AI PRODUCT MANAGEMENT 2 · COMPLETE EDITION

AI Product Management: complete detailed course

26 chapters · 600 topics · every topic explained, with a practice exercise and a self-check for each chapter.

Contents

Chapter	Subject	Topics
01	**Product Management Foundations**	16
02	**Customer Discovery and Market Research**	22
03	**Product Strategy and Roadmapping**	20
04	**Product Design, UX and Prototyping**	17
05	**Product Requirements and Documentation**	20
06	**Agile, Scrum and Product Delivery**	23
07	**Product Analytics and Data-Driven Decision Making**	26
08	**Product Metrics and Business Analytics**	20
09	**Business, Finance and Product Commercialisation**	24
10	**Technical Foundations for Product Managers**	28
11	**Artificial Intelligence and Machine Learning Foundations**	32
12	**Generative AI and Large Language Models**	30
13	**Prompt Engineering and Context Engineering**	23
14	**Retrieval-Augmented Generation (RAG)**	29
15	**Agentic AI and Autonomous AI Systems**	33
16	**AI Product Architecture and Development**	26
17	**AI Product Evaluation, Testing and Observability**	30
18	**Responsible AI, Safety and Governance**	24
19	**AI Product UX and Human-AI Interaction**	17
20	**AI Product Strategy and Monetisation**	20
21	**AI Product Tools and Ecosystem**	32
22	**Product Leadership and Stakeholder Management**	18
23	**Product Management Specialisations**	16
24	**Industry Case Studies and Product Teardowns**	20
25	**End-to-End Practical Capstone Projects**	12
26	**Interview Preparation and Career Readiness**	22

How to use this course

Read a chapter, write the summary of each topic in your own words, then complete the practice exercise. Use the self-check before moving on, and build the capstone projects in chapter 25 to show your work.



AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Product Management Foundations

What a product manager does, how the role differs from its neighbours, and how a product moves from idea to growth. This subject sets the vocabulary used in every later subject.

Chapter 01 of 26 · 16 topics · Edition 2026 · skillnavora.in

CHAPTER 01

Product Management Foundations

What a product manager does, how the role differs from its neighbours, and how a product moves from idea to growth. This subject sets the vocabulary used in every later subject.

Introduction to Product Management

The discipline of deciding what to build, for whom and why, then guiding it from idea to a product that delivers customer and business value.

Product management connects customer problems, business goals and technical feasibility. A PM decides what to build and why, then makes sure it reaches customers. Example: a fintech app notices users abandon KYC; the PM frames the problem before any feature is discussed.

Evolution of Product Management

From brand and packaged-goods managers, through software and agile teams, to today's data-driven and AI-assisted product teams.

The role grew from brand management at consumer goods firms to software product management in the 2000s, and now includes data-driven and AI-assisted practice. Knowing this history explains why modern PMs are expected to run experiments and understand technology.

Product Manager Roles and Responsibilities

Owning the problem, vision, strategy, roadmap, priorities and outcomes, and aligning engineering, design, data, marketing and sales around them.

A PM owns the problem, the vision, the roadmap and the outcome metrics, but not the team. Influence comes from clarity: writing decisions down, sharing evidence and keeping engineering, design and sales aligned on one goal.

Product Manager vs Product Owner vs Project Manager vs Business Analyst

PM owns the why and what; the product owner manages the backlog in Scrum; the project manager owns schedule and delivery; the analyst owns detailed requirements and process.

The PM owns the product strategy and outcomes. The product owner manages the backlog inside a Scrum team. The project manager tracks schedule, budget and delivery. The business analyst details requirements and processes. Confusion between these roles causes most early friction in new teams.

Product Manager vs Product Analyst vs Product Operations

The analyst answers questions with data; product operations builds the processes, tooling and insights that let PMs work faster; the PM decides what to do.

The analyst answers questions with data and builds dashboards. Product operations builds the processes, tools and data pipelines that let teams work well. The PM turns those inputs into decisions about what to build next.

Types of Product Managers

Core, technical, growth, platform, data, AI, B2B, B2C and enterprise PMs, each weighting skills differently.

Core PMs ship features in one area, technical PMs own platforms and APIs, growth PMs optimise funnels, and AI PMs manage uncertainty from models. Pick the type that matches your background, then build the skills it needs.

Product Lifecycle Management

Managing a product through introduction, growth, maturity and decline, adjusting strategy, investment and metrics at each stage.

Products move through introduction, growth, maturity and decline. Each stage needs different investment: acquisition early, retention and cost control in maturity, and a sunset plan at the end. Example: a legacy reporting tool may be kept alive only for a few enterprise clients.

Product Discovery vs Product Delivery

Discovery decides what is worth building through research and experiments; delivery builds and ships it. Strong teams run both continuously.

Discovery answers whether a problem is worth solving, using interviews, prototypes and experiments. Delivery builds and ships the validated solution reliably. Teams that skip discovery build well-engineered features nobody uses.

Product Thinking and Customer-Centricity

Starting from the customer's problem and the outcome they want, then working back to features, instead of starting from solutions.

Product thinking starts from the user's job and pain, not from a solution a stakeholder wants. Ask what changes for the customer if you ship the feature. A useful test is whether you can state the problem without mentioning your product.

Product Vision, Mission and Objectives

Mission says why the company exists, vision describes the future state the product enables, and objectives set the measurable steps toward it.

The mission explains why the company exists. The vision describes the future the product enables, for example a world where every small shop can accept digital payments. Objectives break that vision into measurable steps for the next quarter or year.

Product Strategy Fundamentals

A coherent set of choices on where to play and how to win, backed by evidence about customers, market and capabilities.

Strategy is a set of choices about which customers to serve, which problems to solve and how to win against alternatives. A good strategy states what you will not do. Without that, roadmaps become lists of requests.

Product-Market Fit

The point at which a product satisfies strong market demand, seen in retention, organic growth and customers who would be very disappointed to lose it.

Product-market fit means a clear group of customers depends on the product and would be disappointed if it disappeared. Signs include retention that flattens over time, organic referrals and growing willingness to pay. It is a milestone you test for, not a feeling.

Product-Led Growth

A model where the product itself drives acquisition, conversion and expansion, through free tiers, trials, self-serve onboarding and viral loops.

In product-led growth the product itself drives acquisition and expansion through free tiers, self-serve onboarding and sharing features. Slack and Dropbox are common examples. It works best when value is visible within minutes of signing up.

B2B, B2C, B2B2C and SaaS Product Models

Who buys and who uses differ by model: businesses, consumers, businesses serving consumers, and subscription software delivered online.

B2B sells to organisations with long sales cycles and many stakeholders. B2C sells to consumers at scale with simpler decisions. B2B2C sells through a business to consumers, such as a bank offering a loan app through an employer. SaaS is subscription software delivered online, which shapes pricing and retention work.

Product Management Operating Models

How product work is organised: feature teams, stream-aligned teams, platform teams, centralised or embedded PMs, and how they plan together.

Teams organise around features, streams of value or platforms. Some companies embed PMs in each team, others centralise them. The right model depends on how much coordination the product needs and how fast the company changes direction.

Product Management Career Paths

Associate PM to PM, senior, group, director and CPO, plus branches into specialist, founder and AI product roles.

A common path runs associate PM, PM, senior PM, group PM, director and chief product officer. Specialist paths in AI, growth or platforms are also valid. Build depth in one area and breadth in the others.

Summary of terms

Term	Meaning
Introduction to Product Management	The discipline of deciding what to build, for whom and why, then guiding it from idea to a product that delivers customer and business value.
Evolution of Product Management	From brand and packaged-goods managers, through software and agile teams, to today's data-driven and AI-assisted product teams.

Product Manager Roles and Responsibilities	Owning the problem, vision, strategy, roadmap, priorities and outcomes, and aligning engineering, design, data, marketing and sales around them.
Product Manager vs Product Owner vs Project Manager vs Business Analyst	PM owns the why and what; the product owner manages the backlog in Scrum; the project manager owns schedule and delivery; the analyst owns detailed requirements and process.
Product Manager vs Product Analyst vs Product Operations	The analyst answers questions with data; product operations builds the processes, tooling and insights that let PMs work faster; the PM decides what to do.
Types of Product Managers	Core, technical, growth, platform, data, AI, B2B, B2C and enterprise PMs, each weighting skills differently.
Product Lifecycle Management	Managing a product through introduction, growth, maturity and decline, adjusting strategy, investment and metrics at each stage.
Product Discovery vs Product Delivery	Discovery decides what is worth building through research and experiments; delivery builds and ships it. Strong teams run both continuously.
Product Thinking and Customer-Centricity	Starting from the customer's problem and the outcome they want, then working back to features, instead of starting from solutions.
Product Vision, Mission and Objectives	Mission says why the company exists, vision describes the future state the product enables, and objectives set the measurable steps toward it.
Product Strategy Fundamentals	A coherent set of choices on where to play and how to win, backed by evidence about customers, market and capabilities.
Product-Market Fit	The point at which a product satisfies strong market demand, seen in retention, organic growth and customers who would be very disappointed to lose it.
Product-Led Growth	A model where the product itself drives acquisition, conversion and expansion, through free tiers, trials, self-serve onboarding and viral loops.
B2B, B2C, B2B2C and SaaS Product Models	Who buys and who uses differ by model: businesses, consumers, businesses serving consumers, and subscription software delivered online.
Product Management Operating Models	How product work is organised: feature teams, stream-aligned teams, platform teams, centralised or embedded PMs, and how they plan together.
Product Management Career Paths	Associate PM to PM, senior, group, director and CPO, plus branches into specialist, founder and AI product roles.

Practice exercise: Write a one-page job description for the PM of a product you use, then list three decisions that PM owns and three they only influence.

Self-check

1. Explain Introduction to Product Management in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Product Thinking and Customer-Centricity in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Product Management Career Paths in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- AI Product Management (original programme) : The programme you already have: PM Master Handbook, PRD and BRD templates, interview prep.
- Interactive PM Handbook : 23 modules: product core, AI tools lab, frameworks, case studies and interview prep.
- Product Operations : Free PDF, basic to advanced.
- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Customer Discovery and Market Research

Methods to find real problems worth solving, understand customers deeply and size and test the opportunity before heavy investment.

Chapter 02 of 26 · 22 topics · Edition 2026 · skillnavora.in

CHAPTER 02

Customer Discovery and Market Research

Methods to find real problems worth solving, understand customers deeply and size and test the opportunity before heavy investment.

Design Thinking

A human-centred loop of empathise, define, ideate, prototype and test used to explore problems and solutions.

Design thinking is a loop of empathise, define, ideate, prototype and test. It helps teams explore a problem widely before choosing a solution. Example: a team interviews delivery riders before designing a new earnings screen, and tests two layouts with five riders.

Problem Identification and Validation

Finding a problem, then proving it is frequent, painful and valuable enough that people will change behaviour or pay.

A problem is worth solving when it is frequent, painful and valuable enough that people change behaviour or pay. Validate by checking how often users face it and what they already do about it. Workarounds are a strong signal of real need.

Customer Research Methodologies

Interviews, surveys, observation, diary studies, usability tests and data analysis, chosen to match the question.

Interviews explain motivations, surveys measure patterns at scale, observation reveals real behaviour, and analytics shows what users actually do. Choose the method that answers the decision you must make, and combine two methods where possible.

Qualitative and Quantitative Research

Qualitative research explains why people behave as they do; quantitative research measures how many and how much.

Qualitative research explains why people act as they do, through small interviews and open questions. Quantitative research measures how many people act a certain way. Use qualitative work to find the problem and quantitative work to size it.

Customer Interviews

Open, non-leading conversations about past behaviour and real problems; ask for stories, not opinions about your idea.

A good interview asks about past behaviour and specific stories, not hypothetical future use. Avoid pitching your idea. Ask "tell me about the last time you did this" and let the customer describe the context and workarounds.

Surveys and Feedback Collection

Structured questions at scale; keep them short, unbiased and tied to a decision, and pair them with interviews.

Surveys work when questions are short, unbiased and tied to a decision. Asking "would you use this?" gives misleading results because people are polite. Ask about current behaviour, then follow up with interviews to understand the answers.

User Personas

Evidence-based archetypes of key user groups with goals, context and pain points, used to keep decisions grounded.

Personas are evidence-based descriptions of key user groups, including goals, context and frustrations. Build them from interviews and data, not from imagination. Use them to check each decision: which persona does this help, and which does it hurt?

Jobs to Be Done (JTBD)

People hire products to make progress in a situation; the job, not demographics, explains what they choose.

People hire products to make progress in a specific situation, for example "help me send money home quickly on payday". Focusing on the job reveals competitors you did not expect, such as a notebook, or a relative to call.

Customer Pain Points

Specific frustrations in a workflow, ranked by severity and frequency, that products can relieve.

Pain points are specific frustrations in a workflow, such as re-entering data twice or waiting for approval. Rank them by how often they occur and how much they cost the customer. The highest-ranked pain is usually the best place to start.

Customer Journey Mapping

A map of stages, actions, thoughts and emotions from awareness to loyalty, exposing gaps and moments that matter.

A journey map lays out the stages a customer goes through, with actions, thoughts, emotions and touchpoints at each step. It shows where people drop off and where the experience breaks. Build it from real interview stories, then compare it with analytics.

User Journey Analysis

Examining how users actually move through the product, using behaviour data to find drop-offs and detours.

This examines how users move through the product using event data, such as which paths lead to success and which lead to dead ends. Look for repeated detours, because they often reveal confusing labels or missing features.

Empathy Mapping

A one-page view of what a user says, thinks, does and feels to align the team on their perspective.

An empathy map captures what a user says, thinks, does and feels. It aligns the team on a shared picture of the user before debating solutions. Keep it short and grounded in quotes from real research.

Voice of Customer (VoC)

Systematically collecting and synthesising customer feedback from support, reviews, sales calls and surveys.

VoC combines feedback from support tickets, reviews, sales calls and surveys into themes. Tag each comment, count themes over time and share the top themes with the team every month so customer voice shapes priorities.

Market Research

Gathering and analysing information about market size, trends, customers and competitors to guide decisions.

Market research covers the size of demand, trends, customer segments and competitor moves. Combine secondary sources such as industry reports with primary interviews. Be clear about which numbers are estimates and how you derived them.

Market Segmentation

Dividing a market into groups with shared needs or behaviours so you can target and position precisely.

Segmentation divides a market into groups with different needs, behaviours or willingness to pay. A useful segment is large enough to matter, reachable through a channel and different enough to need its own product or message.

TAM, SAM and SOM

Total addressable market, the serviceable part you can reach with your model, and the share you can realistically obtain.

TAM is the total demand for the category, SAM is the part your model and geography can serve, and SOM is the share you can realistically win in the near term. Build each from clear assumptions, such as users times price times frequency, and keep SOM conservative.

Competitor Analysis

Comparing competitors' offers, pricing, positioning and strengths to find gaps and differentiation.

Compare competitors on target customer, price, features, distribution and weaknesses reported by users. Look for gaps nobody serves well. Competitor analysis should change at least one decision, otherwise it is only a list.

SWOT Analysis

Strengths, weaknesses, opportunities and threats, a quick structured view of internal and external factors.

SWOT lists strengths, weaknesses, opportunities and threats. It is quick, but it often becomes a list of generic points. Make each item specific and link it to an action, so the analysis leads to a decision.

Porter's Five Forces

Industry attractiveness from rivalry, new entrants, substitutes, buyer power and supplier power.

Porter's model assesses industry attractiveness through rivalry, threat of new entrants, substitutes, buyer power and supplier power. It explains why some markets are profitable and others are crowded. Use it to judge whether a market is worth entering.

Value Proposition Canvas

Matches customer jobs, pains and gains to the products, pain relievers and gain creators you offer.

The canvas matches customer jobs, pains and gains against your products, pain relievers and gain creators. It shows whether your offer addresses what customers actually care about. Gaps in the canvas point to features you may not need.

Opportunity Assessment

A short brief that asks the key questions: problem, audience, size, alternatives, why now, success measures and risks.

An opportunity assessment is a short brief covering the problem, audience, market size, existing alternatives, why now, success measures and risks. It lets leaders compare opportunities on the same basis before committing people.

Hypothesis Development and Validation

Turn assumptions into testable statements with a metric and threshold, then run the cheapest test that could prove them wrong.

Write each assumption as a testable statement with a metric and a threshold, for example "at least 30% of trial users will invite a teammate within 7 days". Then run the cheapest test that could prove it wrong, before building anything large.

Summary of terms

Term	Meaning
Design Thinking	A human-centred loop of empathise, define, ideate, prototype and test used to explore problems and solutions.
Problem Identification and Validation	Finding a problem, then proving it is frequent, painful and valuable enough that people will change behaviour or pay.
Customer Research Methodologies	Interviews, surveys, observation, diary studies, usability tests and data analysis, chosen to match the question.

Qualitative and Quantitative Research	Qualitative research explains why people behave as they do; quantitative research measures how many and how much.
Customer Interviews	Open, non-leading conversations about past behaviour and real problems; ask for stories, not opinions about your idea.
Surveys and Feedback Collection	Structured questions at scale; keep them short, unbiased and tied to a decision, and pair them with interviews.
User Personas	Evidence-based archetypes of key user groups with goals, context and pain points, used to keep decisions grounded.
Jobs to Be Done (JTBD)	People hire products to make progress in a situation; the job, not demographics, explains what they choose.
Customer Pain Points	Specific frustrations in a workflow, ranked by severity and frequency, that products can relieve.
Customer Journey Mapping	A map of stages, actions, thoughts and emotions from awareness to loyalty, exposing gaps and moments that matter.
User Journey Analysis	Examining how users actually move through the product, using behaviour data to find drop-offs and detours.
Empathy Mapping	A one-page view of what a user says, thinks, does and feels to align the team on their perspective.
Voice of Customer (VoC)	Systematically collecting and synthesising customer feedback from support, reviews, sales calls and surveys.
Market Research	Gathering and analysing information about market size, trends, customers and competitors to guide decisions.
Market Segmentation	Dividing a market into groups with shared needs or behaviours so you can target and position precisely.
TAM, SAM and SOM	Total addressable market, the serviceable part you can reach with your model, and the share you can realistically obtain.
Competitor Analysis	Comparing competitors' offers, pricing, positioning and strengths to find gaps and differentiation.
SWOT Analysis	Strengths, weaknesses, opportunities and threats, a quick structured view of internal and external factors.
Porter's Five Forces	Industry attractiveness from rivalry, new entrants, substitutes, buyer power and supplier power.
Value Proposition Canvas	Matches customer jobs, pains and gains to the products, pain relievers and gain creators you offer.
Opportunity Assessment	A short brief that asks the key questions: problem, audience, size, alternatives, why now, success measures and risks.
Hypothesis Development and Validation	Turn assumptions into testable statements with a metric and threshold, then run the cheapest test that could prove them wrong.

Practice exercise: Run five 20-minute interviews about one problem. Write the top three pains with a quote each and a hypothesis you could test this week.

Self-check

1. Explain Design Thinking in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Empathy Mapping in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Hypothesis Development and Validation in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Product Strategy and Roadmapping

Turning vision and research into focused strategy, an outcome-based roadmap, and defensible prioritisation and investment decisions.

Chapter 03 of 26 · 20 topics · Edition 2026 · skillnavora.in



CHAPTER 03

Product Strategy and Roadmapping

Turning vision and research into focused strategy, an outcome-based roadmap, and defensible prioritisation and investment decisions.

Product Vision and Strategic Alignment

Connecting the product vision to company strategy so teams share direction and priorities.

Strategic alignment connects the product vision to company goals so teams share priorities. Without alignment, two teams can build excellent features that pull the company in opposite directions. Review the link between vision and goals every quarter.

Strategic Product Planning

Setting goals, bets and sequencing over quarters and years, balancing growth, quality and capacity.

Planning sets bets and sequencing across quarters and years, balancing growth, quality and team capacity. A plan should name the outcome each bet should move and what evidence would make you stop it.

Product Positioning

The place your product holds in the customer's mind relative to alternatives: for whom, what category and why better.

Positioning defines for whom the product is, which category it belongs to, and why it is better than the alternatives. A clear positioning statement lets sales, marketing and product describe the same thing.

Product Differentiation

The distinct, valued advantage that competitors cannot easily copy, such as data, workflow depth, brand or distribution.

Differentiation is a valued advantage competitors cannot easily copy, such as proprietary data, deep workflow integration or a trusted brand. A feature anyone can copy is parity, not differentiation.

Product Strategy Frameworks

Tools such as Playing to Win, Good Strategy (diagnosis, guiding policy, actions), Wardley maps and the three horizons model.

Frameworks such as Playing to Win ask where to play and how to win. Good Strategy separates diagnosis, guiding policy and coherent actions. Wardley maps show how components evolve over time. Use one framework consistently rather than many at once.

Product Roadmap Development

A communication tool showing problems and outcomes planned over time, flexible on detail and honest about uncertainty.

A roadmap communicates intent over time. Show problems and outcomes rather than dated promises for features. Mark what is committed, planned and exploratory, so stakeholders understand the uncertainty.

Outcome-Based Roadmapping

Organising the roadmap around customer and business outcomes instead of a feature list with dates.

An outcome roadmap groups work under the customer or business outcome it should change, such as reducing onboarding time by 30%. Teams then compete on ideas for that outcome rather than delivering a fixed list of features.

OKRs and KPIs

OKRs set ambitious objectives with measurable key results for a period; KPIs track ongoing health of the business or product.

OKRs set ambitious objectives with measurable key results for a period. KPIs track ongoing health, such as uptime or monthly active users. Use OKRs for change you are pursuing and KPIs for the health you must maintain.

North Star Metric

The single metric that best captures the value customers get, and that leads revenue.

The north star is the single measure that best captures the value customers receive, such as weekly songs played per listener. It leads revenue and guides the team. Support it with input metrics the team can directly influence.

Product Portfolio Management

Allocating investment across multiple products or bets by stage, risk and return.

Portfolio management allocates investment across products or bets by stage, risk and expected return. A healthy portfolio mixes proven products that fund growth with a few experiments that could become the next major line.

Feature Prioritisation

Ordering work by value, effort, risk and strategic fit with explicit criteria rather than opinion.

Prioritisation orders work by value, effort, risk and strategic fit using explicit criteria. Write the criteria down so the ranking can be challenged on the inputs rather than on personal opinion.

RICE Framework

Score = (Reach x Impact x Confidence) / Effort, a simple way to compare initiatives.

RICE scores an initiative as (Reach x Impact x Confidence) divided by Effort. Example: reaching 5,000 users with impact 2, confidence 80% and effort 2 person-months gives (5,000 × 2 × 0.8) / 2 = 4,000. Use the score to compare items, not to decide alone.

MoSCoW Framework

Classify requirements as Must, Should, Could and Won't have in this release to fix scope.

MoSCoW sorts requirements into Must, Should, Could and Won't-have for this release. It fixes scope early. Teams often label too many items as Must, so ask what fails if the item is removed.

Kano Model

Separates basic expectations, performance features and delighters to see how features affect satisfaction.

The Kano model separates basic expectations, performance features that scale satisfaction, and delighters that create surprise. Basic expectations must work or users leave. Delighters should be added after the basics are reliable.

ICE Scoring

Impact, Confidence and Ease scored 1-10 and multiplied or averaged for fast ranking.

ICE scores Impact, Confidence and Ease from 1 to 10 and combines them. It is quick for early ideas. Its weakness is subjectivity, so calibrate scores by comparing two or three items the whole team already understands.

Impact vs Effort Matrix

A 2×2 that shows quick wins, big bets, fill-ins and time sinks.

A two-by-two grid places items as quick wins (high impact, low effort), big bets, fill-ins and time sinks. It makes trade-offs visible in a single meeting. Plot items as a group discussion, not as one person's view.

