Market Research*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
4Problem Framing and Prioritization*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
5UX and Design Collaboration*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
6Effective Communication and Stakeholder Management*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
7Product Analytics and Metrics*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
8Product Launch, Growth, and Monetization*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
9Tech 101 and System Design for PMs*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
10LLMs, RAGs, and AI Foundations*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
11Vibe Coding for Product Managers*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
12Agentic AI and AI Systems*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides

AI Tools Lab

13AI Tools Landscape and Your Starter StackPDF ↗ 14Claude for Daily PM WorkPDF ↗ 15Claude Code: Basic to AdvancedPDF ↗ 16Lovable: Basic to AdvancedPDF ↗ 17Replit and Other AI App BuildersPDF ↗ 18Figma and AI Design ToolsPDF ↗ 19Notion, PRD Tools and AI DocumentationPDF ↗ 20Automation with Zapier, Make and n8nPDF ↗ 21AI Chatbots for BusinessPDF ↗ 22Voice Agents for BusinessesPDF ↗ 23Learning Paths, Courses and CommunitiesPDF ↗

Business AI Stack (chapters 24 to 31)

24The business AI stack: tools and where to learn themPDF ↗ 25Next.js: customer-facing dashboardsPDF ↗ 26Zoho CRM: managing customers and leadsPDF ↗ 27WhatsApp Business Platform: customer messagingPDF ↗ 28Deployment with Vercel and AWSPDF ↗ 29Jira: tracking product and development workPDF ↗ 30Meta Business Suite, social media marketing and Google AdsPDF ↗ 31A one-page website that generates leadsPDF ↗

Complete AI tool guides (15)

1n8n: workflows, Gmail and an AI reply agentPDF ↗ 2Make: scenarios, routers and lead routingPDF ↗ 3Zapier: invoices from Gmail to Drive and a sheetPDF ↗ 4Webhooks: how apps talk in real timePDF ↗ 5Replit: one app for the web and a phonePDF ↗ 6Lovable: a job application trackerPDF ↗ 7Figma: components, auto layout and handoffPDF ↗ 8Claude Code: an AI coding agent in your projectPDF ↗ 9MCP: connecting AI assistants to your toolsPDF ↗ 10LLMs: how they work and how to call themPDF ↗ 11RAG: answers from your own documentsPDF ↗ 12AI agents: a job-search agent with approvalsPDF

↗ 13Agentic AI: planning, guardrails and teamsPDF ↗ 14Generative AI: uses, limits and a weekly reportPDF ↗ 15Other AI tools: choosing one per jobPDF ↗

Complete AI tool guides (one PDF, 30 pages) ↗

Libraries

★Framework Library*29 frameworks with worked examples and templates*PDF ↗ ★Case Study Library*12 product case studies*PDF ↗ ★Practical Projects and Portfolio*5 end-to-end portfolio projects*PDF ↗ ★Interview Handbook*100+ PM interview questions with model answers*PDF ↗ ★AI Tools Lab*Every AI tools class with step-by-step labs*PDF ↗

PM interview practice (free, on the original sites)

Hand-picked public question banks and guides from other PM communities. Each link opens the original page in a new tab.

PM frameworks cheat sheet

Framework	Use it to	How it works
RICE	Prioritise a backlog	Reach × Impact × Confidence ÷ Effort. Score each idea, then sanity-check the ranking.
Kano	Decide which features delight	Classify features as Must-be, Performance, Attractive, Indifferent or Reverse using a short survey.
MoSCoW	Scope a release	Must have, Should have, Could have, Won't have (this time).
Jobs-to-be-done	Understand motivation	When [situation], I want to [motivation], so I can [outcome].
AARRR (Pirate metrics)	Map the growth funnel	Acquisition, Activation, Retention, Referral, Revenue.
HEART	Measure user experience	Happiness, Engagement, Adoption, Retention, Task success, each with goals, signals and metrics.
North Star metric	Align the team	One metric that captures the value users get, supported by input metrics the team can move.
Five whys	Find the root cause	Ask 'why?' repeatedly until you reach a cause you can act on.
CIRCLES	Answer product design questions	Comprehend, Identify users, Report needs, Cut through prioritisation, List solutions, Evaluate trade-offs, Summarise.

PRD template

PRD: [Feature name] Owner: [Name] · Status: Draft · Last updated: [Date] 1. Problem Who has the problem, what it is, and the evidence (data, interviews, tickets). 2. Goal and success metrics Primary metric (with baseline and target), secondary metrics, guardrails. 3. Users and use cases Target segment, top 3 user stories: "As a [user], I want [action] so that [benefit]." 4. Solution What we'll build (link designs), what's out of scope, open questions. 5. AI behaviour (if applicable) Inputs, model, prompt/approach, what it must never do, human review, evaluation set and pass criteria, cost per request. 6. Launch plan Experiment design, rollout stages, support and comms, rollback criteria. 7. Risks and dependencies Technical, legal/privacy, operational; owners and mitigations.

Free courses