Opportunity Scoring

Rates how important a customer need is against how satisfied they are; high importance with low satisfaction signals opportunity.

Opportunity scoring rates how important each customer need is and how satisfied customers are with current solutions. Needs that are important but poorly served point to opportunities. It works best with a survey of many customers.

Stakeholder Prioritisation

Aligning priorities with executives, sales, support and engineering using shared criteria and transparent trade-offs.

Stakeholders bring competing requests from sales, support and executives. Agree the criteria first, then score requests together. Show how each request ranks, so people see the trade-off rather than a decision made in private.

Strategic Trade-Offs

Choosing what not to do: speed vs quality, breadth vs depth, build now vs later, and documenting the reasoning.

Strategy requires saying no. Trade-offs include speed against quality, breadth against depth and building now against later. Record the reasoning and the conditions under which the decision would change.

Product Investment Decisions

Deciding where to put people and money using expected value, strategic fit, evidence and option value.

Investment decisions weigh expected value, strategic fit, evidence and option value, which is the value of keeping choices open. Stage investment so a bet earns more funding only after it hits defined evidence milestones.

Summary of terms

Term	Meaning
Product Vision and Strategic Alignment	Connecting the product vision to company strategy so teams share direction and priorities.
Strategic Product Planning	Setting goals, bets and sequencing over quarters and years, balancing growth, quality and capacity.
Product Positioning	The place your product holds in the customer's mind relative to alternatives: for whom, what category and why better.
Product Differentiation	The distinct, valued advantage that competitors cannot easily copy, such as data, workflow depth, brand or distribution.
Product Strategy Frameworks	Tools such as Playing to Win, Good Strategy (diagnosis, guiding policy, actions), Wardley maps and the three horizons model.
Product Roadmap Development	A communication tool showing problems and outcomes planned over time, flexible on detail and honest about uncertainty.
Outcome-Based Roadmapping	Organising the roadmap around customer and business outcomes instead of a feature list with dates.
OKRs and KPIs	OKRs set ambitious objectives with measurable key results for a period; KPIs track ongoing health of the business or product.
North Star Metric	The single metric that best captures the value customers get, and that leads revenue.

Product Portfolio Management	Allocating investment across multiple products or bets by stage, risk and return.
Feature Prioritisation	Ordering work by value, effort, risk and strategic fit with explicit criteria rather than opinion.
RICE Framework	Score = (Reach x Impact x Confidence) / Effort, a simple way to compare initiatives.
MoSCoW Framework	Classify requirements as Must, Should, Could and Won't have in this release to fix scope.
Kano Model	Separates basic expectations, performance features and delighters to see how features affect satisfaction.
ICE Scoring	Impact, Confidence and Ease scored 1-10 and multiplied or averaged for fast ranking.
Impact vs Effort Matrix	A 2×2 that shows quick wins, big bets, fill-ins and time sinks.
Opportunity Scoring	Rates how important a customer need is against how satisfied they are; high importance with low satisfaction signals opportunity.
Stakeholder Prioritisation	Aligning priorities with executives, sales, support and engineering using shared criteria and transparent trade-offs.
Strategic Trade-Offs	Choosing what not to do: speed vs quality, breadth vs depth, build now vs later, and documenting the reasoning.
Product Investment Decisions	Deciding where to put people and money using expected value, strategic fit, evidence and option value.

Practice exercise: Score ten backlog ideas with RICE and again with ICE. Explain every place the rankings differ.

Self-check

1. Explain Product Vision and Strategic Alignment in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Feature Prioritisation in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Product Investment Decisions in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- 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.



AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Product Design, UX and Prototyping

The design knowledge a PM needs to partner with designers, test ideas cheaply and shape usable, accessible products, including AI-first experiences.

Chapter 04 of 26 · 17 topics · Edition 2026 · skillnavora.in

CHAPTER 04

Product Design, UX and Prototyping

The design knowledge a PM needs to partner with designers, test ideas cheaply and shape usable, accessible products, including AI-first experiences.

UX Fundamentals

User experience covers the whole interaction: usefulness, usability, findability, credibility and delight.

UX covers the whole experience: usefulness, usability, findability, credibility and delight. A feature can be useful but hard to find, so the PM must consider each aspect. Start evaluation with whether the user can complete the core task.

UI Fundamentals

Visual interface design: layout, typography, colour, components and consistency.

UI is the visual layer: layout, typography, colour, components and states such as hover and error. Consistent components reduce cognitive load. A design system keeps UI consistent across screens and teams.

Human-Centred Design

Designing with deep understanding of people and testing with them throughout.

Human-centred design involves users throughout the process, from research to testing. It starts from user needs and checks solutions with real people. PMs contribute by bringing research findings into design discussions early.

Design Thinking Process

Empathise, define, ideate, prototype, test, applied iteratively to product problems.

The process of empathise, define, ideate, prototype and test is applied repeatedly to product problems. Each loop should narrow uncertainty. Stop when the riskiest assumptions are tested, not when the design looks finished.

Information Architecture

Organising and labelling content and navigation so users can find things.

Information architecture organises and labels content and navigation so users can find things. Good labels use the words users use, not internal jargon. Test labels with card sorting or tree tests before building menus.

User Flows

Diagrams of the steps a user takes to complete a task, including decisions and error paths.

A user flow diagrams the steps a user takes to complete a task, including decisions and error paths. Flows expose missing states, such as what happens when a payment fails. Review flows with engineers to catch dependencies early.

Wireframing

Low-detail layout sketches that settle structure and content before visual design.

Wireframes are low-detail layouts that settle structure and content before visual design. They keep discussion on function. Use boxes and labels, not colours and icons, so feedback focuses on the structure.

Low-Fidelity and High-Fidelity Prototypes

Low-fi (paper, boxes) tests ideas fast; high-fi looks and behaves like the real product for realistic testing.

Low-fidelity prototypes (paper or simple boxes) test ideas quickly and cheaply. High-fidelity prototypes look and behave like the final product, and are better for testing realistic tasks. Match the fidelity to the question you need answered.

Interactive Prototyping

Clickable prototypes that simulate flows so you can test with users before engineering builds.

Interactive prototypes simulate flows so users can click through a task before engineering starts. Test the riskiest task first, for example a checkout step, and watch where users hesitate.

Figma Fundamentals

Frames, components, auto layout, variables and prototyping in the standard collaborative design tool.

Figma is the common design tool. Learn frames, components, auto layout, variables and prototyping. Components keep designs consistent, and variables let you change colours or spacing in one place.

Usability Testing

Watching representative users attempt tasks, to find where they struggle, with five users per round revealing most issues.

Usability testing watches representative users attempt tasks. Five users per round typically reveal most major usability issues. Record the task, the wrong turns and the words users use, then fix the most severe issues first.

Design Validation

Confirming a design solves the problem with evidence from tests, metrics and feedback before scaling it.

Design validation confirms that a design solves the problem using test results, metrics and feedback. Before scaling a design, define what success looks like, such as task completion above 80% without help.

Accessibility and Inclusive Design

Designing for people with different abilities and contexts, guided by WCAG: contrast, keyboard use, screen readers and clear language.

Inclusive design serves people with different abilities and contexts. WCAG guidelines define contrast, keyboard access, screen reader support and clear language. Accessibility improves the product for everyone, not only for disabled users.

Mobile-First Product Design

Designing for small screens and constraints first, then extending to larger screens.

Designing for the smallest screen first forces focus on essential content and actions. Then expand to larger screens. Many users in India access services mainly on low-end phones and slow networks, so design for those constraints first.

Conversational UX

Designing dialogue: turn-taking, clarifying questions, tone, error recovery and graceful hand-off.

Conversational design covers turn-taking, clarifying questions, tone, error recovery and hand-off to a human. A good assistant states its scope clearly and recovers gracefully when it misunderstands the user.

AI-First User Experience

Experiences built around AI capability from the start, with uncertainty, feedback and control designed in.

AI-first design starts from what the AI does well and designs fallbacks for what it cannot do. It makes uncertainty, sources and user control visible so people can judge when to rely on the output.

Human-AI Interaction Design

Patterns that set expectations, show confidence, let users correct the AI and keep them in control.

These patterns set expectations, show confidence levels, let users correct the AI and keep them in control. Example: an AI draft is shown with editable fields and an undo option, not auto-sent.

Summary of terms

Term	Meaning
UX Fundamentals	User experience covers the whole interaction: usefulness, usability, findability, credibility and delight.
UI Fundamentals	Visual interface design: layout, typography, colour, components and consistency.
Human-Centred Design	Designing with deep understanding of people and testing with them throughout.
Design Thinking Process	Empathise, define, ideate, prototype, test, applied iteratively to product problems.
Information Architecture	Organising and labelling content and navigation so users can find things.
User Flows	Diagrams of the steps a user takes to complete a task, including decisions and error paths.

Wireframing	Low-detail layout sketches that settle structure and content before visual design.
Low-Fidelity and High-Fidelity Prototypes	Low-fi (paper, boxes) tests ideas fast; high-fi looks and behaves like the real product for realistic testing.
Interactive Prototyping	Clickable prototypes that simulate flows so you can test with users before engineering builds.
Figma Fundamentals	Frames, components, auto layout, variables and prototyping in the standard collaborative design tool.
Usability Testing	Watching representative users attempt tasks, to find where they struggle, with five users per round revealing most issues.
Design Validation	Confirming a design solves the problem with evidence from tests, metrics and feedback before scaling it.
Accessibility and Inclusive Design	Designing for people with different abilities and contexts, guided by WCAG: contrast, keyboard use, screen readers and clear language.
Mobile-First Product Design	Designing for small screens and constraints first, then extending to larger screens.
Conversational UX	Designing dialogue: turn-taking, clarifying questions, tone, error recovery and graceful hand-off.
AI-First User Experience	Experiences built around AI capability from the start, with uncertainty, feedback and control designed in.
Human-AI Interaction Design	Patterns that set expectations, show confidence, let users correct the AI and keep them in control.

Practice exercise: Sketch a three-screen flow in Figma for one task, test it with three people, and record what confused them.

Self-check

1. Explain UX Fundamentals in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Interactive Prototyping in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Human-AI Interaction Design in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- Product Design for PMs : UX, prototyping and design collaboration.
- 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.



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.



AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Agile, Scrum and Product Delivery

How teams plan, build and ship iteratively, and the tools PMs use to coordinate delivery.

Chapter 06 of 26 · 23 topics · Edition 2026 · skillnavora.in

CHAPTER 06

Agile, Scrum and Product Delivery

How teams plan, build and ship iteratively, and the tools PMs use to coordinate delivery.

Agile Fundamentals

Iterative delivery, frequent feedback and adapting to change over following a fixed plan.

Agile delivers work in short iterations, gathers feedback quickly and adapts plans to what is learned. It replaces long fixed plans with frequent checkpoints. The PM's job is to keep the backlog ordered by outcome value.

Agile Manifesto

Four values and twelve principles favouring individuals, working software, collaboration and responding to change.

The manifesto values individuals and interactions, working software, customer collaboration and responding to change. The twelve principles add practical guidance such as frequent delivery and sustainable pace. Use the values as a test for practices, not as a rulebook.

Scrum Framework

Fixed-length sprints with defined roles, events and artefacts to deliver increments.

Scrum organises work into fixed-length sprints, with defined roles, events and artefacts. Each sprint produces an increment of working product. Teams adopt Scrum to create a predictable rhythm of planning, review and improvement.

Kanban Methodology

A continuous flow board that limits work in progress and optimises throughput.

Kanban visualises work on a board, limits work in progress and measures flow. It suits support and operations work with unpredictable arrivals. Limits prevent teams from starting too much and finishing nothing.

Scrum Roles and Responsibilities

Product Owner, Scrum Master and Developers, each accountable for a distinct part of the work.

The product owner orders the backlog and maximises value. The Scrum master removes impediments and coaches the process. Developers build the increment. Each role has a distinct accountability, and blurring them is a common failure.

Product Owner Responsibilities

Maximising product value by managing and ordering the backlog and accepting work.

The product owner manages the backlog, writes clear items, accepts work against criteria and represents customers to the team. A product owner who is absent from daily decisions cannot steer the team well.

Sprint Planning

The team selects backlog items and plans how to deliver them within the sprint goal.

The team selects backlog items and agrees a sprint goal. Planning answers what can be delivered and how. A good sprint goal describes the outcome, so the team can adapt scope without losing purpose.

Daily Stand-ups

A 15-minute sync on progress toward the sprint goal and blockers.

The daily stand-up is a 15-minute check on progress toward the sprint goal and on blockers. It is not a status report to the manager. Use it to remove blockers quickly, for example by escalating a waiting API decision the same day.

Sprint Reviews

Demonstrating the increment to stakeholders and gathering feedback.

The sprint review demonstrates the working increment to stakeholders and gathers feedback. It is the best place to show real software, not slides. Feedback here should change the backlog order.

Sprint Retrospectives

The team reflects on how it worked and agrees small improvements.

The retrospective examines how the team worked and agrees small, specific improvements. Keep one or two actions per sprint and track whether they helped. Retrospectives fail when they produce long lists nobody follows.

Backlog Grooming

Older term for refinement: keeping the backlog clear, sized and ordered.

Backlog grooming is the older term for refinement: keeping items clear, sized and ordered. Groom continuously rather than in one large session, so the team is never starting from scratch.

Story Points

A relative unit of effort and complexity for sizing stories, not hours.

Story points measure relative size and complexity, not hours. A story of 5 points should feel about five times the effort of a 1-point story. Use points for forecasting, and do not compare teams by points.

Estimation Techniques

Planning poker, t-shirt sizing, affinity estimation and reference-class estimates to forecast work.

Planning poker, t-shirt sizing and affinity estimation provide quick estimates. Reference-class estimates compare work to similar past items. Estimates are forecasts with error bars, so communicate ranges.

Velocity and Burndown Charts

Velocity is points completed per sprint; burndown shows remaining work over time. Use for forecasting, not for judging people.

Velocity is the points completed per sprint, and burndown shows remaining work over time. They forecast delivery. Using them to judge individuals damages trust and distorts estimates.

Release Planning

Sequencing sprints into releases with dates, scope, dependencies and risk buffers; staged rollout and rollback plans belong here.

Release planning sequences sprints into releases with scope, dates, dependencies and risk buffers. Staged rollouts such as 5%, 25% and 100% of users, with rollback plans, belong here. A release is ready when its go/no-go criteria are met.

Dependency Management

Identifying and tracking work that relies on other teams, systems or vendors, and resolving early.

A dependency is work that relies on another team, system or vendor. List dependencies early, agree dates with those teams and track them in each sprint review. Hidden dependencies cause most missed release dates.

Risk and Issue Management

Risks are possible problems to mitigate; issues are problems that have occurred and need action.

A risk is a possible problem you can prevent or reduce. An issue is a problem that has already happened and needs action. Keep a short risk register with owners, likelihood, impact and the mitigation date.

Cross-Functional Team Collaboration

Product, design, engineering, data, QA and go-to-market working toward one goal with shared context.

Product, design, engineering, data, QA and go-to-market work toward one goal with shared context. Shared goals and regular demos prevent the handoff delays of separate silos.

Engineering Collaboration

Working with engineers on trade-offs, feasibility and scope, treating them as partners from discovery onward.

Involve engineers from discovery onward. They know the cost of options and can suggest simpler solutions. Treat trade-offs on scope and quality as joint decisions.

Product Delivery Lifecycle

Idea to discovery, design, build, test, release, measure and iterate.

The lifecycle runs from idea to discovery, design, build, test, release, measurement and iteration. Measurement closes the loop: the results decide whether to invest more, change direction or stop.

Jira

Issue and sprint tracking tool for backlogs, boards, workflows and reports.

Jira tracks issues, sprints, boards and workflows. Keep the workflow simple, with few statuses, and use a clear naming and labelling convention. Reports are useful only if the data is kept current.

Confluence

Team wiki used for specs, decision records and knowledge sharing, integrated with Jira.

Confluence is a team wiki for specifications, decision records and knowledge. Link pages to Jira issues so context stays connected to the work. Archive old pages so people can find current ones.

Linear and Azure DevOps

Linear is a fast, opinionated issue tracker; Azure DevOps combines boards, repos and pipelines for engineering teams.

Linear is a fast, opinionated issue tracker popular with product teams. Azure DevOps combines boards, repositories and pipelines for engineering organisations. Choose the tool that fits how your engineering team already works.

Summary of terms

Term	Meaning
Agile Fundamentals	Iterative delivery, frequent feedback and adapting to change over following a fixed plan.
Agile Manifesto	Four values and twelve principles favouring individuals, working software, collaboration and responding to change.
Scrum Framework	Fixed-length sprints with defined roles, events and artefacts to deliver increments.
Kanban Methodology	A continuous flow board that limits work in progress and optimises throughput.
Scrum Roles and Responsibilities	Product Owner, Scrum Master and Developers, each accountable for a distinct part of the work.
Product Owner Responsibilities	Maximising product value by managing and ordering the backlog and accepting work.
Sprint Planning	The team selects backlog items and plans how to deliver them within the sprint goal.
Daily Stand-ups	A 15-minute sync on progress toward the sprint goal and blockers.
Sprint Reviews	Demonstrating the increment to stakeholders and gathering feedback.
Sprint Retrospectives	The team reflects on how it worked and agrees small improvements.
Backlog Grooming	Older term for refinement: keeping the backlog clear, sized and ordered.
Story Points	A relative unit of effort and complexity for sizing stories, not hours.
Estimation Techniques	Planning poker, t-shirt sizing, affinity estimation and reference-class estimates to forecast work.
Velocity and Burndown Charts	Velocity is points completed per sprint; burndown shows remaining work over time. Use for forecasting, not for judging people.

Release Planning	Sequencing sprints into releases with dates, scope, dependencies and risk buffers; staged rollout and rollback plans belong here.
Dependency Management	Identifying and tracking work that relies on other teams, systems or vendors, and resolving early.
Risk and Issue Management	Risks are possible problems to mitigate; issues are problems that have occurred and need action.
Cross-Functional Team Collaboration	Product, design, engineering, data, QA and go-to-market working toward one goal with shared context.
Engineering Collaboration	Working with engineers on trade-offs, feasibility and scope, treating them as partners from discovery onward.
Product Delivery Lifecycle	Idea to discovery, design, build, test, release, measure and iterate.
Jira	Issue and sprint tracking tool for backlogs, boards, workflows and reports.
Confluence	Team wiki used for specs, decision records and knowledge sharing, integrated with Jira.
Linear and Azure DevOps	Linear is a fast, opinionated issue tracker; Azure DevOps combines boards, repos and pipelines for engineering teams.

Practice exercise: Break one epic into stories small enough for a sprint, estimate them with planning poker and draft the sprint goal.

Self-check

1. Explain Agile Fundamentals in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Story Points in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Linear and Azure DevOps in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- 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.



AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Product Analytics and Data-Driven Decision Making

The data skills a PM uses to answer questions, measure behaviour and run experiments that inform decisions.

Chapter 07 of 26 · 26 topics · Edition 2026 · skillnavora.in

CHAPTER 07

Product Analytics and Data-Driven Decision Making

The data skills a PM uses to answer questions, measure behaviour and run experiments that inform decisions.

Data Fundamentals for Product Managers

Data types, sources, quality, events vs tables, and the habit of asking what decision the data should inform.

Data types include numbers, categories, timestamps and text. Events record what a user did, while tables store entities such as accounts. Before analysing anything, ask what decision the data should inform.

Descriptive Statistics

Mean, median, mode, spread and percentiles to summarise data; prefer medians and percentiles for skewed metrics.

Mean, median, mode, spread and percentiles summarise data. Use medians and percentiles for skewed measures such as session length, because a few very long sessions distort the mean.

Probability Fundamentals

Likelihood, conditional probability and distributions, the basis for reasoning under uncertainty.

Probability measures how likely an outcome is. Conditional probability asks how likely an outcome is given another event, such as churn given a failed payment. Distributions describe the range of possible values for a metric.

Hypothesis Testing

Testing whether an observed difference is likely due to chance, using null hypothesis, p-values and significance level.

Hypothesis testing asks whether an observed difference is likely due to chance. The null hypothesis states there is no difference. A p-value below a pre-set threshold, often 0.05, suggests the difference is unlikely to be chance, but it does not measure business importance.

SQL Fundamentals

SELECT, WHERE, GROUP BY, ORDER BY and LIMIT to query tables.

SELECT, WHERE, GROUP BY, ORDER BY and LIMIT let you query tables. Example: SELECT city, COUNT(*) FROM orders WHERE status = 'paid' GROUP BY city returns paid orders per city.

SQL Joins and Aggregations

Combining tables (inner, left) and summarising with COUNT, SUM, AVG across groups.

Joins combine tables on keys. Inner joins keep matches only; left joins keep all rows from the left table. Aggregations such as COUNT, SUM and AVG summarise groups. Check row counts after each join to avoid double counting.

Advanced SQL for Product Analytics

Window functions, CTEs, date handling and cohort queries for retention and funnel analysis.

Window functions, common table expressions and date functions answer product questions such as "what share of users returned on day 7?" Build cohort tables step by step and check each intermediate result.

Excel and Google Sheets

Pivot tables, lookups, formulas and charts for fast analysis and sharing.

Pivot tables, lookups, formulas and charts are fast tools for analysis and sharing. Use them for quick exploration and small datasets. Move to SQL and a BI tool when data grows or needs refreshing.

Data Visualisation

Choosing charts that make comparisons obvious, with clear labels and no distortion.

Choose charts that make the comparison obvious: bars for categories, lines for trends. Label axes clearly, start bars at zero and avoid decorative effects that distort the message.

Tableau and Power BI

BI tools for interactive dashboards and self-serve reporting on warehouse data.

These BI tools build interactive dashboards and self-serve reports on warehouse data. Good dashboards show a small set of decisions and the metric definitions beside them, so people interpret the numbers the same way.

Product Analytics Fundamentals

Tracking user behaviour as events with properties, then analysing funnels, cohorts and paths.

Product analytics records user behaviour as events with properties, then analyses funnels, cohorts and paths. Clean event names and a tracking plan prevent confusing results later.

Funnel Analysis

Measuring conversion between steps to find where users drop off.

A funnel measures conversion between steps, such as visit, signup, activation and purchase. The biggest drop is a starting point for investigation, not an answer. Segment the funnel by source and device to see who drops off.

Cohort Analysis

Grouping users by a shared start (such as signup week) and following them over time.

A cohort groups users by a shared start, such as signup week. Following each cohort shows whether newer users behave better or worse than older ones over time.

Retention Analysis

How many users return over days or weeks; the retention curve flattening signals product-market fit.

Retention shows how many users return over days or weeks. A retention curve that flattens at a healthy level is a common sign of product-market fit, while a curve that keeps falling signals a problem.

Churn Analysis

Why and when customers leave, using exit surveys, usage patterns and segment comparison.

Churn analysis asks why and when customers leave, using exit surveys, usage before cancellation and segment comparisons. Churn usually follows a clear drop in usage, so track engagement as an early warning.

Conversion Rate Optimisation

Improving the share of visitors who complete a goal through research, tests and friction removal.

CRO improves the share of visitors who complete a goal through research, testing and friction removal. Start with the steps where users stall, then test changes against a clear hypothesis.

Customer Segmentation

Grouping users by behaviour, value or needs to tailor products and messages.

Segments group users by behaviour, value or need. A segment is useful when it suggests a different product choice or message. Segments based only on demographics often fail to change any decision.

A/B Testing

Randomly split users between versions and compare a pre-chosen metric to measure causal impact.

An A/B test randomly splits users between two versions and compares a pre-chosen metric. Randomisation makes the difference causal. Decide the sample size and duration before starting, and do not stop early because one result looks good.

Multivariate Testing

Testing combinations of multiple elements at once; needs more traffic than A/B.

Multivariate tests vary several elements at once to find the best combination. They need far more traffic than A/B tests. Use them only when you have enough users and a clear reason to test interactions.

Experimentation Frameworks

A repeatable process: hypothesis, metric, sample size, guardrails, run length, analysis and decision.

A framework sets a repeatable process: hypothesis, primary metric, guardrails, sample size, run length, analysis and decision rule. Teams that follow one framework make fewer false conclusions.

Event Tracking

Defining and logging user actions with a consistent naming plan so analysis is trustworthy.

Event tracking defines and logs user actions with a consistent naming plan, for example "checkout_started" with properties. Review the plan when features change so historical data remains comparable.

Product Analytics Dashboards

Live views of key metrics for the team, with a clear owner, definitions and thresholds for action.

A dashboard shows key metrics for the team, with a clear owner, definitions and thresholds that trigger action. A dashboard without an owner tends to be ignored within months.

Mixpanel

Event-based product analytics for funnels, retention and cohorts.

Mixpanel is event-based analytics for funnels, retention and cohorts. It suits teams that want quick self-serve analysis of user behaviour in web and mobile products.

Amplitude

Product analytics platform with behavioural cohorts, journeys and experimentation.

Amplitude offers behavioural cohorts, user journeys and experimentation tools. Like Mixpanel, it works from events, so the quality of the tracking plan determines the quality of the analysis.

Google Analytics

Web and app analytics for traffic, acquisition and conversion.

Google Analytics measures web and app traffic, acquisition channels and conversions. It is a common starting point for marketing data. Its event model and reports differ from product analytics tools, so check definitions before comparing numbers.

Data-Driven Product Decisions

Combining quantitative data with qualitative insight, stating the decision rule in advance.

Combine quantitative data with qualitative insight. Write the decision rule before looking at results, for example "we will launch if activation rises by at least 3 points without lowering revenue per user". This prevents rationalising the result afterward.

Summary of terms

Term	Meaning
Data Fundamentals for Product Managers	Data types, sources, quality, events vs tables, and the habit of asking what decision the data should inform.
Descriptive Statistics	Mean, median, mode, spread and percentiles to summarise data; prefer medians and percentiles for skewed metrics.
Probability Fundamentals	Likelihood, conditional probability and distributions, the basis for reasoning under uncertainty.

Hypothesis Testing	Testing whether an observed difference is likely due to chance, using null hypothesis, p-values and significance level.
SQL Fundamentals	SELECT, WHERE, GROUP BY, ORDER BY and LIMIT to query tables.
SQL Joins and Aggregations	Combining tables (inner, left) and summarising with COUNT, SUM, AVG across groups.
Advanced SQL for Product Analytics	Window functions, CTEs, date handling and cohort queries for retention and funnel analysis.
Excel and Google Sheets	Pivot tables, lookups, formulas and charts for fast analysis and sharing.
Data Visualisation	Choosing charts that make comparisons obvious, with clear labels and no distortion.
Tableau and Power BI	BI tools for interactive dashboards and self-serve reporting on warehouse data.
Product Analytics Fundamentals	Tracking user behaviour as events with properties, then analysing funnels, cohorts and paths.
Funnel Analysis	Measuring conversion between steps to find where users drop off.
Cohort Analysis	Grouping users by a shared start (such as signup week) and following them over time.
Retention Analysis	How many users return over days or weeks; the retention curve flattening signals product-market fit.
Churn Analysis	Why and when customers leave, using exit surveys, usage patterns and segment comparison.
Conversion Rate Optimisation	Improving the share of visitors who complete a goal through research, tests and friction removal.
Customer Segmentation	Grouping users by behaviour, value or needs to tailor products and messages.
A/B Testing	Randomly split users between versions and compare a pre-chosen metric to measure causal impact.
Multivariate Testing	Testing combinations of multiple elements at once; needs more traffic than A/B.
Experimentation Frameworks	A repeatable process: hypothesis, metric, sample size, guardrails, run length, analysis and decision.
Event Tracking	Defining and logging user actions with a consistent naming plan so analysis is trustworthy.
Product Analytics Dashboards	Live views of key metrics for the team, with a clear owner, definitions and thresholds for action.
Mixpanel	Event-based product analytics for funnels, retention and cohorts.
Amplitude	Product analytics platform with behavioural cohorts, journeys and experimentation.
Google Analytics	Web and app analytics for traffic, acquisition and conversion.
Data-Driven Product Decisions	Combining quantitative data with qualitative insight, stating the decision rule in advance.

Practice exercise: Write SQL for a weekly signup-to-activation funnel and week-1 retention on a sample dataset, and state what decision each result supports.

Self-check

1. Explain Data Fundamentals for Product Managers in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Retention Analysis in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Data-Driven Product Decisions in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Product Metrics and Business Analytics

The metrics that describe product health and business performance, and how to read and review them.

Chapter 08 of 26 · 20 topics · Edition 2026 · skillnavora.in



CHAPTER 08

Product Metrics and Business Analytics

The metrics that describe product health and business performance, and how to read and review them.

Product KPIs

A small set of measures tied to goals that show whether the product is succeeding.

Product KPIs are a small set of measures tied to goals that show whether the product is succeeding. Pick five or fewer, and define each one precisely, including the numerator, denominator, time window and filters.

North Star Metrics

The one measure of delivered customer value that guides the whole team, supported by input metrics.

The north star is the one measure of the value customers receive, such as completed bookings per week for a travel app. It should lead revenue and be supported by input metrics the teams can influence.

Acquisition Metrics

How users arrive: visitors, signups, cost per acquisition and channel conversion.

Acquisition metrics show how users arrive: visitors, signups, cost per acquisition and conversion by channel. Compare channels on the quality of the users they bring, not only the volume.

Activation Metrics

Whether new users reach the first moment of value, such as completing a key action.

Activation measures whether new users reach their first moment of value, such as sending a first invoice. Define the activation event from data showing which early actions predict long-term retention.

Engagement Metrics

How deeply and often users use the product: sessions, actions, time in product, stickiness.

Engagement measures how deeply and often users use the product: sessions, key actions, time in product and stickiness. Engagement should be tied to value, because time spent alone can reflect confusion.

Retention Metrics

The share of users or revenue that remains over time.

Retention is the share of users or revenue that remains over time. Report it by cohort and over several time periods, so changes in product quality are visible.

Revenue Metrics

Revenue, ARPU, expansion and contraction, margin and bookings.

Revenue metrics include total revenue, average revenue per user, expansion and contraction, margin and bookings. Separate recurring and one-time revenue to see the health of the subscription base.

Customer Satisfaction Metrics

Direct measures of how customers feel about the product or interaction.

These capture how customers feel about the product or an interaction. Satisfaction is a lagging signal, so pair it with behavioural metrics to see whether opinions match actions.

NPS, CSAT and CES

NPS asks likelihood to recommend; CSAT rates satisfaction with an interaction; CES measures how easy a task was.

NPS asks how likely people are to recommend the product, CSAT rates satisfaction with an interaction, and CES measures how much effort a task took. Use CES after support interactions and NPS for relationship health.

DAU, WAU and MAU

Daily, weekly and monthly active users; DAU/MAU indicates stickiness.

Daily, weekly and monthly active users count distinct users. The DAU/MAU ratio indicates stickiness. A product used once a month needs different measures than one used every day.

ARPU and ARPPU

Average revenue per user and per paying user.

ARPU is average revenue per user across all users, and ARPPU is average revenue per paying user. Comparing them shows how much of the change comes from conversion to paid versus price or upsell.

Customer Lifetime Value (LTV)

Expected gross profit from a customer over their lifetime, from average revenue, margin and lifespan.

LTV is the expected gross profit from a customer over their relationship with the product. A simple version multiplies average monthly gross profit by average lifespan in months. Use it with CAC to decide how much acquisition spending is sustainable.

Customer Acquisition Cost (CAC)

Total sales and marketing spend divided by customers acquired in the period.

CAC is total sales and marketing spend divided by the number of new customers acquired in the same period. Track it by channel, and be clear about whether it includes sales salaries.

LTV/CAC Ratio

Return on acquisition spend; around 3 or higher is often a healthy target for subscription businesses.

LTV divided by CAC shows the return on acquisition spending. A ratio around 3 is often cited as healthy for subscription businesses, but the benchmark depends on payback period and cash position.

Monthly Recurring Revenue (MRR)

Predictable subscription revenue normalised to a month.

MRR is predictable subscription revenue normalised to a month. Track its components: new, expansion, contraction and churned MRR. The movement between them explains growth or decline.

Annual Recurring Revenue (ARR)

Predictable subscription revenue annualised, typically MRR x 12.

ARR annualises recurring revenue, typically MRR multiplied by 12. It is useful for comparing subscription businesses, but it must exclude one-time fees and be defined consistently.

Churn and Retention Rate

Churn is the share lost in a period; retention is the share kept, and the two sum to one.

Churn is the share of customers or revenue lost in a period, and retention is the share kept. They sum to 100% over the same base. Report logo churn and revenue churn separately, since large customers can distort revenue figures.

Feature Adoption

The share of eligible users who try and keep using a feature.

Feature adoption is the share of eligible users who try a feature and keep using it. A launched feature used by 2% of users after three months needs investigation into discovery, value or fit.

Product Health Metrics

A balanced set: reliability, performance, quality, support load, satisfaction and usage.

A balanced set includes reliability, performance, quality, support load, satisfaction and usage. Health metrics warn of problems before growth metrics decline.

Product Performance Reviews

A regular forum to read metrics, explain changes, agree actions and adjust the roadmap.

A regular review reads metrics, explains changes, agrees actions and adjusts the roadmap. Each review should end with named owners and dates for the actions it creates.

Summary of terms

Term	Meaning
Product KPIs	A small set of measures tied to goals that show whether the product is succeeding.

North Star Metrics	The one measure of delivered customer value that guides the whole team, supported by input metrics.
Acquisition Metrics	How users arrive: visitors, signups, cost per acquisition and channel conversion.
Activation Metrics	Whether new users reach the first moment of value, such as completing a key action.
Engagement Metrics	How deeply and often users use the product: sessions, actions, time in product, stickiness.
Retention Metrics	The share of users or revenue that remains over time.
Revenue Metrics	Revenue, ARPU, expansion and contraction, margin and bookings.
Customer Satisfaction Metrics	Direct measures of how customers feel about the product or interaction.
NPS, CSAT and CES	NPS asks likelihood to recommend; CSAT rates satisfaction with an interaction; CES measures how easy a task was.
DAU, WAU and MAU	Daily, weekly and monthly active users; DAU/MAU indicates stickiness.
ARPU and ARPPU	Average revenue per user and per paying user.
Customer Lifetime Value (LTV)	Expected gross profit from a customer over their lifetime, from average revenue, margin and lifespan.
Customer Acquisition Cost (CAC)	Total sales and marketing spend divided by customers acquired in the period.
LTV/CAC Ratio	Return on acquisition spend; around 3 or higher is often a healthy target for subscription businesses.
Monthly Recurring Revenue (MRR)	Predictable subscription revenue normalised to a month.
Annual Recurring Revenue (ARR)	Predictable subscription revenue annualised, typically MRR x 12.
Churn and Retention Rate	Churn is the share lost in a period; retention is the share kept, and the two sum to one.
Feature Adoption	The share of eligible users who try and keep using a feature.
Product Health Metrics	A balanced set: reliability, performance, quality, support load, satisfaction and usage.
Product Performance Reviews	A regular forum to read metrics, explain changes, agree actions and adjust the roadmap.

Practice exercise: Define a north star and five input metrics for a product, with exact formulas and the guardrail for each.

Self-check

1. Explain Product KPIs in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain ARPU and ARPPU in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Product Performance Reviews in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Business, Finance and Product Commercialisation

How products make money and how a PM builds a financial case, sets price and takes the product to market.

Chapter 09 of 26 · 24 topics · Edition 2026 · skillnavora.in



CHAPTER 09

Business, Finance and Product Commercialisation

How products make money and how a PM builds a financial case, sets price and takes the product to market.

Business Fundamentals

Revenue, cost, profit, cash flow, margins and how value flows through a company.

A business earns revenue and pays costs to make profit. Cash flow timing matters as much as profit, because a growing business can still run out of cash. Product decisions change revenue, cost and cash, so PMs need this vocabulary.

Business Model Canvas

Nine blocks describing customers, value, channels, revenue, resources, activities, partners and costs.

The canvas has nine blocks: customer segments, value propositions, channels, customer relationships, revenue streams, key resources, key activities, key partners and cost structure. It helps check that the parts of a business fit together.

Revenue Models

Subscription, transaction fee, advertising, licensing, usage-based, marketplace commission and services.

Common models include subscription, transaction fee, advertising, licensing, usage-based pricing, marketplace commission and services. Each model shapes what the product should optimise for.

Subscription Business Models

Recurring fees for continued access, optimising retention, expansion and predictable revenue.

Customers pay a recurring fee for continued access. Revenue is predictable, but retention becomes the main lever. Improving onboarding and value delivery matters as much as acquisition.

Freemium Models

A free tier drives adoption and conversion to paid; works when marginal cost is low and upgrade triggers are clear.

A free tier drives adoption and conversion to paid plans. It works when the cost of serving free users is low and the upgrade trigger is clear, such as a limit that users hit when they are already getting value.

Usage-Based Pricing

Customers pay according to consumption, aligning price with value but making revenue less predictable.

Customers pay according to consumption, such as API calls or messages sent. The price matches value, but revenue is less predictable, so forecasting and customer budgets need extra attention.

Product Pricing Strategy

Setting price using cost, competitors and value to the customer, with packaging and tiers that fit segments.

Pricing uses cost, competitor prices and the value to the customer. Packaging into tiers fits different segments. Test price changes on new customers first, because changing prices for existing customers creates churn risk.

Monetisation Strategies

The mechanisms by which a product earns revenue and when in the user journey they appear.

These are the mechanisms by which a product earns money and when in the journey they appear. A paywall shown before any value is delivered often reduces activation.

Financial Modelling

A spreadsheet model of revenue, costs and cash under stated assumptions, with scenarios.

A financial model is a spreadsheet of revenue, costs and cash under stated assumptions. Build best, base and worst scenarios and state which assumptions matter most, so the model supports decisions rather than hiding risk.

Unit Economics

Profit or loss per unit (customer or transaction) after variable costs, such as CAC, LTV and payback period.

Unit economics measures profit or loss per customer or transaction after variable costs. Key measures are CAC, LTV and payback period. Healthy unit economics mean each customer pays back acquisition cost within an acceptable time.

ROI and ROE

Return on investment is gain relative to cost; return on equity is net income relative to shareholder equity.

ROI is the gain from an investment divided by its cost. ROE is net income divided by shareholder equity. For product work, ROI is the more common tool to compare investment options.

Cost-Benefit Analysis

Comparing total expected costs and benefits, in money terms, to decide.

This compares total expected costs and benefits in money terms, including time saved and risk reduced. Quantify benefits conservatively and show the assumptions, so the comparison is credible.

Break-Even Analysis

The volume at which revenue equals total cost; fixed cost divided by contribution margin per unit.

Break-even is the volume at which revenue equals total cost. It equals fixed costs divided by contribution margin per unit. It shows how many customers a new product must win to stop losing money.

Profitability Analysis

Assessing margins by product, segment or channel to see where money is made.

Profitability analysis looks at margins by product, segment or channel to see where money is made or lost. A profitable product line can hide an unprofitable segment that drives support costs.

Business Case Development

A structured argument for an investment: problem, options, costs, benefits, risks and recommendation.

A business case argues for an investment: problem, options, costs, benefits, risks and a recommendation. Lead with the decision requested. Include the cost of doing nothing.

Budget Planning

Allocating funds to teams and initiatives, tracking against plan and forecasting.

Budget planning allocates funds to teams and initiatives, tracks spending against plan and forecasts the rest of the year. Keep a reserve for unexpected work, such as an urgent compliance change.

Go-to-Market Strategy

Who you sell to, with what message and price, through which channels, and how you measure launch.

The GTM strategy defines who the product is for, the message, the price, the channels and how success will be measured after launch. It connects product decisions with sales and marketing.

Product Launch Strategy

A staged plan for readiness, beta, announcement, enablement and post-launch monitoring.

A launch plan covers readiness checks, a beta with selected users, announcement, sales enablement, support preparation and post-launch monitoring. The launch date should follow the readiness checklist, not the other way round.

Product Adoption

Moving customers from awareness to trial to regular use through onboarding and value moments.

Adoption moves customers from awareness to trial to regular use. Onboarding, first success and reminders at the right moments drive adoption more than announcements.

Customer Acquisition

Channels and tactics to attract customers efficiently: content, paid, partners, sales and referrals.

Acquisition channels include content, paid media, partners, sales and referrals. Compare channels by cost, quality of customers and payback. A channel that brings many customers who never activate is expensive.

Customer Retention Strategy

Keeping customers through onboarding, ongoing value, support, community and proactive outreach.

Retention comes from onboarding, ongoing value, support, community and proactive outreach to users who show declining engagement. Retention improvements often pay back faster than new acquisition.

Product Growth Strategy

Loops and levers across acquisition, activation, retention, revenue and referral.

Growth strategy identifies loops and levers across acquisition, activation, retention, revenue and referral. Choose the lever with the largest gap in the funnel and test changes against it.

Product-Led Sales

Using product usage signals to identify and convert qualified accounts, blending self-serve with sales.

Product usage signals identify accounts ready to buy. Sales then engages those accounts with context from the product. It blends self-serve adoption with human sales for larger deals.

Enterprise Product Commercialisation

Selling to large organisations: security reviews, procurement, pilots, integration and long sales cycles.

Selling to large organisations involves security reviews, procurement, pilots, integration and long sales cycles. Plan these steps into the product roadmap, such as single sign-on and audit logs, because they often decide the deal.

Summary of terms

Term	Meaning
Business Fundamentals	Revenue, cost, profit, cash flow, margins and how value flows through a company.
Business Model Canvas	Nine blocks describing customers, value, channels, revenue, resources, activities, partners and costs.
Revenue Models	Subscription, transaction fee, advertising, licensing, usage-based, marketplace commission and services.
Subscription Business Models	Recurring fees for continued access, optimising retention, expansion and predictable revenue.
Freemium Models	A free tier drives adoption and conversion to paid; works when marginal cost is low and upgrade triggers are clear.
Usage-Based Pricing	Customers pay according to consumption, aligning price with value but making revenue less predictable.
Product Pricing Strategy	Setting price using cost, competitors and value to the customer, with packaging and tiers that fit segments.

Monetisation Strategies	The mechanisms by which a product earns revenue and when in the user journey they appear.
Financial Modelling	A spreadsheet model of revenue, costs and cash under stated assumptions, with scenarios.
Unit Economics	Profit or loss per unit (customer or transaction) after variable costs, such as CAC, LTV and payback period.
ROI and ROE	Return on investment is gain relative to cost; return on equity is net income relative to shareholder equity.
Cost-Benefit Analysis	Comparing total expected costs and benefits, in money terms, to decide.
Break-Even Analysis	The volume at which revenue equals total cost; fixed cost divided by contribution margin per unit.
Profitability Analysis	Assessing margins by product, segment or channel to see where money is made.
Business Case Development	A structured argument for an investment: problem, options, costs, benefits, risks and recommendation.
Budget Planning	Allocating funds to teams and initiatives, tracking against plan and forecasting.
Go-to-Market Strategy	Who you sell to, with what message and price, through which channels, and how you measure launch.
Product Launch Strategy	A staged plan for readiness, beta, announcement, enablement and post-launch monitoring.
Product Adoption	Moving customers from awareness to trial to regular use through onboarding and value moments.
Customer Acquisition	Channels and tactics to attract customers efficiently: content, paid, partners, sales and referrals.
Customer Retention Strategy	Keeping customers through onboarding, ongoing value, support, community and proactive outreach.
Product Growth Strategy	Loops and levers across acquisition, activation, retention, revenue and referral.
Product-Led Sales	Using product usage signals to identify and convert qualified accounts, blending self-serve with sales.
Enterprise Product Commercialisation	Selling to large organisations: security reviews, procurement, pilots, integration and long sales cycles.

Practice exercise: Build a small model of revenue, CAC, LTV and payback for a subscription product with best, base and worst cases.

Self-check

1. Explain Business Fundamentals in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Break-Even Analysis in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Enterprise Product Commercialisation in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- AI Product Management (original programme) : The programme you already have: PM Master Handbook, PRD and BRD templates, interview prep.
- Interactive PM Handbook : 23 modules: product core, AI tools lab, frameworks, case studies and interview prep.
- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Technical Foundations for Product Managers

Enough software and systems knowledge to discuss trade-offs credibly with engineers and make sound decisions.

Chapter 10 of 26 · 28 topics · Edition 2026 · skillnavora.in



CHAPTER 10

Technical Foundations for Product Managers

Enough software and systems knowledge to discuss trade-offs credibly with engineers and make sound decisions.

How Software Applications Work

Users interact with a front end that calls back-end services, which read and write data and return results.

A user interacts with a front end, which sends requests to back-end services. Those services apply business rules, read and write data and return results. Knowing this flow helps a PM describe where a bug or delay happens.

Software Development Lifecycle (SDLC)

Plan, design, build, test, deploy and maintain, in sequential or iterative form.

The SDLC covers planning, design, build, test, deploy and maintain. It can run in sequence or iteratively. Each stage has outputs such as requirements, designs, code and release notes that the PM reviews.

Frontend Fundamentals

HTML, CSS and JavaScript in the browser or native code on mobile, handling what users see and touch.

Frontend code (HTML, CSS and JavaScript on the web, or native code on mobile) controls what users see and touch. Performance on frontend depends on page weight, rendering and network requests.

Backend Fundamentals

Server logic, business rules, databases and integrations that power the front end.

The back end runs server logic, business rules, databases and integrations. Latency and reliability problems often start here, so PMs should understand what runs on the back end and what each request costs.

Client-Server Architecture

Clients send requests; servers process and respond, over a network.

Clients send requests and servers process them and send responses over a network. Most apps follow this pattern. Consider where data is processed when designing offline or low-bandwidth features.

Databases and Data Storage

Persistent stores for application data, chosen for structure, scale and query needs.

Databases persist application data. The choice depends on structure, scale, consistency and query needs. Changing a database later is expensive, so involve engineers early in data decisions.

SQL and NoSQL Databases

SQL stores structured tables with strong consistency; NoSQL covers document, key-value, column and graph stores built for flexibility and scale.

SQL databases store structured tables with strong consistency and joins. NoSQL covers document, key-value, column and graph stores that trade some consistency or queryability for flexibility and scale.

Relational Database Concepts

Tables, keys, relationships, normalisation and transactions (ACID).

Relational databases use tables, keys, relationships and normalisation. Transactions follow ACID properties (atomicity, consistency, isolation, durability), which keep money movements correct.

APIs and API Fundamentals

Contracts that let software components talk, with defined requests, responses and rules.

An API is a contract that lets software components communicate. It defines requests, responses, authentication and error behaviour. Good APIs are stable and documented, because other teams depend on them.

REST APIs

Resource-based HTTP APIs using verbs like GET, POST, PUT and DELETE and status codes.

REST exposes resources through HTTP verbs such as GET, POST, PUT and DELETE, and uses status codes to report results. Example: GET /orders/123 returns one order.

GraphQL

A query language where clients request exactly the fields they need from one endpoint.

GraphQL lets clients request exactly the fields they need from a single endpoint. It reduces over-fetching for complex screens but needs care with query cost and caching.

Webhooks

Event-driven callbacks where one system sends an HTTP request to another when something happens.

A webhook sends an HTTP request from one system to another when an event occurs, such as payment succeeded. It avoids constant polling, but the receiver must handle retries and duplicate events.

Authentication and Authorisation

Authentication proves who you are; authorisation decides what you may do (OAuth, SSO, roles).

Authentication proves identity, for example with a password or single sign-on. Authorisation decides what that identity may do, using roles or permissions. OAuth is a common standard for delegated access.

JSON and XML

Text formats for exchanging structured data; JSON is lighter and more common today.

JSON and XML are text formats for structured data. JSON is lighter and standard for web APIs. Knowing the format helps a PM read API responses and specifications.

API Documentation

Reference, guides and examples (often OpenAPI) that make APIs easy to adopt.

Good documentation includes reference pages, guides and examples, often generated from an OpenAPI specification. Clear docs cut support requests and speed up integration.

Postman

A tool to explore, test and document APIs by sending requests and viewing responses.

Postman lets you send requests, inspect responses and save collections of API calls. PMs use it to understand how an API behaves before writing requirements.

Git and GitHub Fundamentals

Git tracks changes; GitHub hosts repositories and supports pull requests and reviews.

Git tracks changes to code. GitHub hosts repositories and supports pull requests and code review. Knowing the basic workflow lets a PM follow release discussions and documentation changes.

Version Control

Recording history of changes so teams can collaborate, branch, review and roll back.

Version control records history so teams can collaborate, branch, review and roll back. Rollbacks depend on it, and it is the audit trail for what changed in each release.

Microservices Architecture

Many small, independently deployable services that communicate over APIs.

Microservices split an application into small, independently deployable services that talk over APIs. They help large teams work in parallel, but add operational complexity and network failure modes.

Monolithic Architecture

One deployable unit containing all functionality: simple to start, harder to scale teams.

A monolith is one deployable unit containing all functionality. It is simpler to start and debug, but large teams can slow each other down. Many successful products begin as monoliths.

System Design Fundamentals

Choosing components, data flow and trade-offs to meet requirements for scale, cost and reliability.

System design chooses components, data flows and trade-offs to meet requirements for scale, cost and reliability. PMs do not design systems alone, but they should know which trade-offs their requirements create.

Scalability and Reliability

Handling growth in load and staying available despite failures, using redundancy, caching and graceful degradation.

Scalability handles growing load, and reliability keeps the service available despite failures. Techniques include redundancy, caching, load balancing and graceful degradation, such as showing cached data when a service is slow.

Latency and Throughput

Latency is time per request; throughput is requests handled per unit time.

Latency is the time for one request. Throughput is the number of requests handled per unit time. A system can have high throughput and poor latency, so track both.

Cloud Computing Fundamentals

On-demand computing, storage and services paid by use: IaaS, PaaS, SaaS, regions and shared responsibility.

Cloud services provide compute, storage and managed services on demand, paid by use. Models include IaaS, PaaS and SaaS. Responsibility for security is shared between provider and customer.

AWS Fundamentals

Amazon's cloud: EC2, S3, Lambda, RDS and IAM as core building blocks.

AWS offers services such as EC2 for virtual machines, S3 for storage, Lambda for functions, RDS for managed databases and IAM for access control. Knowing these names helps in technical discussions.

Microsoft Azure Fundamentals

Microsoft's cloud: virtual machines, storage, Azure AD, Azure OpenAI and DevOps.

Azure provides virtual machines, storage, Azure Active Directory for identity, Azure OpenAI for model access and DevOps tools. Many enterprises already use Azure, which affects procurement and integration choices.

Google Cloud Platform Fundamentals

Google's cloud: Compute Engine, Cloud Storage, BigQuery and Vertex AI.

GCP offers Compute Engine, Cloud Storage, BigQuery for analytics and Vertex AI for machine learning. Its data and AI services are often chosen for analytics-heavy products.

Cybersecurity Fundamentals

Confidentiality, integrity, availability, encryption, secure coding, access control and common attacks.

Security covers confidentiality, integrity and availability, plus encryption, secure coding, access control and common attacks. A PM must treat security requirements as product requirements, not as later clean-up.

Summary of terms

Term	Meaning
How Software Applications Work	Users interact with a front end that calls back-end services, which read and write data and return results.
Software Development Lifecycle (SDLC)	Plan, design, build, test, deploy and maintain, in sequential or iterative form.
Frontend Fundamentals	HTML, CSS and JavaScript in the browser or native code on mobile, handling what users see and touch.
Backend Fundamentals	Server logic, business rules, databases and integrations that power the front end.
Client-Server Architecture	Clients send requests; servers process and respond, over a network.
Databases and Data Storage	Persistent stores for application data, chosen for structure, scale and query needs.
SQL and NoSQL Databases	SQL stores structured tables with strong consistency; NoSQL covers document, key-value, column and graph stores built for flexibility and scale.
Relational Database Concepts	Tables, keys, relationships, normalisation and transactions (ACID).
APIs and API Fundamentals	Contracts that let software components talk, with defined requests, responses and rules.
REST APIs	Resource-based HTTP APIs using verbs like GET, POST, PUT and DELETE and status codes.
GraphQL	A query language where clients request exactly the fields they need from one endpoint.
Webhooks	Event-driven callbacks where one system sends an HTTP request to another when something happens.
Authentication and Authorisation	Authentication proves who you are; authorisation decides what you may do (OAuth, SSO, roles).
JSON and XML	Text formats for exchanging structured data; JSON is lighter and more common today.
API Documentation	Reference, guides and examples (often OpenAPI) that make APIs easy to adopt.
Postman	A tool to explore, test and document APIs by sending requests and viewing responses.
Git and GitHub Fundamentals	Git tracks changes; GitHub hosts repositories and supports pull requests and reviews.
Version Control	Recording history of changes so teams can collaborate, branch, review and roll back.
Microservices Architecture	Many small, independently deployable services that communicate over APIs.
Monolithic Architecture	One deployable unit containing all functionality: simple to start, harder to scale teams.
System Design Fundamentals	Choosing components, data flow and trade-offs to meet requirements for scale, cost and reliability.
Scalability and Reliability	Handling growth in load and staying available despite failures, using redundancy, caching and graceful degradation.

Latency and Throughput	Latency is time per request; throughput is requests handled per unit time.
Cloud Computing Fundamentals	On-demand computing, storage and services paid by use: IaaS, PaaS, SaaS, regions and shared responsibility.
AWS Fundamentals	Amazon's cloud: EC2, S3, Lambda, RDS and IAM as core building blocks.
Microsoft Azure Fundamentals	Microsoft's cloud: virtual machines, storage, Azure AD, Azure OpenAI and DevOps.
Google Cloud Platform Fundamentals	Google's cloud: Compute Engine, Cloud Storage, BigQuery and Vertex AI.
Cybersecurity Fundamentals	Confidentiality, integrity, availability, encryption, secure coding, access control and common attacks.

Practice exercise: Call a public REST API in Postman, read the JSON response, and draw the client, server and database flow for that request.

Self-check

1. Explain How Software Applications Work in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain API Documentation in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Cybersecurity Fundamentals in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- System Design and Databases for PMs : Architecture, databases and trade-offs.
- Platform and API Product Management : API products, versioning and developer experience.
- 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.



AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Artificial Intelligence and Machine Learning Foundations

How AI and machine learning work, the main problem types and how models are trained and judged.

Chapter 11 of 26 · 32 topics · Edition 2026 · skillnavora.in

CHAPTER 11

Artificial Intelligence and Machine Learning Foundations

How AI and machine learning work, the main problem types and how models are trained and judged.

Introduction to Artificial Intelligence

Systems that perform tasks associated with human intelligence, such as perception, language and decision making.

AI refers to systems that perform tasks associated with human intelligence, such as perceiving images, understanding language and making decisions. Most AI in products today is machine learning, which learns patterns from data.

History and Evolution of AI

Rules and symbolic AI, expert systems, statistical ML, deep learning from 2012, transformers from 2017, and generative AI.

The field moved from rule-based and symbolic systems to expert systems, then statistical machine learning, deep learning from around 2012, transformer models from 2017 and today's generative models. Each shift changed what products could do.

Traditional AI vs Generative AI

Traditional models classify or predict from data; generative models create new text, images, audio or code.

Traditional models classify or predict from data, such as flagging fraud. Generative models create new text, images, audio or code. Traditional AI outputs a label or number; generative AI outputs content that needs review.

Narrow AI vs General AI

Narrow AI does specific tasks; general AI would match human breadth, and does not yet exist.

Narrow AI performs specific tasks well, such as translation or recommendation. General AI would match human breadth across tasks. Only narrow AI exists today, so products should be designed around specific tasks.

AI Applications Across Industries

Fraud detection, recommendations, diagnostics, document processing, support automation, forecasting and more.

Examples include fraud detection in banking, recommendations in e-commerce, diagnostics in healthcare, document processing in insurance, support automation and demand forecasting in retail.

Machine Learning Fundamentals

Algorithms that learn patterns from data to predict or decide without explicit rules.

Machine learning algorithms learn patterns from data to predict or decide without explicit rules for every case. Quality depends mostly on data quality and on defining the target correctly.

Supervised Learning

Learning from labelled examples to predict labels for new data.

Supervised learning trains on labelled examples to predict labels for new data. Example: historical loan applications labelled repaid or defaulted, used to predict risk for a new applicant.

Unsupervised Learning

Finding structure in unlabelled data, such as clusters or anomalies.

Unsupervised learning finds structure in unlabelled data, such as clusters of similar customers or unusual transactions. Results need interpretation, because no correct answer is given.

Semi-Supervised Learning

Using a small labelled set plus a large unlabelled set.

It uses a small labelled dataset together with a large unlabelled one. This helps when labelling is expensive, such as medical images labelled only by specialists.

Reinforcement Learning

An agent learns by trial and error to maximise cumulative reward.

An agent learns by trial and error to maximise cumulative reward. It suits sequential decisions such as pricing or robot control, but needs a safe simulation or careful rollout.

Classification

Predicting a category, such as spam or not spam.

Classification predicts a category, such as spam or not spam, or churn or stay. Its output is usually a probability, which the product converts into a decision using a threshold.

Regression

Predicting a continuous number, such as price or demand.

Regression predicts a continuous number, such as delivery time or demand next week. Errors are measured as average distance from the true value.

Clustering

Grouping similar items without labels, used for segmentation.

Clustering groups similar items without labels. It is used for customer segmentation and for finding natural groups in support tickets. Clusters must be named and checked by people.

Recommendation Systems

Suggesting items using collaborative filtering, content features or hybrid models.

Recommenders suggest items using collaborative filtering (what similar users chose), content features, or hybrid models. They need careful evaluation for diversity and for avoiding narrow feedback loops.

Predictive Analytics

Using historical data to forecast outcomes such as churn or demand.

Predictive analytics uses historical data to forecast outcomes such as churn, demand or default. A forecast is only useful if someone acts on it, so define the intervention before building the model.

Feature Engineering

Creating informative inputs from raw data to help models learn.

Feature engineering creates informative inputs from raw data, such as days since last purchase. Good features often matter more than a more complex algorithm.

Training, Validation and Testing

Train to fit, validate to tune, and test once on unseen data for an honest estimate.

The model is trained on one set, tuned on a validation set and evaluated once on an unseen test set. Keeping the test set untouched gives an honest estimate of performance.

Model Training and Inference

Training learns parameters from data; inference uses the trained model to make predictions.

Training learns model parameters from data. Inference uses the trained model to make predictions on new inputs. Inference cost and latency matter in production even when training is done once.

Overfitting and Underfitting

Overfitting memorises training noise and fails on new data; underfitting is too simple to capture the pattern.

Overfitting memorises training noise and performs poorly on new data. Underfitting is too simple to capture the pattern. Compare training and validation performance to detect either.

Bias-Variance Trade-Off

Simple models have high bias; complex models have high variance; aim for the balance with best generalisation.

Simple models have high bias and miss patterns. Complex models have high variance and react to noise. The aim is the level of complexity that generalises best to new data.

Model Performance Metrics

Accuracy, precision, recall, F1, AUC for classification; MAE and RMSE for regression, picked to match business cost of errors.

Classification uses accuracy, precision, recall, F1 and AUC. Regression uses mean absolute error and root mean squared error. Pick the metric that matches the business cost of each kind of error.

Precision, Recall and F1 Score

Precision is correctness of positive predictions; recall is coverage of actual positives; F1 balances both.

Precision is the share of positive predictions that are correct. Recall is the share of actual positives found. F1 combines both. In fraud detection, recall often matters more; in automatic account blocking, precision matters more.

Confusion Matrix

A table of true/false positives and negatives that shows the types of errors.

A confusion matrix tabulates true positives, false positives, true negatives and false negatives. It shows what kinds of mistakes a model makes, which a single accuracy number hides.

ROC-AUC

Area under the curve of true-positive vs false-positive rate across thresholds; 0.5 is chance, 1.0 is perfect.

The ROC curve plots true-positive rate against false-positive rate across thresholds. AUC is the area under it: 0.5 means chance and 1.0 means perfect ranking.

Deep Learning Fundamentals

Multi-layer neural networks that learn features automatically from large data.

Deep learning uses multi-layer neural networks that learn features directly from large data, such as images or text. It needs more data and compute than classical methods.

Neural Networks

Layers of connected units with weights, trained by backpropagation and gradient descent.

Neural networks connect layers of units with weights. Training adjusts the weights with backpropagation and gradient descent to reduce error.

CNN, RNN and LSTM

CNNs process images, RNNs process sequences, LSTMs handle longer dependencies in sequences.

CNNs process images using local patterns. RNNs process sequences step by step. LSTMs handle longer dependencies in sequences. Transformers have largely replaced RNNs for language tasks.

Transformers

An architecture based on attention that processes sequences in parallel and underlies modern language models.

Transformers process sequences in parallel using attention. They underpin modern language and multimodal models, and they scale well with data and compute.

Attention Mechanisms

A way for a model to weigh which parts of the input matter for each output.

Attention lets the model weigh which parts of the input matter for each output. It is the reason models can relate a word to a distant word in the same document.

Natural Language Processing

Techniques for understanding and generating human language, from tokenisation to LLMs.

NLP covers understanding and generating human language, from tokenisation and classification to question answering. Large language models now handle most NLP tasks in products.

Computer Vision Fundamentals

Techniques for classifying, detecting and segmenting content in images and video.

Computer vision classifies, detects and segments content in images and video. Applications include document scanning, quality inspection and product tagging.

AI Model Lifecycle

Problem framing, data, training, evaluation, deployment, monitoring and retraining.

The lifecycle covers problem framing, data collection, training, evaluation, deployment, monitoring and retraining. The PM owns framing, the success definition and the monitoring plan.

Summary of terms

Term	Meaning
Introduction to Artificial Intelligence	Systems that perform tasks associated with human intelligence, such as perception, language and decision making.
History and Evolution of AI	Rules and symbolic AI, expert systems, statistical ML, deep learning from 2012, transformers from 2017, and generative AI.
Traditional AI vs Generative AI	Traditional models classify or predict from data; generative models create new text, images, audio or code.
Narrow AI vs General AI	Narrow AI does specific tasks; general AI would match human breadth, and does not yet exist.
AI Applications Across Industries	Fraud detection, recommendations, diagnostics, document processing, support automation, forecasting and more.
Machine Learning Fundamentals	Algorithms that learn patterns from data to predict or decide without explicit rules.
Supervised Learning	Learning from labelled examples to predict labels for new data.

Unsupervised Learning	Finding structure in unlabelled data, such as clusters or anomalies.
Semi-Supervised Learning	Using a small labelled set plus a large unlabelled set.
Reinforcement Learning	An agent learns by trial and error to maximise cumulative reward.
Classification	Predicting a category, such as spam or not spam.
Regression	Predicting a continuous number, such as price or demand.
Clustering	Grouping similar items without labels, used for segmentation.
Recommendation Systems	Suggesting items using collaborative filtering, content features or hybrid models.
Predictive Analytics	Using historical data to forecast outcomes such as churn or demand.
Feature Engineering	Creating informative inputs from raw data to help models learn.
Training, Validation and Testing	Train to fit, validate to tune, and test once on unseen data for an honest estimate.
Model Training and Inference	Training learns parameters from data; inference uses the trained model to make predictions.
Overfitting and Underfitting	Overfitting memorises training noise and fails on new data; underfitting is too simple to capture the pattern.
Bias-Variance Trade-Off	Simple models have high bias; complex models have high variance; aim for the balance with best generalisation.
Model Performance Metrics	Accuracy, precision, recall, F1, AUC for classification; MAE and RMSE for regression, picked to match business cost of errors.
Precision, Recall and F1 Score	Precision is correctness of positive predictions; recall is coverage of actual positives; F1 balances both.
Confusion Matrix	A table of true/false positives and negatives that shows the types of errors.
ROC-AUC	Area under the curve of true-positive vs false-positive rate across thresholds; 0.5 is chance, 1.0 is perfect.
Deep Learning Fundamentals	Multi-layer neural networks that learn features automatically from large data.
Neural Networks	Layers of connected units with weights, trained by backpropagation and gradient descent.
CNN, RNN and LSTM	CNNs process images, RNNs process sequences, LSTMs handle longer dependencies in sequences.
Transformers	An architecture based on attention that processes sequences in parallel and underlies modern language models.
Attention Mechanisms	A way for a model to weigh which parts of the input matter for each output.
Natural Language Processing	Techniques for understanding and generating human language, from tokenisation to LLMs.
Computer Vision Fundamentals	Techniques for classifying, detecting and segmenting content in images and video.
AI Model Lifecycle	Problem framing, data, training, evaluation, deployment, monitoring and retraining.

Practice exercise: Train a simple classifier on a public dataset and report precision, recall and F1 at two thresholds. Say which one the business should pick.

Self-check

1. Explain Introduction to Artificial Intelligence in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Training, Validation and Testing in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain AI Model Lifecycle in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- Machine Learning Products for PMs : ML concepts, metrics and lifecycle.
- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Generative AI and Large Language Models

What generative models are, how LLMs work, and the practical limits that shape product decisions.

Chapter 12 of 26 · 30 topics · Edition 2026 · skillnavora.in

CHAPTER 12

Generative AI and Large Language Models

What generative models are, how LLMs work, and the practical limits that shape product decisions.

Introduction to Generative AI

Models that generate new content from learned patterns, across text, image, audio, video and code.

Generative AI models create new content such as text, images, audio, video and code, based on patterns learned from data. Its outputs are probabilistic, so the same prompt can give different answers.

Generative AI Architecture

The stack from foundation model to prompts, retrieval, tools, guardrails and application layer.

A typical stack includes the foundation model, prompts, retrieval of company data, tools the model can call, guardrails and the application layer that the user sees. Each layer can fail, so each needs monitoring.

Foundation Models

Large models pre-trained on broad data and adapted to many tasks.

Foundation models are large models pre-trained on broad data and adapted to many tasks. Products build on them through prompting, retrieval and fine-tuning rather than training from scratch.

Large Language Models (LLMs)

Transformer models trained to predict the next token, which gives them broad language ability.

LLMs are transformer models trained to predict the next token in text. That simple objective, at scale and with alignment training, gives broad language ability, including summarising and coding.

GPT Architecture

A decoder-only transformer trained to predict the next token, scaled up and then aligned with instructions and feedback.

GPT models are decoder-only transformers trained to predict the next token, then aligned with instructions and human feedback. The name refers to a family of OpenAI models; the architecture idea is shared widely.

Claude Model Ecosystem

Anthropic's model family across capability and cost tiers, with long context, tool use and strong coding and writing; check current model names in its documentation.

Anthropic offers a family of Claude models across capability and cost tiers, with long context windows and strong performance on writing and coding. Model names and prices change, so check the current documentation before choosing.

Gemini Model Ecosystem

Google's multimodal model family for text, image, audio, video and code, offered through the Gemini API and Vertex AI.

Google's Gemini models handle text, images, audio, video and code, and are available through the Gemini API and Vertex AI. Their multimodal inputs suit document and video workflows.

Open-Source LLMs

Models with downloadable weights, such as Llama, Mistral and Qwen, that you can host and tune, with licence terms to check.

Models with downloadable weights, such as Llama, Mistral and Qwen, can be hosted and tuned by the company. Check the licence, because some restrict commercial use or large-scale deployment.

Transformer Architecture

Stacked attention and feed-forward layers with embeddings and positional information.

A transformer stacks attention and feed-forward layers, with embeddings and positional information. It processes all tokens in parallel, which makes training at scale practical.

Tokens and Tokenisation

Text is split into sub-word tokens; cost, limits and speed are measured in tokens.

Text is split into sub-word tokens. Cost, context limits and speed are measured in tokens, so a 1,000-word document is roughly 1,300 to 1,500 tokens in English, and more in some other languages.

Context Windows

The maximum tokens a model can consider at once, covering prompt and response.

The context window is the maximum number of tokens the model can consider at once, covering the prompt and the response. Long documents may need retrieval rather than being pasted in full.

Embeddings

Numeric vectors that capture meaning so similar items sit close together.

Embeddings are numeric vectors that capture meaning, so similar texts sit close together. They power semantic search, recommendations and clustering.

Pre-Training

Initial large-scale training on broad data to learn language and world knowledge.

Pre-training trains the model on very large general data to learn language and world knowledge. It is expensive and done by model providers, not by most product teams.

Instruction Tuning

Training on instruction-response pairs so the model follows directions.

Instruction tuning trains the model on instruction and response pairs so it follows directions. It turns a text predictor into an assistant.

Fine-Tuning

Further training on task or domain data to change behaviour, style or accuracy.

Fine-tuning trains a model further on task or domain data to change behaviour, style or accuracy on a narrow task. It is often unnecessary when good prompting and retrieval work.

RLHF

Reinforcement learning from human feedback: human preferences train a reward signal to align outputs.

Reinforcement learning from human feedback trains a reward model on human preferences and optimises the model against it. It makes outputs more helpful and safer, but it can also make them overly cautious.

Reasoning Models

Models that spend extra compute thinking through steps before answering, improving hard tasks at higher cost and latency.

Reasoning models spend extra computation working through steps before answering. They improve on hard maths, planning and coding, but cost more and respond more slowly.

Multimodal AI

Models that handle several input or output types, such as text, images and audio.

Multimodal models accept or produce several types of data, such as text, images and audio. A product can let users photograph a receipt and ask a question about it.

Text Generation

Producing written content, summaries, answers, code and structured data.

Text generation produces summaries, drafts, answers, code and structured data. Quality depends on the prompt, the context and the evaluation applied to the output.

Image Generation

Creating images from prompts using diffusion or similar models.

Image models create pictures from prompts, often using diffusion methods. Commercial use depends on the provider's licence and on the rights of training data, so check terms.

Audio Generation

Generating speech, music or sound effects.

Audio models generate speech, music and sound effects. Voice cloning raises consent and misuse risks that the product must control.

Video Generation

Creating video clips from text or images; quality, length and cost are still constraints.

Video models create clips from text or images. Quality, length and cost remain constraints, so products usually limit duration and review outputs before publishing.

Speech-to-Text

Transcribing audio into text; accuracy depends on accents, noise and vocabulary.

Speech recognition converts audio to text. Accuracy depends on accents, background noise and domain vocabulary, so test with real recordings from your users.

Text-to-Speech

Converting text into natural-sounding voice with controllable style.

Text-to-speech generates natural voice from text, with control over style and pace. Disclose when users hear AI-generated voices.

Structured Outputs

Constraining model output to a schema such as JSON so software can use it reliably.

Structured outputs constrain the model to produce data in a schema such as JSON, so software can use the result reliably. Validate the output in code as well.

Function Calling

The model returns a structured request to call a named function, which your code runs.

The model returns a structured request to call a named function with arguments, and the application runs it. This lets models query databases or book appointments safely.

Model Selection and Benchmarking

Choosing a model by testing it on your own task data, not only public leaderboards.

Choose a model by testing it on your own tasks, measuring quality, latency and cost. Public leaderboards are a hint, not proof of fit for your use case.

Model Latency

Time to first token and total time; shaped by model size, output length and load.

Latency is shaped by model size, output length and load. Time to first token affects perceived speed, so streaming responses improves user experience.

Model Cost and Token Economics

Price per input and output token, multiplied by volume; shorter prompts, caching and smaller models cut cost.

Cost equals tokens processed multiplied by price per token. Shorter prompts, caching, smaller models for easy tasks and batching reduce cost. Track cost per successful task, not per call.

LLM Limitations and Hallucinations

Models can state false things confidently, lack current knowledge, and vary between runs; design for verification.

Models can state false information confidently, lack current knowledge and vary between runs. Design for verification: show sources, check high-stakes outputs and allow users to correct errors.

Summary of terms

Term	Meaning
Introduction to Generative AI	Models that generate new content from learned patterns, across text, image, audio, video and code.
Generative AI Architecture	The stack from foundation model to prompts, retrieval, tools, guardrails and application layer.
Foundation Models	Large models pre-trained on broad data and adapted to many tasks.
Large Language Models (LLMs)	Transformer models trained to predict the next token, which gives them broad language ability.
GPT Architecture	A decoder-only transformer trained to predict the next token, scaled up and then aligned with instructions and feedback.
Claude Model Ecosystem	Anthropic's model family across capability and cost tiers, with long context, tool use and strong coding and writing; check current model names in its documentation.
Gemini Model Ecosystem	Google's multimodal model family for text, image, audio, video and code, offered through the Gemini API and Vertex AI.
Open-Source LLMs	Models with downloadable weights, such as Llama, Mistral and Qwen, that you can host and tune, with licence terms to check.
Transformer Architecture	Stacked attention and feed-forward layers with embeddings and positional information.
Tokens and Tokenisation	Text is split into sub-word tokens; cost, limits and speed are measured in tokens.
Context Windows	The maximum tokens a model can consider at once, covering prompt and response.
Embeddings	Numeric vectors that capture meaning so similar items sit close together.
Pre-Training	Initial large-scale training on broad data to learn language and world knowledge.
Instruction Tuning	Training on instruction-response pairs so the model follows directions.
Fine-Tuning	Further training on task or domain data to change behaviour, style or accuracy.
RLHF	Reinforcement learning from human feedback: human preferences train a reward signal to align outputs.

Reasoning Models	Models that spend extra compute thinking through steps before answering, improving hard tasks at higher cost and latency.
Multimodal AI	Models that handle several input or output types, such as text, images and audio.
Text Generation	Producing written content, summaries, answers, code and structured data.
Image Generation	Creating images from prompts using diffusion or similar models.
Audio Generation	Generating speech, music or sound effects.
Video Generation	Creating video clips from text or images; quality, length and cost are still constraints.
Speech-to-Text	Transcribing audio into text; accuracy depends on accents, noise and vocabulary.
Text-to-Speech	Converting text into natural-sounding voice with controllable style.
Structured Outputs	Constraining model output to a schema such as JSON so software can use it reliably.
Function Calling	The model returns a structured request to call a named function, which your code runs.
Model Selection and Benchmarking	Choosing a model by testing it on your own task data, not only public leaderboards.
Model Latency	Time to first token and total time; shaped by model size, output length and load.
Model Cost and Token Economics	Price per input and output token, multiplied by volume; shorter prompts, caching and smaller models cut cost.
LLM Limitations and Hallucinations	Models can state false things confidently, lack current knowledge, and vary between runs; design for verification.

Practice exercise: Compare two models on twenty of your own tasks, noting quality, latency and cost per task.

Self-check

1. Explain Introduction to Generative AI in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain RLHF in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain LLM Limitations and Hallucinations in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- Machine Learning Products for PMs : ML concepts, metrics and lifecycle.
- 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.



AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Prompt Engineering and Context Engineering

Writing and managing the instructions and context that make model behaviour reliable in production.

Chapter 13 of 26 · 23 topics · Edition 2026 · skillnavora.in

CHAPTER 13

Prompt Engineering and Context Engineering

Writing and managing the instructions and context that make model behaviour reliable in production.

Prompt Engineering Fundamentals

Giving clear task, context, constraints and format so the model produces the output you need.

A good prompt states the task, context, constraints and output format. Clear instructions reduce variation and make outputs easier to evaluate.

Prompt Structure

Typical parts: role, task, context, examples, constraints and output format.

Typical parts are role, task, context, examples, constraints and output format. Keep the order consistent so teams can read and edit prompts quickly.

System Prompts

Persistent instructions that set role, rules and behaviour for a whole conversation.

A system prompt sets persistent role, rules and behaviour for a whole conversation or product. It is the first place to put product policy, tone and safety rules.

User Prompts

The per-turn request or input from the user.

The user prompt is the per-turn request. Treat its content as untrusted input, especially when it is combined with company data.

Role-Based Prompting

Assigning a persona or expertise to shape tone and focus.

Asking the model to act as an expert, such as a senior tax advisor, can shape tone and focus. It does not add knowledge the model lacks, so verify the content.

Zero-Shot Prompting

Asking for the task with no examples.

Zero-shot asks for the task with no examples. It is the fastest approach and works well for common tasks with clear instructions.

One-Shot Prompting

Providing a single example of the desired output.

One-shot gives a single example of the desired output. It helps the model match format or style.

Few-Shot Prompting

Providing several examples so the model infers the pattern.

Few-shot provides several examples so the model infers the pattern. Choose examples that cover typical and difficult cases.

Instruction-Based Prompting

Explicit step-by-step instructions and rules for the task.

Explicit step-by-step instructions and rules define the task. Numbered steps and clear do and do-not lists reduce ambiguity.

Chain-of-Thought Concepts

Asking the model to reason through steps, which can improve accuracy on multi-step problems.

Asking the model to reason through steps can improve accuracy on multi-step problems. For products, consider whether the reasoning should be shown to users or kept internal.

Prompt Chaining

Splitting a task into several prompts where each output feeds the next.

A task is split into several prompts where each output feeds the next, such as extract, then validate, then summarise. Chains are easier to test than one very long prompt.

Structured Prompting

Using sections, delimiters and schemas to make prompts unambiguous.

Sections, delimiters and schemas make prompts unambiguous. Use clear markers around user-provided text so the model does not confuse it with instructions.

JSON-Based Outputs

Requesting JSON with defined fields and validating it in code.

Requesting JSON with defined fields lets software use the result. Validate it in code and handle malformed output with a retry or a fallback.

Prompt Templates

Reusable prompts with placeholders filled at run time.

Templates are reusable prompts with placeholders filled at run time. They keep wording consistent across features and make changes easier to review.

Dynamic Prompt Variables

Inserting user, document or session data into templates safely.

Variables insert user, document or session data into a template. Escape or delimit inserted text, so it cannot change the instructions.

Context Engineering

Deciding exactly what information, tools and memory the model sees at each step.

Context engineering decides exactly what information, tools and memory the model sees at each step. It often matters more than the wording of the prompt.

Context Window Optimisation

Prioritising, summarising and trimming content to fit limits while keeping what matters.

Prioritise, summarise and trim content to fit the window while keeping what matters. Put the most important information where the model attends most reliably, usually near the instructions or the question.

Memory and Context Management

Storing and recalling conversation facts, summaries and user preferences across turns and sessions.

Memory stores conversation facts, summaries and preferences across turns and sessions. Ask users what should be remembered and let them delete it.

Prompt Versioning

Tracking prompt changes like code, with versions, owners and test results.

Track prompt changes like code, with versions, owners and test results. Without versioning, teams cannot tell which change caused a quality drop.

Prompt Evaluation

Scoring prompt versions against a fixed test set before release.

Score prompt versions against a fixed test set before release. Use the same inputs for every version so the comparison is fair.

Prompt Injection Awareness

Untrusted text can try to override instructions; treat external content as data, not commands.

Untrusted text, including content from web pages or documents, can try to override instructions. Treat external content as data, not commands, and limit what tools the model can use.

Prompt Guardrails

Input and output checks, refusals and policy rules that keep behaviour within bounds.

Input and output checks, refusals and policy rules keep behaviour within bounds. Guardrails are layers, so combine rules, classifiers and human review for high-risk cases.

Production Prompt Management

Central storage, review, rollout, monitoring and rollback of prompts used in live products.

Production prompts need central storage, review, staged rollout, monitoring and rollback. Treat a prompt change like a code change with the same release discipline.

Summary of terms

Term	Meaning
Prompt Engineering Fundamentals	Giving clear task, context, constraints and format so the model produces the output you need.
Prompt Structure	Typical parts: role, task, context, examples, constraints and output format.
System Prompts	Persistent instructions that set role, rules and behaviour for a whole conversation.
User Prompts	The per-turn request or input from the user.
Role-Based Prompting	Assigning a persona or expertise to shape tone and focus.
Zero-Shot Prompting	Asking for the task with no examples.
One-Shot Prompting	Providing a single example of the desired output.
Few-Shot Prompting	Providing several examples so the model infers the pattern.
Instruction-Based Prompting	Explicit step-by-step instructions and rules for the task.
Chain-of-Thought Concepts	Asking the model to reason through steps, which can improve accuracy on multi-step problems.
Prompt Chaining	Splitting a task into several prompts where each output feeds the next.
Structured Prompting	Using sections, delimiters and schemas to make prompts unambiguous.
JSON-Based Outputs	Requesting JSON with defined fields and validating it in code.
Prompt Templates	Reusable prompts with placeholders filled at run time.
Dynamic Prompt Variables	Inserting user, document or session data into templates safely.
Context Engineering	Deciding exactly what information, tools and memory the model sees at each step.
Context Window Optimisation	Prioritising, summarising and trimming content to fit limits while keeping what matters.
Memory and Context Management	Storing and recalling conversation facts, summaries and user preferences across turns and sessions.
Prompt Versioning	Tracking prompt changes like code, with versions, owners and test results.
Prompt Evaluation	Scoring prompt versions against a fixed test set before release.
Prompt Injection Awareness	Untrusted text can try to override instructions; treat external content as data, not commands.
Prompt Guardrails	Input and output checks, refusals and policy rules that keep behaviour within bounds.
Production Prompt Management	Central storage, review, rollout, monitoring and rollback of prompts used in live products.

Practice exercise: Write a system prompt with a role, rules and JSON output, then version it and test five variations on a fixed set of inputs.

Self-check

1. Explain Prompt Engineering Fundamentals in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Structured Prompting in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Production Prompt Management in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- Prompt and Context Engineering : 13 classes with templates.
- AI Skills Academy : Every AI skills course in one place.
- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Retrieval-Augmented Generation (RAG)

Grounding model answers in your own data: architecture, retrieval quality, evaluation and production operation.

Chapter 14 of 26 · 29 topics · Edition 2026 · skillnavora.in

CHAPTER 14

Retrieval-Augmented Generation (RAG)

Grounding model answers in your own data: architecture, retrieval quality, evaluation and production operation.

RAG Fundamentals

Retrieve relevant documents at request time and give them to the model to ground its answer.

Retrieval-augmented generation finds relevant documents at request time and gives them to the model so its answer is grounded in them. It lets a product answer from current company data without retraining the model.

Why RAG Is Required

Models lack your private or recent data and can hallucinate; RAG supplies sources and enables citations.

Models do not know your private or recent data and can hallucinate. RAG supplies sources, which enables citations and makes answers checkable.

RAG vs Fine-Tuning

RAG adds knowledge at query time and is easy to update; fine-tuning changes behaviour or style. They are often combined.

RAG adds knowledge at query time and is easy to update. Fine-tuning changes behaviour, style or format. Many products start with RAG and fine-tune only if behaviour still falls short.

RAG Architecture

Ingest and index, retrieve, rerank, build context, generate, cite, and monitor.

The pipeline ingests and indexes documents, retrieves candidates for a query, reranks them, builds the context, generates the answer with citations and is monitored for quality.

Data Ingestion

Collecting documents from sources and keeping the index in sync with changes.

Ingestion collects documents from sources such as drives, wikis and databases, then keeps the index in sync when content changes. Stale content is a common cause of wrong answers.

Document Parsing

Extracting clean text, tables and structure from PDFs, web pages and office files.

Parsing extracts clean text, tables and headings from PDFs, web pages and office files. Poor parsing silently degrades every later step, so check parsed output by hand.

Data Cleaning

Removing noise, duplicates and boilerplate and fixing encoding so retrieval is accurate.

Cleaning removes boilerplate, duplicates and noise, and fixes encoding problems. Clean data makes retrieval more accurate and smaller.

Document Chunking

Splitting documents into passages sized for retrieval and context limits.

Chunking splits documents into passages sized for retrieval and the context window. Chunks that are too small lose meaning; chunks that are too large dilute relevance.

Chunking Strategies

Fixed size, recursive, by heading or semantic boundaries, with overlap to preserve context.

Options include fixed-size chunks, recursive splitting, splitting by heading, and semantic boundaries, often with some overlap to keep context. Evaluate each option on your own questions.

Embedding Models

Models that convert text to vectors; choose by quality, language coverage, dimensions and cost.

Embedding models turn text into vectors. Compare them on retrieval quality for your language and domain, plus dimensions, cost and speed.

Vector Databases

Stores optimised for nearest-neighbour search over embeddings.

Vector databases store embeddings and run fast nearest-neighbour search. They also support metadata filters. Choose based on scale, hosting needs and integration.

Pinecone

A managed vector database service with simple scaling.

Pinecone is a managed vector database service with simple scaling. Teams that do not want to run infrastructure often choose a managed option.

Weaviate

An open-source vector database with hybrid search and modules.

Weaviate is an open-source vector database with hybrid search and modules for embedding. It can be self-hosted or used as a managed service.

Chroma

A lightweight open-source vector store popular for prototypes.

Chroma is a lightweight open-source vector store, popular for prototypes and small applications. Check its production readiness for your scale before committing.

FAISS

Meta's library for efficient similarity search over vectors, run in your own process.

FAISS is a library from Meta for efficient similarity search, run inside your own process. It is fast but does not provide a full database service by itself.

Semantic Search

Finding passages by meaning using embeddings, even without shared keywords.

Semantic search finds passages by meaning, even when they do not share keywords. It handles paraphrases well but can miss exact codes and names.

Keyword Search

Matching exact terms (BM25), strong for names, codes and rare words.

Keyword search, often BM25, matches exact terms. It is strong for names, product codes and rare words that embeddings can blur.

Hybrid Search

Combining keyword and semantic scores for better recall.

Hybrid search combines keyword and semantic scores. It usually improves recall over either method alone, at the cost of more tuning.

Metadata Filtering

Restricting retrieval by attributes such as date, source, permission or product.

Filters restrict retrieval by attributes such as date, source, product or user permission. Permission filters are essential so users only retrieve documents they may read.

Retrieval Strategies

Top-k, multi-query, parent-document, query rewriting and filters tuned to the content.

Options include top-k retrieval, multi-query retrieval, parent-document retrieval and query rewriting. Choose based on the failure patterns you see in evaluation.

Reranking

A second model reorders retrieved candidates by relevance before generation.

A reranker, a second model, reorders retrieved candidates by relevance before generation. It improves precision at the cost of extra latency.

Context Construction

Selecting, ordering and formatting retrieved passages inside the prompt.

Context construction selects, orders and formats retrieved passages inside the prompt, with source labels. Include only what supports the answer, since extra text can distract the model.

Grounded Generation

Instructing the model to answer only from provided sources and say when they are insufficient.

The model is instructed to answer only from the provided sources and to say when they are insufficient. Test that it does this on questions the documents do not cover.

Citation and Source Attribution

Linking each claim to its source so users can verify.

Each claim links to its source so users can verify it. Citations build trust, but they must point to the passage that actually supports the claim.

RAG Evaluation

Measuring retrieval and answer quality separately on a labelled question set.

Evaluate retrieval and answer quality separately on a labelled set of questions. A wrong answer may come from retrieval or from generation, and you need to know which.

Retrieval Precision and Recall

Precision is the share of retrieved passages that are relevant; recall is the share of relevant passages retrieved.

Precision is the share of retrieved passages that are relevant. Recall is the share of relevant passages that were retrieved. Both must be measured to tune chunking and ranking.

Faithfulness and Context Relevance

Faithfulness checks the answer is supported by the context; context relevance checks the context fits the question.

Faithfulness checks the answer is supported by the context. Context relevance checks the retrieved context fits the question. Both can be scored with a model-based judge validated against human labels.

RAG Performance Optimisation

Tuning chunking, embeddings, hybrid search, reranking, caching and prompts for quality, speed and cost.

Tune chunking, embeddings, hybrid search, reranking, caching and prompts for quality, speed and cost. Change one factor at a time and re-run the evaluation.

Production RAG Monitoring

Tracking retrieval quality, unanswered questions, latency, cost and freshness in live use.

Monitor retrieval quality, unanswered questions, latency, cost and document freshness in live use. Questions with no good source are a useful signal for content gaps.

Summary of terms

Term	Meaning

RAG Fundamentals	Retrieve relevant documents at request time and give them to the model to ground its answer.
Why RAG Is Required	Models lack your private or recent data and can hallucinate; RAG supplies sources and enables citations.
RAG vs Fine-Tuning	RAG adds knowledge at query time and is easy to update; fine-tuning changes behaviour or style. They are often combined.
RAG Architecture	Ingest and index, retrieve, rerank, build context, generate, cite, and monitor.
Data Ingestion	Collecting documents from sources and keeping the index in sync with changes.
Document Parsing	Extracting clean text, tables and structure from PDFs, web pages and office files.
Data Cleaning	Removing noise, duplicates and boilerplate and fixing encoding so retrieval is accurate.
Document Chunking	Splitting documents into passages sized for retrieval and context limits.
Chunking Strategies	Fixed size, recursive, by heading or semantic boundaries, with overlap to preserve context.
Embedding Models	Models that convert text to vectors; choose by quality, language coverage, dimensions and cost.
Vector Databases	Stores optimised for nearest-neighbour search over embeddings.
Pinecone	A managed vector database service with simple scaling.
Weaviate	An open-source vector database with hybrid search and modules.
Chroma	A lightweight open-source vector store popular for prototypes.
FAISS	Meta's library for efficient similarity search over vectors, run in your own process.
Semantic Search	Finding passages by meaning using embeddings, even without shared keywords.
Keyword Search	Matching exact terms (BM25), strong for names, codes and rare words.
Hybrid Search	Combining keyword and semantic scores for better recall.
Metadata Filtering	Restricting retrieval by attributes such as date, source, permission or product.
Retrieval Strategies	Top-k, multi-query, parent-document, query rewriting and filters tuned to the content.
Reranking	A second model reorders retrieved candidates by relevance before generation.
Context Construction	Selecting, ordering and formatting retrieved passages inside the prompt.
Grounded Generation	Instructing the model to answer only from provided sources and say when they are insufficient.
Citation and Source Attribution	Linking each claim to its source so users can verify.
RAG Evaluation	Measuring retrieval and answer quality separately on a labelled question set.
Retrieval Precision and Recall	Precision is the share of retrieved passages that are relevant; recall is the share of relevant passages retrieved.
Faithfulness and Context Relevance	Faithfulness checks the answer is supported by the context; context relevance checks the context fits the question.

RAG Performance Optimisation	Tuning chunking, embeddings, hybrid search, reranking, caching and prompts for quality, speed and cost.
Production RAG Monitoring	Tracking retrieval quality, unanswered questions, latency, cost and freshness in live use.

Practice exercise: Build a small RAG assistant over ten documents and measure retrieval precision and recall on twenty questions.

Self-check

1. Explain RAG Fundamentals in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain FAISS in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Production RAG Monitoring in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Agentic AI and Autonomous AI Systems

How AI agents plan, use tools and act, and how to run them safely and reliably in production.

Chapter 15 of 26 · 33 topics · Edition 2026 · skillnavora.in

CHAPTER 15

Agentic AI and Autonomous AI Systems

How AI agents plan, use tools and act, and how to run them safely and reliably in production.

Introduction to Agentic AI

Systems where a model pursues a goal over several steps by planning, calling tools and reacting to results.

An agent pursues a goal over several steps: it plans, calls tools, observes the results and decides the next step. Products differ by how much autonomy they give the agent.

AI Agents vs Chatbots

A chatbot replies to messages; an agent takes actions toward a goal.

A chatbot replies to messages. An agent takes actions toward a goal, such as creating a ticket and updating a customer record. Agents need permissions and oversight that chatbots do not.

AI Agents vs Traditional Automation

Automation follows fixed rules; agents handle variable inputs and decide the next step, with less predictability.

Automation follows fixed rules and fails on unexpected inputs. Agents handle variable inputs and choose the next step, but they are less predictable, so they need evaluation and limits.

Agent Architecture

A model, instructions, tools, memory, a control loop and guardrails.

An agent combines a model, instructions, tools, memory, a control loop and guardrails. The loop repeats: decide, act, observe, until the goal is met or a limit is reached.

Single-Agent Systems

One agent with a set of tools handles the task; simpler to build and debug.

One agent with a defined set of tools handles the task. Single agents are simpler to build, test and debug, so start here unless the task clearly separates into roles.

Multi-Agent Systems

Several specialised agents coordinate, useful for separable roles but costlier and harder to control.

Several specialised agents coordinate, for example a researcher, a writer and a checker. They suit separable roles, but add cost, latency and harder failure analysis.

Agent Planning

Deciding the sequence of steps to reach a goal, revising as results arrive.

Planning decides the sequence of steps toward a goal and revises it as results arrive. Plans should be inspectable so users and developers can see what the agent intended.

Reasoning and Task Decomposition

Breaking a goal into smaller sub-tasks the agent can complete and verify.

Breaking a goal into smaller sub-tasks the agent can complete and verify. Each sub-task should have a clear success check.

Tool Calling

The agent selects and invokes tools such as search, databases, code and APIs.

The agent selects and invokes tools such as search, databases, code execution and APIs. Each tool needs a clear description, input schema and error messages the agent can understand.

Function Calling

Structured model output naming a function and arguments, executed by your code.

The model returns a structured request naming a function and its arguments. Your code runs the function and returns the result to the model.

Agent Memory

Information kept so the agent can use past steps, facts and preferences.

Memory keeps facts, steps and preferences the agent needs later. Decide what is stored, for how long and who can see it.

Short-Term and Long-Term Memory

Short-term lives in the current context; long-term is stored externally and retrieved when needed.

Short-term memory lives in the current context. Long-term memory is stored externally and retrieved when needed, such as a user's saved preferences.

State Management

Tracking progress, inputs and results across steps so runs can resume and be audited.

State tracks progress, inputs and results across steps. It lets runs resume after failure and gives auditors a record of what happened.

Agent Orchestration

Coordinating agents, tools and steps, including routing, ordering and error paths.

Orchestration coordinates agents, tools and steps, including routing, ordering and error paths. It is the control layer that makes multi-step work reliable.

Workflow Orchestration

Defining fixed or conditional flows around model calls for predictable behaviour.

A workflow defines fixed or conditional steps around model calls. Workflows give predictable behaviour and are often the right choice before full autonomy.

Agent Communication

How agents exchange messages and shared state, including protocols such as MCP for tools and A2A for agents.

Agents exchange messages and shared state. Protocols such as MCP connect agents to tools and data, and agent-to-agent protocols connect agents to each other.

Human-in-the-Loop

Humans approve, edit or take over at defined checkpoints.

Humans approve, edit or take over at defined checkpoints. Place checkpoints before irreversible or costly actions, such as sending a payment or deleting data.

Autonomous Execution

Agents act without per-step approval, appropriate only for low-risk, reversible tasks.

The agent acts without per-step approval. This suits low-risk, reversible tasks with good evaluation, such as tagging support tickets.

Agent Permissions

Least-privilege access to tools and data, scoped per task and revocable.

Agents get least-privilege access to tools and data, scoped to each task and revocable. An agent with broad access is a security incident waiting to happen.

Agent Error Handling

Detecting tool failures, bad outputs and loops, and responding safely.

Handle tool failures, bad outputs and loops. Set timeouts, validate outputs and return clear errors to the agent so it can recover or stop safely.

Retry and Recovery Mechanisms

Retries with backoff, fallbacks, checkpoints and idempotent actions so failures do not cause harm.

Use retries with backoff, fallbacks, checkpoints and idempotent actions. Idempotent actions are safe to repeat, so a retry does not create a duplicate payment.

Agent Evaluation

Measuring task success, steps, cost and safety on realistic scenarios.

Measure task success, number of steps, cost and safety on realistic scenarios. Include tasks that should be refused, not only tasks that should succeed.

Agent Observability

Traces of every model call, tool call and decision for debugging and audit.

Traces record every model call, tool call and decision. Traces let engineers debug failures and let auditors reconstruct what an agent did.

Agent Security

Defending against prompt injection, data exfiltration, excessive permissions and malicious tools.

Defend against prompt injection through tool outputs, data exfiltration, excessive permissions and malicious tools. Treat every external input as untrusted.

Agent Governance

Policies for what agents may do, approval rules, logging, ownership and incident response.

Governance sets what agents may do, approval rules, logging, ownership and incident response. Assign a named owner to each production agent.

LangChain

A framework for building LLM applications with chains, tools and retrievers.

LangChain is a framework for LLM applications, with chains, tools and retrievers. It speeds up prototyping, but check its abstractions against your needs before adopting it widely.

LangGraph

A library for stateful, graph-based agent workflows with loops and human checkpoints.

LangGraph builds stateful, graph-based agent workflows with loops and human checkpoints. It suits agents that must follow a defined but branching process.

CrewAI

A framework for teams of role-based agents collaborating on tasks.

CrewAI is a framework for teams of role-based agents collaborating on tasks. It makes multi-agent designs quick to prototype, so test whether the roles add value.

LlamaIndex

A framework centred on data ingestion, indexing and retrieval for LLM apps.

LlamaIndex focuses on data ingestion, indexing and retrieval for LLM applications. It is a strong choice when the main challenge is connecting models to documents.

OpenAI Agents SDK

OpenAI's SDK for building agents with tools, handoffs and guardrails.

OpenAI's SDK builds agents with tools, handoffs between agents and guardrails. Handoffs let one agent pass the conversation to a specialist.

Microsoft Agent Framework

Microsoft's framework for building and orchestrating agents, evolved from Semantic Kernel and AutoGen.

Microsoft's framework builds and orchestrates agents, evolved from earlier Semantic Kernel and AutoGen work. It fits organisations already using Microsoft cloud services.

Google Agent Development Kit

Google's open-source kit for building and deploying agents, integrated with Gemini and Vertex AI.

Google's kit builds and deploys agents, integrated with Gemini and Vertex AI. It suits teams already on Google Cloud.

Production Agentic Systems

Reliable agents need evals, limits on steps and spend, observability, permissions, human review and rollback.

Reliable agents need evaluation, limits on steps and spend, observability, permissions, human review and rollback. A demo that works once is not a production agent.

Summary of terms

Term	Meaning
Introduction to Agentic AI	Systems where a model pursues a goal over several steps by planning, calling tools and reacting to results.
AI Agents vs Chatbots	A chatbot replies to messages; an agent takes actions toward a goal.
AI Agents vs Traditional Automation	Automation follows fixed rules; agents handle variable inputs and decide the next step, with less predictability.
Agent Architecture	A model, instructions, tools, memory, a control loop and guardrails.
Single-Agent Systems	One agent with a set of tools handles the task; simpler to build and debug.
Multi-Agent Systems	Several specialised agents coordinate, useful for separable roles but costlier and harder to control.
Agent Planning	Deciding the sequence of steps to reach a goal, revising as results arrive.
Reasoning and Task Decomposition	Breaking a goal into smaller sub-tasks the agent can complete and verify.
Tool Calling	The agent selects and invokes tools such as search, databases, code and APIs.
Function Calling	Structured model output naming a function and arguments, executed by your code.
Agent Memory	Information kept so the agent can use past steps, facts and preferences.
Short-Term and Long-Term Memory	Short-term lives in the current context; long-term is stored externally and retrieved when needed.
State Management	Tracking progress, inputs and results across steps so runs can resume and be audited.
Agent Orchestration	Coordinating agents, tools and steps, including routing, ordering and error paths.
Workflow Orchestration	Defining fixed or conditional flows around model calls for predictable behaviour.
Agent Communication	How agents exchange messages and shared state, including protocols such as MCP for tools and A2A for agents.
Human-in-the-Loop	Humans approve, edit or take over at defined checkpoints.
Autonomous Execution	Agents act without per-step approval, appropriate only for low-risk, reversible tasks.

Agent Permissions	Least-privilege access to tools and data, scoped per task and revocable.
Agent Error Handling	Detecting tool failures, bad outputs and loops, and responding safely.
Retry and Recovery Mechanisms	Retries with backoff, fallbacks, checkpoints and idempotent actions so failures do not cause harm.
Agent Evaluation	Measuring task success, steps, cost and safety on realistic scenarios.
Agent Observability	Traces of every model call, tool call and decision for debugging and audit.
Agent Security	Defending against prompt injection, data exfiltration, excessive permissions and malicious tools.
Agent Governance	Policies for what agents may do, approval rules, logging, ownership and incident response.
LangChain	A framework for building LLM applications with chains, tools and retrievers.
LangGraph	A library for stateful, graph-based agent workflows with loops and human checkpoints.
CrewAI	A framework for teams of role-based agents collaborating on tasks.
LlamaIndex	A framework centred on data ingestion, indexing and retrieval for LLM apps.
OpenAI Agents SDK	OpenAI's SDK for building agents with tools, handoffs and guardrails.
Microsoft Agent Framework	Microsoft's framework for building and orchestrating agents, evolved from Semantic Kernel and AutoGen.
Google Agent Development Kit	Google's open-source kit for building and deploying agents, integrated with Gemini and Vertex AI.
Production Agentic Systems	Reliable agents need evals, limits on steps and spend, observability, permissions, human review and rollback.

Practice exercise: Design an agent that books meetings. List its tools, permissions, approval points and what happens when a tool fails.

Self-check

1. Explain Introduction to Agentic AI in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Human-in-the-Loop in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Production Agentic Systems in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- How AI Agents Work : Single and multi-agent systems, context and evals.
- Agent Orchestration and Monitoring : Orchestrating and observing agents.
- AI Skills Academy : Every AI skills course in one place.
- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

AI Product Architecture and Development

How AI features are built, served and operated, and how a PM makes build, buy and partner decisions.

Chapter 16 of 26 · 26 topics · Edition 2026 · skillnavora.in

CHAPTER 16

AI Product Architecture and Development

How AI features are built, served and operated, and how a PM makes build, buy and partner decisions.

AI Product Architecture Fundamentals

The layers of an AI product: data, models, orchestration, application, monitoring and governance.

An AI product has layers: data, models, orchestration, application, monitoring and governance. A weakness in any layer limits the whole product, so review all layers, not only the model.

AI Application Architecture

How the interface, backend, model calls, data stores and tools fit together.

The interface, backend, model calls, data stores and tools fit together through defined APIs. Draw this picture early so engineers and PMs agree on boundaries.

LLM Application Architecture

Prompt assembly, retrieval, model call, tool execution, validation and response handling.

An LLM request usually assembles a prompt, retrieves context, calls the model, runs any tools, validates the output and returns the response. Each step adds latency and failure points.

Model Integration

Connecting models to the product through APIs or SDKs, with timeouts, retries and fallbacks.

Connect models through APIs or SDKs with timeouts, retries and fallbacks. Plan for provider outages, because a model dependency is an external service.

AI APIs

Hosted endpoints for models, speech, vision and embeddings, billed per use.

Hosted endpoints provide models for text, speech, vision and embeddings, billed per use. They remove infrastructure work but add cost per request and dependence on the provider.

Model Hosting

Running models on a provider's servers or your own infrastructure.

Hosting runs models on a provider's servers or your own infrastructure. Self-hosting gives control over data and cost at scale, but needs machine learning operations skills.

Model Serving

Systems that load models and answer requests efficiently with batching and scaling.

Serving systems load models and answer requests efficiently, using batching and scaling. Serving is where latency and cost targets are met or missed.

Inference Endpoints

Network endpoints that return predictions or generations on request.

Endpoints return predictions or generations on request. Monitor their latency, errors and cost separately from the rest of the application.

GPU Fundamentals

Parallel processors used to train and run models; memory size and availability drive cost and capacity.

GPUs run the parallel calculations behind training and inference. Memory size and availability drive both cost and capacity, so plan capacity with engineers.

AI Data Pipelines

Automated flows that collect, clean, label and deliver data for training, retrieval and monitoring.

Pipelines collect, clean, label and deliver data for training, retrieval and monitoring. Pipelines run continuously, so their failures can silently degrade quality.

ETL and ELT

Extract-transform-load transforms before loading; ELT loads first and transforms in the warehouse.

ETL transforms data before loading it into a store. ELT loads raw data first and transforms it in the warehouse. Modern analytics stacks often prefer ELT.

AI Application Databases

Operational stores for users, sessions, prompts, outputs and feedback.

Operational stores hold users, sessions, prompts, outputs and feedback. Feedback stored here feeds later evaluation and improvement.

Vector Database Integration

Embedding content, indexing it and querying by similarity within the application.

Embed content, index it and query by similarity inside the application. Keep the index in sync with source content and respect permissions at query time.

AI Orchestration Layers

Code or frameworks that sequence model calls, retrieval and tools.

Code or frameworks sequence model calls, retrieval and tools. A clear orchestration layer makes behaviour easier to test and change.

Microservices for AI

Separating model, retrieval and business services so each scales and deploys independently.

Separating model, retrieval and business services lets each scale and deploy independently. The cost is more network calls and more operational work.

Cloud AI Infrastructure

Managed platforms for training, hosting and monitoring models, such as Bedrock, Azure AI Foundry and Vertex AI.

Managed platforms for training, hosting and monitoring models include Amazon Bedrock, Azure AI Foundry and Google Vertex AI. They reduce operational work and tie you to a provider's services.

AI Application Scalability

Handling rising load with queues, caching, rate limits, batching and autoscaling.

Scale with queues, caching, rate limits, batching and autoscaling. Rate limits protect the product and the provider budget during traffic spikes.

Model Deployment

Releasing a model or prompt change to production with testing, staged rollout and rollback.

Release a model or prompt change to production with testing, staged rollout and rollback. Treat it like any software release with monitoring after each stage.

Model Versioning

Recording model, prompt and data versions so results are reproducible and regressions traceable.

Record model, prompt and data versions so results can be reproduced and regressions traced. Without versions, you cannot explain why behaviour changed.

MLOps

Practices for training, deploying, monitoring and retraining machine-learning models reliably.

MLOps covers training, deploying, monitoring and retraining machine learning models reliably. It applies software engineering discipline to models.

LLMOps

MLOps adapted to LLM apps: prompt and eval management, tracing, cost control and guardrails.

LLMOps adapts MLOps to LLM applications, adding prompt and evaluation management, tracing, cost control and guardrails.

CI/CD for AI

Automated tests and eval gates that run on every change to code, prompts or models.

Automated tests and evaluation gates run on every change to code, prompts or models. A change that lowers the quality score is blocked before release.

AI System Monitoring

Tracking quality, latency, errors, cost, drift and safety signals in production.

Track quality, latency, errors, cost, drift and safety signals in production. Set alerts on changes, not only on outages.

AI Application Reliability

Keeping the product working despite model outages and variance, via fallbacks, timeouts and degraded modes.

Keep the product working despite model outages and variable outputs, using fallbacks, timeouts and degraded modes. A degraded mode might show search results when the AI answer is unavailable.

Production AI Architecture

A complete design including security, observability, evaluation, cost controls and operational runbooks.

A production design includes security, observability, evaluation, cost controls and operational runbooks. Runbooks say what to do when something goes wrong at 2 a.m.

Build vs Buy vs Partner Decisions

Build where it differentiates you; buy for commodity capability; partner for speed or expertise. Weigh cost, control, time and risk.

Build where the capability differentiates you, buy commodity capability, and partner for speed or expertise. Weigh cost, control over data, time to market and risk.

Summary of terms

Term	Meaning
AI Product Architecture Fundamentals	The layers of an AI product: data, models, orchestration, application, monitoring and governance.
AI Application Architecture	How the interface, backend, model calls, data stores and tools fit together.
LLM Application Architecture	Prompt assembly, retrieval, model call, tool execution, validation and response handling.
Model Integration	Connecting models to the product through APIs or SDKs, with timeouts, retries and fallbacks.
AI APIs	Hosted endpoints for models, speech, vision and embeddings, billed per use.
Model Hosting	Running models on a provider's servers or your own infrastructure.
Model Serving	Systems that load models and answer requests efficiently with batching and scaling.
Inference Endpoints	Network endpoints that return predictions or generations on request.
GPU Fundamentals	Parallel processors used to train and run models; memory size and availability drive cost and capacity.
AI Data Pipelines	Automated flows that collect, clean, label and deliver data for training, retrieval and monitoring.
ETL and ELT	Extract-transform-load transforms before loading; ELT loads first and transforms in the warehouse.
AI Application Databases	Operational stores for users, sessions, prompts, outputs and feedback.
Vector Database Integration	Embedding content, indexing it and querying by similarity within the application.

AI Orchestration Layers	Code or frameworks that sequence model calls, retrieval and tools.
Microservices for AI	Separating model, retrieval and business services so each scales and deploys independently.
Cloud AI Infrastructure	Managed platforms for training, hosting and monitoring models, such as Bedrock, Azure AI Foundry and Vertex AI.
AI Application Scalability	Handling rising load with queues, caching, rate limits, batching and autoscaling.
Model Deployment	Releasing a model or prompt change to production with testing, staged rollout and rollback.
Model Versioning	Recording model, prompt and data versions so results are reproducible and regressions traceable.
MLOps	Practices for training, deploying, monitoring and retraining machine-learning models reliably.
LLMOps	MLOps adapted to LLM apps: prompt and eval management, tracing, cost control and guardrails.
CI/CD for AI	Automated tests and eval gates that run on every change to code, prompts or models.
AI System Monitoring	Tracking quality, latency, errors, cost, drift and safety signals in production.
AI Application Reliability	Keeping the product working despite model outages and variance, via fallbacks, timeouts and degraded modes.
Production AI Architecture	A complete design including security, observability, evaluation, cost controls and operational runbooks.
Build vs Buy vs Partner Decisions	Build where it differentiates you; buy for commodity capability; partner for speed or expertise. Weigh cost, control, time and risk.

Practice exercise: Draw the architecture of an AI feature with model, retrieval, database, monitoring and fallback. Write the build, buy or partner decision for each part.

Self-check

1. Explain AI Product Architecture Fundamentals in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain AI Orchestration Layers in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Build vs Buy vs Partner Decisions in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- System Design and Databases for PMs : Architecture, databases and trade-offs.
- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

AI Product Evaluation, Testing and Observability

How to know whether an AI product is good, safe and staying good over time.

Chapter 17 of 26 · 30 topics · Edition 2026 · skillnavora.in

CHAPTER 17

AI Product Evaluation, Testing and Observability

How to know whether an AI product is good, safe and staying good over time.

Traditional Software Testing vs AI Testing

Traditional tests expect exact outputs; AI outputs vary, so you score quality statistically on many cases.

Traditional tests expect exact outputs for given inputs. AI outputs vary, so quality is scored statistically across many cases, with a pass threshold set in advance.

AI Product Evaluation Fundamentals

Define what good means for users, build test sets, pick metrics and set a launch bar.

Define what good means to users, build representative test sets, choose metrics and set a launch bar. Without a definition of good, every release becomes an argument.

Offline Evaluation

Scoring a model or prompt on a fixed dataset before release.

Score a model or prompt on a fixed dataset before release. It is fast and repeatable, but it may not reflect real users.

Online Evaluation

Measuring real behaviour after release through experiments and user signals.

Measure real behaviour after release through experiments and user signals such as edits, retries and ratings. It shows the impact that matters, but carries user risk.

Human Evaluation

Reviewers rate outputs against a rubric; the gold standard but slow and costly.

Reviewers rate outputs against a rubric. It is the most trusted method but slow and expensive, so use it to calibrate automated scoring.

Automated Evaluation

Programmatic checks and model-based scoring that run quickly at scale.

Programmatic checks and model-based scoring run quickly at scale. Check the automated scores against human ratings regularly.

LLM-as-a-Judge

Using a model to grade outputs against a rubric; validate its agreement with human labels first.

A model grades outputs against a rubric. Validate its agreement with human labels first, and watch for bias toward longer or similar-sounding answers.

Golden Datasets

Curated, trusted examples with expected answers that anchor evaluation.

Curated, trusted examples with expected answers anchor evaluation. Keep them representative and update them when the product or users change.

Evaluation Datasets

Representative collections of inputs, including hard and adversarial cases, versioned and refreshed.

Representative collections of inputs, including hard and adversarial cases, are versioned and refreshed. A dataset that never changes stops finding new failures.

Benchmarking

Comparing models or versions on shared tasks; treat public benchmarks as a hint and test on your own data.

Benchmarking compares models or versions on shared tasks. Public benchmarks are a hint; test on your own data before deciding.

Accuracy and Relevance

Whether the answer is correct and addresses the question.

Accuracy asks whether the answer is correct. Relevance asks whether it addresses the question. An answer can be accurate but irrelevant.

Faithfulness

Whether claims are supported by the provided sources.

Faithfulness checks that claims are supported by the provided sources. It is the key quality check for grounded answers.

Hallucination Measurement

The rate of unsupported or false statements, tracked on a fixed test set.

Track the rate of unsupported or false statements on a fixed test set. Measure it on every model or prompt change.

Retrieval Evaluation

Checking whether the right passages are found, using precision, recall and ranking metrics.

Check whether the right passages are found, using precision, recall and ranking measures. Retrieval errors look like generation errors unless measured separately.

Agent Success Rate

The share of tasks an agent completes correctly end to end.

The share of tasks an agent completes correctly end to end. Break failures down by step to see where the agent struggles.

Task Completion Rate

The share of user tasks finished, with or without assistance.

The share of user tasks finished, with or without help. A low rate with high satisfaction can mean the task is too hard to finish.

Safety Evaluation

Testing for harmful, unsafe or policy-violating outputs.

Test for harmful, unsafe or policy-violating outputs using adversarial and everyday inputs. Safety tests should run before every release.

Toxicity and Bias Testing

Measuring offensive output and unequal treatment across groups.

Measure offensive output and unequal treatment across groups, using paired test inputs that differ only in the group mentioned.

Robustness Testing

Checking behaviour under typos, paraphrases, noise and adversarial inputs.

Check behaviour under typos, paraphrases, noisy inputs and adversarial prompts. A product that works only on perfectly written questions will fail in use.

Regression Testing

Re-running the evaluation suite after every change to catch quality drops.

Re-run the evaluation suite after every change to catch quality drops. Regressions are often silent until a user reports them.

AI Red Teaming

Deliberately attacking the system to find failures before attackers or users do.

Deliberately attack the system to find failures before attackers or users do. Include prompt injection, data extraction and misuse scenarios.

A/B Testing for AI

Comparing model or prompt variants on users, with quality, cost and safety guardrails.

Compare model or prompt variants on real users with quality, cost and safety guardrails. Keep the metrics defined before the test starts.

Model Latency

Response time measured at median and tail percentiles against a target.

Measure response time at median and tail percentiles against a target. Tail latency (the slowest 5 to 1 per cent) often drives user complaints.

Throughput

Requests handled per second under load.

Throughput is requests handled per second under load. Load test before launch so you know where the system slows down.

Cost per Request

Total model and infrastructure cost divided by requests; track per successful task as well.

Total model and infrastructure cost divided by requests. Also track cost per successful task, which is the number that shows real efficiency.

Token Consumption

Input and output tokens used, a driver of cost and latency.

Input and output tokens drive cost and latency. Monitor token use per feature to find prompts that grew without anyone noticing.

AI Observability

Logs, traces and metrics that explain what the system did and why.

Logs, traces and metrics explain what the system did and why. Observability turns a vague complaint into a specific failing step.

Production Monitoring

Dashboards and alerts on quality, errors, latency, cost and safety in live traffic.

Dashboards and alerts track quality, errors, latency, cost and safety in live traffic. Alerts should point to an owner and a runbook.

Model Drift

Performance changes as data, users or models change over time.

Performance changes as data, user behaviour or the model itself changes. Compare recent inputs with the data the system was tested on to detect drift early.

Continuous Evaluation

Ongoing scoring of live samples and test suites so quality is tracked, not assumed.

Score live samples and run the test suite continuously, so quality is tracked, not assumed. Quality can fall slowly, and continuous scoring is the only reliable way to notice.

Summary of terms

Term	Meaning
Traditional Software Testing vs AI Testing	Traditional tests expect exact outputs; AI outputs vary, so you score quality statistically on many cases.
AI Product Evaluation Fundamentals	Define what good means for users, build test sets, pick metrics and set a launch bar.
Offline Evaluation	Scoring a model or prompt on a fixed dataset before release.
Online Evaluation	Measuring real behaviour after release through experiments and user signals.
Human Evaluation	Reviewers rate outputs against a rubric; the gold standard but slow and costly.
Automated Evaluation	Programmatic checks and model-based scoring that run quickly at scale.
LLM-as-a-Judge	Using a model to grade outputs against a rubric; validate its agreement with human labels first.
Golden Datasets	Curated, trusted examples with expected answers that anchor evaluation.
Evaluation Datasets	Representative collections of inputs, including hard and adversarial cases, versioned and refreshed.
Benchmarking	Comparing models or versions on shared tasks; treat public benchmarks as a hint and test on your own data.
Accuracy and Relevance	Whether the answer is correct and addresses the question.
Faithfulness	Whether claims are supported by the provided sources.
Hallucination Measurement	The rate of unsupported or false statements, tracked on a fixed test set.
Retrieval Evaluation	Checking whether the right passages are found, using precision, recall and ranking metrics.
Agent Success Rate	The share of tasks an agent completes correctly end to end.
Task Completion Rate	The share of user tasks finished, with or without assistance.
Safety Evaluation	Testing for harmful, unsafe or policy-violating outputs.
Toxicity and Bias Testing	Measuring offensive output and unequal treatment across groups.
Robustness Testing	Checking behaviour under typos, paraphrases, noise and adversarial inputs.
Regression Testing	Re-running the evaluation suite after every change to catch quality drops.
AI Red Teaming	Deliberately attacking the system to find failures before attackers or users do.
A/B Testing for AI	Comparing model or prompt variants on users, with quality, cost and safety guardrails.
Model Latency	Response time measured at median and tail percentiles against a target.
Throughput	Requests handled per second under load.
Cost per Request	Total model and infrastructure cost divided by requests; track per successful task as well.
Token Consumption	Input and output tokens used, a driver of cost and latency.
AI Observability	Logs, traces and metrics that explain what the system did and why.
Production Monitoring	Dashboards and alerts on quality, errors, latency, cost and safety in live traffic.
Model Drift	Performance changes as data, users or models change over time.

Continuous Evaluation	Ongoing scoring of live samples and test suites so quality is tracked, not assumed.

Practice exercise: Create a golden set of fifty cases for one AI feature, define a rubric, and score two prompt versions.

Self-check

1. Explain Traditional Software Testing vs AI Testing in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Task Completion Rate in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Continuous Evaluation in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- Agent Orchestration and Monitoring : Orchestrating and observing agents.
- AI Evals for Product Managers : Evaluation methods and datasets.
- AI Skills Academy : Every AI skills course in one place.
- 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.



Responsible AI, Safety and Governance

Managing the ethical, legal and security risks of AI products so they can be launched and trusted.

Chapter 18 of 26 · 24 topics · Edition 2026 · skillnavora.in

CHAPTER 18

Responsible AI, Safety and Governance

Managing the ethical, legal and security risks of AI products so they can be launched and trusted.

Responsible AI Fundamentals

Principles of fairness, reliability, safety, privacy, transparency, accountability and inclusiveness.

Responsible AI principles include fairness, reliability, safety, privacy, transparency, accountability and inclusiveness. Turn each principle into a testable requirement for your product.

AI Ethics

Weighing benefits and harms to people and society when designing and deploying AI.

Ethics weighs the benefits and harms to people and society when designing and deploying AI. Ask who could be harmed and who has not been consulted.

Data Privacy

Respecting what personal data is collected, how it is used and who sees it.

Respect what personal data is collected, how it is used and who can see it. Collect only what the feature needs and explain it clearly to users.

Data Protection

Technical and organisational safeguards such as encryption, access control, retention limits and minimisation.

Safeguards include encryption, access control, retention limits and data minimisation. Retention limits mean deleting data when it is no longer needed.

GDPR

The EU regulation giving people rights over personal data and imposing duties on organisations processing it.

The EU regulation gives people rights over their personal data and places duties on organisations that process it. It applies to products serving EU residents, even when the company is based elsewhere.

India's Digital Personal Data Protection Act

India's 2023 law on processing digital personal data, based on consent and notice, with obligations for data fiduciaries; confirm current rules with counsel.

India's 2023 law regulates digital personal data, based on consent, notice and obligations for data fiduciaries. Check current rules and their implementation timeline with legal counsel.

AI Security

Protecting models, data, prompts and tools from misuse and attack.

Protect models, data, prompts and tools from misuse and attack. Security reviews should cover the model provider, the data flow and the tools the model can call.

Data Leakage

Sensitive information exposed through prompts, outputs, logs or training data.

Sensitive information can leak through prompts, outputs, logs or training data. Redact personal data before it reaches a model or a log, when the feature does not need it.

Prompt Injection

Malicious text that tries to override instructions or trigger unwanted actions, including through retrieved content.

Malicious text tries to override instructions or trigger unwanted actions, including text hidden in retrieved documents. Separate instructions from data and limit the tools available to the model.

Jailbreak Risks

Prompts designed to bypass safety rules; test regularly and layer defences.

Jailbreak prompts try to bypass safety rules. Test regularly with known patterns, and layer defences rather than relying on one filter.

AI Bias

Systematic unfairness from skewed data, labels or design.

Bias is systematic unfairness from skewed data, labels or design choices. Audit outcomes across groups, not only overall accuracy.

Fairness

Ensuring outcomes are equitable across groups, measured with agreed metrics.

Fairness means outcomes are equitable across groups, measured with agreed metrics. There is no single fairness metric, so choose the one that matches the harm you are guarding against.

Explainability

Giving understandable reasons for outputs, especially for high-stakes decisions.

Explainability gives understandable reasons for outputs, especially for high-stakes decisions such as credit. Users and regulators may need the reasons, not only the score.

Transparency

Being clear about when AI is used, its limits and how data is handled.

Be clear about when AI is used, its limits and how data is handled. Users calibrate trust better when they know what the system is.

Human Oversight

Meaningful human review, override and accountability for important decisions.

Meaningful human review, override and accountability for important decisions. A human who rubber-stamps every output does not provide real oversight.

AI Guardrails

Input and output filters, policy checks and permission limits that constrain behaviour.

Input and output filters, policy checks and permission limits constrain behaviour. Guardrails work best as several layers rather than one rule.

Content Moderation

Detecting and handling harmful or disallowed content from users and models.

Detect and handle harmful or disallowed content from users and models. Define the policy first, then build tools to enforce it, with an appeals path.

Access Control

Limiting who and what can reach data, tools and models, based on role and need.

Limit who and what can reach data, tools and models, based on role and need. Apply the same rules to the AI feature as to the underlying data.

AI Governance

Policies, roles, review processes and standards for responsible AI across the organisation.

Governance sets policies, roles, review processes and standards for responsible AI across the organisation. It answers who approves an AI use case and who is accountable when it fails.

AI Risk Management

Identifying, rating and mitigating AI risks, using frameworks such as NIST AI RMF.

Identify, rate and mitigate AI risks. Frameworks such as the NIST AI Risk Management Framework give a structure for this work.

Enterprise AI Governance

Central oversight of AI use cases, vendors, data and approvals across a large organisation.

Central oversight of AI use cases, vendors, data and approvals in large organisations. It prevents teams from adopting unapproved tools with customer data.

Regulatory Compliance

Meeting laws and standards that apply to your market and use case, such as the EU AI Act, GDPR and sector rules.

Meet the laws and standards that apply to your market and use case, such as the EU AI Act, GDPR and sector rules. Compliance requirements shape product design, so involve legal early.

AI Incident Management

Detecting, triaging, fixing and learning from AI failures, with communication and post-incident review.

Detect, triage, fix and learn from AI failures, with clear communication and a post-incident review. Blameless reviews find systemic causes.

Responsible AI Product Launch

A checklist before launch covering risks, testing, approvals, user disclosure, monitoring and rollback.

A pre-launch checklist covers risks, testing, approvals, user disclosure, monitoring and rollback. Launch only when every item has an owner and a status.

Summary of terms

Term	Meaning
Responsible AI Fundamentals	Principles of fairness, reliability, safety, privacy, transparency, accountability and inclusiveness.
AI Ethics	Weighing benefits and harms to people and society when designing and deploying AI.
Data Privacy	Respecting what personal data is collected, how it is used and who sees it.
Data Protection	Technical and organisational safeguards such as encryption, access control, retention limits and minimisation.
GDPR	The EU regulation giving people rights over personal data and imposing duties on organisations processing it.
India's Digital Personal Data Protection Act	India's 2023 law on processing digital personal data, based on consent and notice, with obligations for data fiduciaries; confirm current rules with counsel.
AI Security	Protecting models, data, prompts and tools from misuse and attack.
Data Leakage	Sensitive information exposed through prompts, outputs, logs or training data.
Prompt Injection	Malicious text that tries to override instructions or trigger unwanted actions, including through retrieved content.
Jailbreak Risks	Prompts designed to bypass safety rules; test regularly and layer defences.
AI Bias	Systematic unfairness from skewed data, labels or design.
Fairness	Ensuring outcomes are equitable across groups, measured with agreed metrics.
Explainability	Giving understandable reasons for outputs, especially for high-stakes decisions.
Transparency	Being clear about when AI is used, its limits and how data is handled.
Human Oversight	Meaningful human review, override and accountability for important decisions.
AI Guardrails	Input and output filters, policy checks and permission limits that constrain behaviour.
Content Moderation	Detecting and handling harmful or disallowed content from users and models.
Access Control	Limiting who and what can reach data, tools and models, based on role and need.

AI Governance	Policies, roles, review processes and standards for responsible AI across the organisation.
AI Risk Management	Identifying, rating and mitigating AI risks, using frameworks such as NIST AI RMF.
Enterprise AI Governance	Central oversight of AI use cases, vendors, data and approvals across a large organisation.
Regulatory Compliance	Meeting laws and standards that apply to your market and use case, such as the EU AI Act, GDPR and sector rules.
AI Incident Management	Detecting, triaging, fixing and learning from AI failures, with communication and post-incident review.
Responsible AI Product Launch	A checklist before launch covering risks, testing, approvals, user disclosure, monitoring and rollback.

Practice exercise: Run a risk review for one AI feature covering privacy, bias, injection and misuse. Assign an owner and mitigation to each risk.

Self-check

1. Explain Responsible AI Fundamentals in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Explainability in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Responsible AI Product Launch in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- AI Governance and Responsible AI : Risk, safety and compliance.
- 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.



AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

AI Product UX and Human-AI Interaction

Designing AI experiences people trust, understand and can correct.

Chapter 19 of 26 · 17 topics · Edition 2026 · skillnavora.in

CHAPTER 19

AI Product UX and Human-AI Interaction

Designing AI experiences people trust, understand and can correct.

AI-First Product Design

Designing around what AI can do well, with fallbacks for what it cannot.

Design around what AI does well, with clear fallbacks for what it cannot do. Decide early which parts of the job the AI handles and which the user still controls.

Conversational UX

Natural dialogue patterns with clear scope, suggestions and recovery from misunderstanding.

Natural dialogue with clear scope, suggested prompts and recovery from misunderstanding. Users should know what the assistant can and cannot do.

AI Chat Interfaces

Layouts and controls for chat: streaming, history, attachments, citations and actions.

Chat layouts need streaming responses, history, attachments, citations and actions the user can take on an answer. Make it easy to copy, edit or regenerate.

Copilot Experiences

AI embedded in a workflow that suggests and assists while the user stays in charge.

AI embedded in a workflow suggests and assists while the user stays in charge. The copilot should appear where the work happens, not in a separate tab.

AI-Assisted Workflows

Steps in a task where AI drafts, classifies or automates, with review points.

Steps in a task where AI drafts, classifies or automates, with review points. Show what the AI did so the reviewer can check it quickly.

Human-AI Collaboration

Dividing work so each does what they do best, with clear hand-offs.

Divide work so each side does what it does best, with clear hand-offs. People set goals and judge; AI drafts, searches and summarises.

Trust and Transparency

Showing sources, reasoning summaries and limits so users calibrate trust.

Show sources, reasoning summaries and limits so users calibrate trust. Trust should match accuracy: too much trust is as harmful as too little.

Communicating AI Uncertainty

Conveying confidence honestly through wording, indicators and prompts to verify.

Convey confidence honestly through wording, indicators and prompts to verify. Avoid false precision such as "97.3% sure" unless the number is calibrated.

AI Error Recovery

Letting users undo, edit, regenerate and report wrong outputs easily.

Let users undo, edit, regenerate and report wrong outputs easily. A visible undo path makes users more willing to try the feature.

Human Approval Workflows

Requiring user confirmation before consequential actions.

Require user confirmation before consequential actions such as sending messages, spending money or deleting records. Show exactly what will happen before the user confirms.

AI Personalisation

Adapting content and behaviour to the user, with consent and control.

Adapt content and behaviour to the user, with consent and control. Explain why a recommendation appears and let users adjust it.

Multimodal UX

Combining text, voice, image and video input and output coherently.

Combine text, voice, image and video input and output coherently. Test each mode separately and together, since errors often appear at the boundaries.

AI Onboarding

Setting expectations, showing example prompts and guiding first success.

Set expectations, show example prompts and guide the first success. Users who get a good first result are far more likely to return.

AI Feature Adoption

Helping users discover, try and keep using AI features, measured by repeat use.

Help users discover, try and keep using AI features, measured by repeat use rather than first clicks. Repeat use shows whether the feature is useful.

AI Accessibility

Making AI experiences usable with assistive technology, including voice, captions and plain language.

Make AI experiences usable with assistive technology, including voice, captions and plain language. Streaming text needs care so screen readers are not overwhelmed.

Feedback Loops

Collecting ratings, edits and corrections and feeding them into evaluation and improvement.

Collect ratings, edits and corrections, and feed them into evaluation and improvement. Edits the user makes are some of the most valuable signals you will get.

AI User Experience Evaluation

Testing usefulness, trust, effort and satisfaction with real users and metrics.

Test usefulness, trust, effort and satisfaction with real users and metrics. Combine task success with qualitative interviews to learn why users trust or reject the output.

Summary of terms

Term	Meaning
AI-First Product Design	Designing around what AI can do well, with fallbacks for what it cannot.
Conversational UX	Natural dialogue patterns with clear scope, suggestions and recovery from misunderstanding.
AI Chat Interfaces	Layouts and controls for chat: streaming, history, attachments, citations and actions.
Copilot Experiences	AI embedded in a workflow that suggests and assists while the user stays in charge.
AI-Assisted Workflows	Steps in a task where AI drafts, classifies or automates, with review points.
Human-AI Collaboration	Dividing work so each does what they do best, with clear hand-offs.
Trust and Transparency	Showing sources, reasoning summaries and limits so users calibrate trust.
Communicating AI Uncertainty	Conveying confidence honestly through wording, indicators and prompts to verify.
AI Error Recovery	Letting users undo, edit, regenerate and report wrong outputs easily.
Human Approval Workflows	Requiring user confirmation before consequential actions.
AI Personalisation	Adapting content and behaviour to the user, with consent and control.
Multimodal UX	Combining text, voice, image and video input and output coherently.
AI Onboarding	Setting expectations, showing example prompts and guiding first success.
AI Feature Adoption	Helping users discover, try and keep using AI features, measured by repeat use.
AI Accessibility	Making AI experiences usable with assistive technology, including voice, captions and plain language.
Feedback Loops	Collecting ratings, edits and corrections and feeding them into evaluation and improvement.
AI User Experience Evaluation	Testing usefulness, trust, effort and satisfaction with real users and metrics.

Practice exercise: Redesign one AI screen so it shows sources, admits uncertainty and lets the user undo and correct the output.

Self-check

1. Explain AI-First Product Design in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain AI Error Recovery in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain AI User Experience Evaluation in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- Designing AI Experiences (AI UX) : Trust, uncertainty and human-AI interaction.
- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

AI Product Strategy and Monetisation

Finding AI opportunities, deciding how to build them, and making them commercially sound.

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



CHAPTER 20

AI Product Strategy and Monetisation

Finding AI opportunities, deciding how to build them, and making them commercially sound.

Identifying AI Product Opportunities

Look for frequent, costly tasks involving language, patterns or judgement where partial automation helps.

Look for frequent, costly tasks involving language, patterns or judgement where partial automation helps. The best opportunities also have data available and a clear person who gains.

AI Opportunity Assessment

Scoring a use case on value, feasibility, data readiness, risk and cost before committing.

Score a use case on value, feasibility, data readiness, risk and cost before committing. Decide quickly which opportunities need a prototype and which should be dropped.

AI Product-Market Fit

Evidence that users rely on the AI feature for real outcomes, seen in repeat use and willingness to pay.

Evidence that users rely on the AI feature for real outcomes: repeat use, measurable time saved and willingness to pay. Novelty use fades; reliance does not.

Build vs Buy vs Partner

Decide by differentiation, speed, cost, control and data sensitivity.

Decide by differentiation, speed, cost, control and data sensitivity. Build what makes you different, buy commodity capability and partner when you need speed or expertise.

AI Business Models

Seat-based, usage-based, outcome-based, add-on and platform models, each with different margins and risks.

Seat-based, usage-based, outcome-based, add-on and platform models each have different margins and risks. Usage-based models protect margin when costs scale, but they make revenue less predictable.

AI Product Pricing

Price to value delivered while covering variable inference cost, with tiers and limits.

Price to the value delivered while covering variable inference cost. Tiers and usage limits keep heavy users from eroding margin.

Usage-Based Pricing

Charging by tokens, requests, tasks or outcomes, matching revenue to cost.

Charge by tokens, requests, tasks or outcomes, so revenue follows cost. Customers need a clear way to estimate and control their spend.

Subscription Monetisation

Fixed plans with usage allowances, simple to buy but sensitive to heavy users.

Fixed plans with usage allowances are simple to buy but sensitive to heavy users. Monitor cost per plan so a small group of heavy users does not make a plan unprofitable.

Enterprise AI Monetisation

Contracts, platform fees, integrations and services for large customers, often with security reviews.

Large customers pay through contracts, platform fees, integrations and services, often after security reviews. Price in the cost of those reviews and of custom integration.

AI Unit Economics

Margin per user or task after inference, data, support and infrastructure costs.

Margin per user or per task after inference, data, support and infrastructure costs. Know the margin before scaling, not after.

Inference Cost Management

Reducing cost via smaller models, caching, shorter prompts, batching and routing.

Reduce cost through smaller models for simple tasks, caching, shorter prompts, batching and routing. Routing sends easy requests to cheap models and hard ones to strong models.

Token Economics

Understanding price per million tokens and how prompt design and volume change cost.

Understand the price per million tokens and how prompt design and volume change cost. Long system prompts sent with every request can dominate cost.

AI ROI Calculation

Benefit (time saved, revenue gained, cost avoided) minus total cost, over the cost.

ROI is the benefit (time saved, revenue gained or cost avoided) minus total cost, divided by total cost. Measure the benefit against a baseline, not against a guess.

Total Cost of Ownership

All costs over time: models, infrastructure, data, people, governance and maintenance.

Total cost includes models, infrastructure, data, people, governance and maintenance over the life of the product. Many AI projects underestimate maintenance, especially evaluation.

AI Product Differentiation

Advantages from proprietary data, workflow integration, evaluation quality, distribution and trust.

Advantages come from proprietary data, workflow integration, evaluation quality, distribution and trust. A better model alone is a weak advantage because competitors can access the same models.

AI Competitive Positioning

Defining where you win against general-purpose assistants and rival tools.

Define where you win against general-purpose assistants and rival tools. Be specific about the job, the user and the workflow you serve better.

AI Product GTM

Positioning, pricing, proof points, pilots and enablement for launching AI products.

Positioning, pricing, proof points, pilots and enablement for launching AI products. Enterprise buyers often want a pilot with measurable results before a contract.

Enterprise AI Adoption

Moving from pilot to production through change management, integration and governance.

Moving from pilot to production through change management, integration and governance. Most AI pilots fail at adoption, not at the model.

AI Customer Retention

Keeping customers through reliability, improving quality, integration and shown outcomes.

Keep customers through reliability, improving quality, integration and shown outcomes. Regular reports on results help customers see the value.

AI Product Scaling

Growing usage while controlling cost, quality, latency and risk.

Grow usage while controlling cost, quality, latency and risk. Scale only after unit economics and quality are stable.

Summary of terms

Term	Meaning
Identifying AI Product Opportunities	Look for frequent, costly tasks involving language, patterns or judgement where partial automation helps.
AI Opportunity Assessment	Scoring a use case on value, feasibility, data readiness, risk and cost before committing.
AI Product-Market Fit	Evidence that users rely on the AI feature for real outcomes, seen in repeat use and willingness to pay.
Build vs Buy vs Partner	Decide by differentiation, speed, cost, control and data sensitivity.

AI Business Models	Seat-based, usage-based, outcome-based, add-on and platform models, each with different margins and risks.
AI Product Pricing	Price to value delivered while covering variable inference cost, with tiers and limits.
Usage-Based Pricing	Charging by tokens, requests, tasks or outcomes, matching revenue to cost.
Subscription Monetisation	Fixed plans with usage allowances, simple to buy but sensitive to heavy users.
Enterprise AI Monetisation	Contracts, platform fees, integrations and services for large customers, often with security reviews.
AI Unit Economics	Margin per user or task after inference, data, support and infrastructure costs.
Inference Cost Management	Reducing cost via smaller models, caching, shorter prompts, batching and routing.
Token Economics	Understanding price per million tokens and how prompt design and volume change cost.
AI ROI Calculation	Benefit (time saved, revenue gained, cost avoided) minus total cost, over the cost.
Total Cost of Ownership	All costs over time: models, infrastructure, data, people, governance and maintenance.
AI Product Differentiation	Advantages from proprietary data, workflow integration, evaluation quality, distribution and trust.
AI Competitive Positioning	Defining where you win against general-purpose assistants and rival tools.
AI Product GTM	Positioning, pricing, proof points, pilots and enablement for launching AI products.
Enterprise AI Adoption	Moving from pilot to production through change management, integration and governance.
AI Customer Retention	Keeping customers through reliability, improving quality, integration and shown outcomes.
AI Product Scaling	Growing usage while controlling cost, quality, latency and risk.

Practice exercise: Estimate unit economics for an AI feature: tokens per task, cost per task, price, margin and the usage level where it stops being profitable.

Self-check

1. Explain Identifying AI Product Opportunities in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Inference Cost Management in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain AI Product Scaling in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- AI Product Strategy : Opportunities, positioning and strategy.
- Pricing and Monetising AI Products : Pricing models and unit economics.
- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

AI Product Tools and Ecosystem

The tools a modern AI PM should recognise and know when to use. Tool features change quickly, so always check the official site.

Chapter 21 of 26 · 32 topics · Edition 2026 · skillnavora.in



CHAPTER 21

AI Product Tools and Ecosystem

The tools a modern AI PM should recognise and know when to use. Tool features change quickly, so always check the official site.

ChatGPT

OpenAI's assistant for writing, analysis, coding and research, with custom GPTs and tools.

OpenAI's assistant for writing, analysis, coding and research, with custom GPTs and tool integrations. Use it to draft, summarise and test ideas; check current features on the official site.

Claude

Anthropic's assistant and API, known for long documents, writing and coding, with projects and tool use.

Anthropic's assistant and API, known for long documents, writing and coding, with projects and tool use. Use it for reviewing long specifications and drafting PRDs.

Gemini

Google's multimodal assistant and models, integrated with Workspace and Google Cloud.

Google's multimodal assistant and models, integrated with Workspace and Google Cloud. Useful when your documents live in Google Drive.

Perplexity

An answer engine that searches the web and cites sources.

An answer engine that searches the web and cites sources. Good for quick, sourced research; still verify the sources it cites.

NotebookLM

Google's tool for questioning and summarising your own uploaded sources.

Google's tool for asking questions of and summarising your own uploaded sources. Useful for synthesising interview notes or research reports.

Cursor

An AI code editor that edits code across a project with a chat and agent mode.

An AI code editor that edits code across a project through chat and agent modes. PMs use it to prototype small tools; engineers use it for production code.

Lovable

A prompt-to-app builder for quickly generating web apps with a backend.

A prompt-to-app builder that generates web apps with a backend. Good for fast prototypes to test with users.

Replit

A browser-based development environment with an AI agent that builds and hosts apps.

A browser development environment with an AI agent that builds and hosts apps. Useful for quick prototypes and demos.

Bolt

A browser tool that generates and runs full-stack web apps from prompts.

A browser tool that generates and runs full-stack web apps from prompts. Use it to prototype flows for user testing.

n8n

An open-source workflow automation tool with AI and agent nodes.

An open-source workflow automation tool with AI and agent nodes. Good for connecting tools and automating internal processes.

Make

A visual automation platform connecting apps through scenarios.

A visual automation platform connecting apps through scenarios. Useful for no-code integrations with clear branching.

Zapier

No-code automation linking thousands of apps through triggers and actions.

No-code automation linking thousands of apps through triggers and actions. Quick for simple automations, but costs rise with volume.

LangChain

Framework for LLM applications, tools and retrievers.

A framework for LLM applications with tools, chains and retrievers. Understand it at a conceptual level to discuss implementation with engineers.

LangGraph

Graph-based orchestration for stateful agents.

Graph-based orchestration for stateful agents with loops and human checkpoints. Useful when an agent must follow a branching process.

LlamaIndex

Framework for data ingestion, indexing and retrieval.

A framework for data ingestion, indexing and retrieval. Relevant when your main challenge is connecting models to documents.

CrewAI

Multi-agent framework with role-based teams.

A multi-agent framework with role-based teams. Test whether separate agent roles improve results before adopting it.

Pinecone

Managed vector database.

A managed vector database. Relevant when you evaluate retrieval infrastructure for a RAG product.

Weaviate

Open-source vector database with hybrid search.

An open-source vector database with hybrid search. Consider it when you need self-hosting and hybrid retrieval.

Chroma

Lightweight open-source vector store.

A lightweight open-source vector store, common in prototypes. Check production readiness before scaling.

Postman

API testing and documentation tool.

A tool for testing and documenting APIs. PMs use it to understand and check integrations.

GitHub

Code hosting, pull requests, issues and automation.

Code hosting, pull requests, issues and automation. PMs read pull requests and documentation to follow releases.

Figma

Collaborative interface design and prototyping.

Collaborative interface design and prototyping. Learn to read components and comment on designs.

Jira

Issue and sprint tracking.

Issue and sprint tracking. Learn the team's workflow and keep tickets clear and current.

Linear

Fast, minimal issue tracking for product and engineering teams.

Fast issue tracking for product and engineering teams. Its simple workflow suits small teams.

Notion

Workspace for docs, wikis and lightweight databases, with built-in AI.

A workspace for docs, wikis and lightweight databases, with built-in AI. Good for shared product documentation.

Mixpanel

Event-based product analytics.

Event-based product analytics for funnels, retention and cohorts. Used to answer everyday product questions.

Amplitude

Product analytics with cohorts, journeys and experiments.

Product analytics with cohorts, journeys and experimentation. Used for behavioural analysis at scale.

LangSmith

Tracing, evaluation and monitoring for LLM applications.

Tracing, evaluation and monitoring for LLM applications. Use it to inspect why an AI response was produced.

Arize Phoenix

Open-source observability and evaluation for LLM apps.

Open-source observability and evaluation for LLM applications. Useful for tracing and evaluating retrieval quality.

MLflow

Open-source platform for experiment tracking, model registry and deployment.

An open-source platform for experiment tracking, model registry and deployment. It records which model and parameters produced which result.

Cloud AI Platforms

Managed AI services: Amazon Bedrock, Azure AI Foundry and Google Vertex AI.

Managed AI services such as Amazon Bedrock, Azure AI Foundry and Google Vertex AI. They combine model access, security controls and monitoring in one place.

AI Evaluation Platforms

Tools that manage test sets, scoring and regression checks for LLM apps, such as LangSmith, Braintrust and Arize.

Tools that manage test sets, scoring and regression checks for LLM applications, such as LangSmith, Braintrust and Arize. Choose one that fits your stack and data rules.

Summary of terms

Term	Meaning
ChatGPT	OpenAI's assistant for writing, analysis, coding and research, with custom GPTs and tools.
Claude	Anthropic's assistant and API, known for long documents, writing and coding, with projects and tool use.
Gemini	Google's multimodal assistant and models, integrated with Workspace and Google Cloud.
Perplexity	An answer engine that searches the web and cites sources.
NotebookLM	Google's tool for questioning and summarising your own uploaded sources.
Cursor	An AI code editor that edits code across a project with a chat and agent mode.
Lovable	A prompt-to-app builder for quickly generating web apps with a backend.
Replit	A browser-based development environment with an AI agent that builds and hosts apps.
Bolt	A browser tool that generates and runs full-stack web apps from prompts.
n8n	An open-source workflow automation tool with AI and agent nodes.
Make	A visual automation platform connecting apps through scenarios.
Zapier	No-code automation linking thousands of apps through triggers and actions.
LangChain	Framework for LLM applications, tools and retrievers.
LangGraph	Graph-based orchestration for stateful agents.
LlamaIndex	Framework for data ingestion, indexing and retrieval.
CrewAI	Multi-agent framework with role-based teams.
Pinecone	Managed vector database.
Weaviate	Open-source vector database with hybrid search.
Chroma	Lightweight open-source vector store.
Postman	API testing and documentation tool.
GitHub	Code hosting, pull requests, issues and automation.
Figma	Collaborative interface design and prototyping.
Jira	Issue and sprint tracking.
Linear	Fast, minimal issue tracking for product and engineering teams.
Notion	Workspace for docs, wikis and lightweight databases, with built-in AI.
Mixpanel	Event-based product analytics.
Amplitude	Product analytics with cohorts, journeys and experiments.
LangSmith	Tracing, evaluation and monitoring for LLM applications.
Arize Phoenix	Open-source observability and evaluation for LLM apps.
MLflow	Open-source platform for experiment tracking, model registry and deployment.
Cloud AI Platforms	Managed AI services: Amazon Bedrock, Azure AI Foundry and Google Vertex AI.
AI Evaluation Platforms	Tools that manage test sets, scoring and regression checks for LLM apps, such as LangSmith, Braintrust and Arize.

Practice exercise: Pick one tool per category (assistant, builder, automation, vector store, observability) and write when you would use it and when you would not.

Self-check

1. Explain ChatGPT in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Pinecone in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain AI Evaluation Platforms in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- AI Skills Academy : Every AI skills course in one place.
- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Product Leadership and Stakeholder Management

The people skills that turn good plans into shipped results: influence, communication and decision-making.

Chapter 22 of 26 · 18 topics · Edition 2026 · skillnavora.in

CHAPTER 22

Product Leadership and Stakeholder Management

The people skills that turn good plans into shipped results: influence, communication and decision-making.

Product Leadership Fundamentals

Setting direction, building clarity and accountability, and growing others while keeping the product outcome in view.

Product leaders set direction, build clarity and accountability, and grow others while keeping the customer outcome in view. Leadership is measured by what the team can do without the leader.

Stakeholder Mapping

Listing stakeholders by influence and interest to decide how to engage each.

List stakeholders by influence and interest, then decide how to engage each. An executive with high influence and low interest needs short, decision-focused updates.

Stakeholder Communication

Tailored, regular updates that state decisions, risks and asks clearly.

Tailored, regular updates state decisions, risks and requests clearly. Stakeholders trust a PM who reports bad news early.

Influence Without Authority

Gaining support through evidence, relationships, shared goals and understanding others' incentives.

Gain support through evidence, relationships, shared goals and understanding of others' incentives. Most PM work happens without direct authority over the people involved.

Cross-Functional Leadership

Aligning product, engineering, design, data, sales and operations on one outcome.

Align product, engineering, design, data, sales and operations on one outcome. Shared goals and a common view of the data reduce conflict.

Engineering Collaboration

Involving engineers early, respecting constraints and agreeing trade-offs on scope and quality.

Involve engineers early, respect constraints, and agree trade-offs on scope and quality together. An engineer's estimate of risk is valuable input, not an obstacle.

Design Collaboration

Co-owning the problem with designers, sharing research and critiquing work against goals.

Co-own the problem with designers, share research and critique work against goals rather than taste. Good critique asks what the user is trying to do.

Data Science Collaboration

Framing the problem, agreeing success metrics and evaluation, and understanding model limits together.

Frame the problem together, agree success metrics and evaluation, and share an understanding of model limits. Data scientists know what the data can and cannot support.

Executive Communication

Lead with the conclusion, give the numbers and the decision needed, and keep it short.

Lead with the conclusion, give the key numbers and state the decision needed. Keep it short, and prepare for the question you least want to be asked.

Product Presentations

Structure around the audience's question: context, insight, recommendation and next steps.

Structure the talk around the audience's question: context, insight, recommendation and next steps. Show the evidence behind each claim.

Negotiation

Reaching agreements by understanding interests, creating options and trading what each side values.

Reach agreement by understanding interests, creating options and trading what each side values less for what it values more. Negotiate scope and date together, not separately.

Conflict Resolution

Surfacing the real disagreement, using shared facts and focusing on the goal.

Surface the real disagreement, use shared facts and refocus on the goal. Most product conflicts are about different assumptions about the user.

Decision-Making

Clear owner, defined criteria, enough evidence for the risk, and a record of the decision.

A good decision has a clear owner, defined criteria, enough evidence for the risk involved and a written record. Decide at the lowest level that has the information needed.

Product Reviews

Scheduled reviews of progress, metrics and risks with leadership.

Scheduled reviews of progress, metrics and risks with leadership. Bring the metrics, what changed, what you learned and what you need.

Product Governance

Standards and forums that guide what is built, approved and retired.

Standards and forums that guide what is built, approved and retired. Good governance speeds decisions by making rules clear in advance.

Managing Product Teams

Hiring, coaching, setting expectations and giving feedback to PMs.

Hire, coach, set expectations and give feedback to PMs. Give feedback close to the event and focus on behaviour and outcome.

Strategic Alignment

Making sure team goals connect to company strategy and to each other.

Make sure team goals connect to company strategy and to each other. Check alignment by asking teams to explain the company goal in their own words.

Product Culture

Shared habits of curiosity, customer focus, evidence and accountability.

Shared habits of curiosity, customer focus, evidence and accountability. Culture shows in what teams do when no one is watching, such as how they handle a failed launch.

Summary of terms

Term	Meaning
Product Leadership Fundamentals	Setting direction, building clarity and accountability, and growing others while keeping the product outcome in view.
Stakeholder Mapping	Listing stakeholders by influence and interest to decide how to engage each.
Stakeholder Communication	Tailored, regular updates that state decisions, risks and asks clearly.
Influence Without Authority	Gaining support through evidence, relationships, shared goals and understanding others' incentives.
Cross-Functional Leadership	Aligning product, engineering, design, data, sales and operations on one outcome.
Engineering Collaboration	Involving engineers early, respecting constraints and agreeing trade-offs on scope and quality.
Design Collaboration	Co-owning the problem with designers, sharing research and critiquing work against goals.

Data Science Collaboration	Framing the problem, agreeing success metrics and evaluation, and understanding model limits together.
Executive Communication	Lead with the conclusion, give the numbers and the decision needed, and keep it short.
Product Presentations	Structure around the audience's question: context, insight, recommendation and next steps.
Negotiation	Reaching agreements by understanding interests, creating options and trading what each side values.
Conflict Resolution	Surfacing the real disagreement, using shared facts and focusing on the goal.
Decision-Making	Clear owner, defined criteria, enough evidence for the risk, and a record of the decision.
Product Reviews	Scheduled reviews of progress, metrics and risks with leadership.
Product Governance	Standards and forums that guide what is built, approved and retired.
Managing Product Teams	Hiring, coaching, setting expectations and giving feedback to PMs.
Strategic Alignment	Making sure team goals connect to company strategy and to each other.
Product Culture	Shared habits of curiosity, customer focus, evidence and accountability.

Practice exercise: Map stakeholders for a launch, then write the executive update: decision needed, recommendation, risks, and next steps in under 200 words.

Self-check

1. Explain Product Leadership Fundamentals in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Product Presentations in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Product Culture in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- Product Operations : Free PDF, basic to advanced.
- 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.



AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Product Management Specialisations

How the PM role changes by product type, and what each specialisation emphasises.

Chapter 23 of 26 · 16 topics · Edition 2026 · skillnavora.in

CHAPTER 23

Product Management Specialisations

How the PM role changes by product type, and what each specialisation emphasises.

Technical Product Management

Products where technical depth matters: infrastructure, developer tools, APIs and data platforms.

Products where technical depth matters, such as infrastructure, developer tools, APIs and data platforms. The PM must discuss architecture trade-offs and developer experience credibly.

Growth Product Management

Owning acquisition, activation, retention and revenue through experiments.

Owns acquisition, activation, retention and revenue through experiments. Growth PMs work from funnels and cohorts and ship many small tests.

Platform Product Management

Building shared capabilities that other teams or partners build on, with internal or external customers.

Builds shared capabilities that other teams or partners build on. Success means other teams adopt the platform without friction.

B2B Product Management

Selling to organisations: multiple stakeholders, integrations, security and long cycles.

Selling to organisations involves several stakeholders, integrations, security reviews and long cycles. Success depends on admin controls, reporting and reliability.

B2C Product Management

Consumer products: scale, delight, habits and rapid experimentation.

Consumer products depend on scale, delight, habit formation and rapid experiments. Retention and daily use are central metrics.

SaaS Product Management

Subscription software: onboarding, retention, expansion and continuous delivery.

Subscription software needs strong onboarding, retention, expansion and continuous delivery. Pricing and packaging are ongoing product decisions.

Fintech Product Management

Financial products with regulation, risk, fraud, reliability and trust at the centre.

Financial products have regulation, risk, fraud, reliability and trust at the centre. Compliance and security requirements shape the roadmap.

Banking Product Management

Retail, corporate and payments products inside regulated institutions with legacy integration.

Retail, corporate and payments products inside regulated institutions, often with legacy system integration. Change moves slowly, so plan releases carefully.

E-commerce Product Management

Catalogue, search, checkout, fulfilment and conversion.

Catalogue, search, checkout, fulfilment and conversion. Small improvements in checkout speed can have large revenue effects.

Marketplace Product Management

Balancing supply and demand, trust and safety, pricing and liquidity.

Balances supply and demand, trust and safety, pricing and liquidity. A marketplace fails when one side leaves, so watch both sides' metrics.

API Product Management

Treating an API as a product: developer experience, documentation, versioning and pricing.

Treats an API as a product: developer experience, documentation, versioning and pricing. Breaking changes damage trust quickly.

Data Product Management

Managing data assets and analytics products with quality, access and usability for consumers.

Manages data assets and analytics products with quality, access and usability for consumers. Data products are judged by whether people trust and use them.

AI Product Management

Managing products whose core value comes from machine learning, with uncertainty, evaluation and data in scope.

Products whose core value comes from machine learning, with uncertainty, evaluation and data in scope. The PM must define quality in measurable terms.

Generative AI Product Management

Products built on foundation models: prompts, retrieval, evaluation, cost and safety.

Products built on foundation models: prompts, retrieval, evaluation, cost and safety. Output quality varies, so the PM manages it continuously.

Agentic AI Product Management

Products where agents act on a user's behalf: autonomy, permissions, reliability and oversight.

Products where agents act on a user's behalf: autonomy, permissions, reliability and human oversight. The main questions are what the agent may do and how it is stopped.

Enterprise AI Product Management

AI for organisations: integration, governance, security, adoption and measurable ROI.

AI for organisations: integration, governance, security, adoption and measurable return on investment. Enterprise buyers ask for audit logs, data controls and proof of value.

Summary of terms

Term	Meaning
Technical Product Management	Products where technical depth matters: infrastructure, developer tools, APIs and data platforms.
Growth Product Management	Owning acquisition, activation, retention and revenue through experiments.
Platform Product Management	Building shared capabilities that other teams or partners build on, with internal or external customers.
B2B Product Management	Selling to organisations: multiple stakeholders, integrations, security and long cycles.
B2C Product Management	Consumer products: scale, delight, habits and rapid experimentation.
SaaS Product Management	Subscription software: onboarding, retention, expansion and continuous delivery.
Fintech Product Management	Financial products with regulation, risk, fraud, reliability and trust at the centre.
Banking Product Management	Retail, corporate and payments products inside regulated institutions with legacy integration.
E-commerce Product Management	Catalogue, search, checkout, fulfilment and conversion.
Marketplace Product Management	Balancing supply and demand, trust and safety, pricing and liquidity.
API Product Management	Treating an API as a product: developer experience, documentation, versioning and pricing.
Data Product Management	Managing data assets and analytics products with quality, access and usability for consumers.
AI Product Management	Managing products whose core value comes from machine learning, with uncertainty, evaluation and data in scope.
Generative AI Product Management	Products built on foundation models: prompts, retrieval, evaluation, cost and safety.
Agentic AI Product Management	Products where agents act on a user's behalf: autonomy, permissions, reliability and oversight.
Enterprise AI Product Management	AI for organisations: integration, governance, security, adoption and measurable ROI.

Practice exercise: Choose the specialisation you are targeting. List the five skills it weights most and your current gap on each.

Self-check

1. Explain Technical Product Management in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain E-commerce Product Management in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Enterprise AI Product Management in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- Platform and API Product Management : API products, versioning and developer experience.
- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Industry Case Studies and Product Teardowns

A method for studying real products. The notes below are starting prompts for your own research from public sources, not claims about internal data.

Chapter 24 of 26 · 20 topics · Edition 2026 · skillnavora.in



CHAPTER 24

Industry Case Studies and Product Teardowns

A method for studying real products. The notes below are starting prompts for your own research from public sources, not claims about internal data.

OpenAI – ChatGPT

Study how a chat interface became a platform: free vs paid tiers, model releases, plugins/GPTs and enterprise offerings.

Study how a chat interface grew into a platform: free and paid tiers, model releases, custom GPTs and enterprise offerings. Research from public sources only.

Anthropic – Claude

Study the safety-focused positioning, long context, coding use and API-first business.

Study the safety-focused positioning, long context, coding use and API-first business. Compare its packaging for individuals and for enterprises.

Google – Gemini

Study integration across Search, Workspace and Cloud, and the multimodal strategy.

Study the integration across Search, Workspace and Cloud, and the multimodal strategy. Ask how distribution through existing products changes adoption.

Microsoft – Copilot

Study embedding AI across Office, GitHub and Windows, seat-based pricing and enterprise distribution.

Study AI across Office, GitHub and Windows, the seat-based pricing and enterprise distribution. Ask how bundling affects adoption.

Perplexity – AI Search

Study answer-engine UX, citations and the challenge to traditional search.

Study answer-engine interface design, citations and the challenge to traditional search. Ask how trust is built when the answer replaces the list of links.

Salesforce – Agentforce

Study agents inside a CRM: data grounding, actions, trust layer and pricing.

Study agents inside a CRM: data grounding, actions, trust controls and pricing. Ask how permissions are managed across customer records.

Adobe – Firefly

Study commercially safe generative models integrated into creative workflows.

Study commercially safer generative models integrated into creative workflows. Ask how licensing shapes the product promise.

Amazon – AI Shopping Experiences

Study AI in discovery, reviews summaries and assistants within a marketplace.

Study AI in product discovery, review summaries and shopping assistants inside a marketplace. Ask how the assistant changes conversion.

Netflix – Recommendation Systems

Study personalisation, artwork selection, experimentation culture and retention.

Study personalisation, artwork selection, experimentation culture and retention. Ask how the team measures the value of recommendations.

Spotify – Personalisation

Study discovery playlists, taste profiles and how personalisation drives engagement.

Study discovery playlists, taste profiles and how personalisation drives engagement. Ask how the product balances familiar and new content.

Uber – Marketplace and Pricing

Study matching, dynamic pricing, supply-demand balance and trust.

Study matching, dynamic pricing, supply and demand balance and trust. Ask how each side of the marketplace is measured.

Airbnb – Marketplace Product Strategy

Study trust mechanisms, search ranking, two-sided growth and host experience.

Study trust mechanisms, search ranking, two-sided growth and host experience. Ask how quality is maintained as supply grows.

Notion – AI Workspace

Study AI embedded in documents and databases and its add-on monetisation.

Study AI embedded in documents and databases, and its add-on monetisation. Ask how users discover the feature.

ServiceNow – AI Workflows

Study enterprise workflow automation with AI agents and governance.

Study enterprise workflow automation with AI agents and governance. Ask how approvals and audit trails are designed.

Freshworks – AI Customer Support

Study AI-assisted support: deflection, agent assist and resolution metrics.

Study AI-assisted support: deflection, agent assist and resolution metrics. Ask how customer satisfaction is protected when the AI answers.

a large company – Financial Services and AI Use Cases

Study fraud detection, personalisation and servicing in a regulated setting.

Study fraud detection, personalisation and servicing in a regulated setting. Ask how the bank balances customer experience with risk controls.

Flipkart – E-commerce Product Strategy

Study serving diverse Indian customers: regional languages, payments, logistics and sales events.

Study serving diverse Indian customers: regional languages, payments, logistics and large sales events. Ask how each constraint shapes the product.

PhonePe – Fintech Product Strategy

Study UPI-led growth, trust, merchant payments and adjacent financial services.

Study UPI-led growth, trust, merchant payments and adjacent financial services. Ask how the product expands beyond payments.

Razorpay – Payment Products

Study payment gateway, developer experience, banking stack and merchant onboarding.

Study the payment gateway, developer experience, banking stack and merchant onboarding. Ask how the API design lowers integration effort.

Swiggy – Marketplace and Delivery Products

Study three-sided marketplace dynamics, delivery times, pricing and quick commerce.

Study three-sided marketplace dynamics, delivery times, pricing and quick commerce. Ask how the product balances speed and cost.

Summary of terms

Term	Meaning
OpenAI – ChatGPT	Study how a chat interface became a platform: free vs paid tiers, model releases, plugins/GPTs and enterprise offerings.
Anthropic – Claude	Study the safety-focused positioning, long context, coding use and API-first business.
Google – Gemini	Study integration across Search, Workspace and Cloud, and the multimodal strategy.
Microsoft – Copilot	Study embedding AI across Office, GitHub and Windows, seat-based pricing and enterprise distribution.

Perplexity – AI Search	Study answer-engine UX, citations and the challenge to traditional search.
Salesforce – Agentforce	Study agents inside a CRM: data grounding, actions, trust layer and pricing.
Adobe – Firefly	Study commercially safe generative models integrated into creative workflows.
Amazon – AI Shopping Experiences	Study AI in discovery, reviews summaries and assistants within a marketplace.
Netflix – Recommendation Systems	Study personalisation, artwork selection, experimentation culture and retention.
Spotify – Personalisation	Study discovery playlists, taste profiles and how personalisation drives engagement.
Uber – Marketplace and Pricing	Study matching, dynamic pricing, supply-demand balance and trust.
Airbnb – Marketplace Product Strategy	Study trust mechanisms, search ranking, two-sided growth and host experience.
Notion – AI Workspace	Study AI embedded in documents and databases and its add-on monetisation.
ServiceNow – AI Workflows	Study enterprise workflow automation with AI agents and governance.
Freshworks – AI Customer Support	Study AI-assisted support: deflection, agent assist and resolution metrics.
a large company **– Financial Services and AI Use Cases**	Study fraud detection, personalisation and servicing in a regulated setting.
Flipkart – E-commerce Product Strategy	Study serving diverse Indian customers: regional languages, payments, logistics and sales events.
PhonePe – Fintech Product Strategy	Study UPI-led growth, trust, merchant payments and adjacent financial services.
Razorpay – Payment Products	Study payment gateway, developer experience, banking stack and merchant onboarding.
Swiggy – Marketplace and Delivery Products	Study three-sided marketplace dynamics, delivery times, pricing and quick commerce.

Practice exercise: Teardown one product from this list using public sources: users, job, business model, metrics you would watch, and one change you would make.

Self-check

1. Explain OpenAI – ChatGPT in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Uber – Marketplace and Pricing in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Swiggy – Marketplace and Delivery Products in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- 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.

AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

End-to-End Practical Capstone Projects

Twelve portfolio projects. Each must include the full deliverable checklist below, so the work reads like a real product case.

Chapter 25 of 26 · 12 topics · Edition 2026 · skillnavora.in



CHAPTER 25

End-to-End Practical Capstone Projects

Twelve portfolio projects. Each must include the full deliverable checklist below, so the work reads like a real product case.

Build an E-commerce Product Strategy.

Define target shoppers, positioning, growth levers (search, recommendations, checkout) and a prioritised roadmap with metrics.

Define target shoppers, positioning, growth levers such as search, recommendations and checkout, and a prioritised roadmap with metrics for each lever.

Design a Fintech Payment Product.

Cover onboarding and KYC, payment flow, fraud and risk controls, compliance, reliability targets and merchant or consumer value.

Cover onboarding and KYC, the payment flow, fraud and risk controls, compliance requirements, reliability targets and the value to merchants and consumers.

Build a Product Analytics Dashboard.

Choose a north star and input metrics, define events, build funnel and retention views, and document how each metric drives a decision.

Choose a north star and input metrics, define events, build funnel and retention views, and document how each metric drives a decision.

Create a Customer Onboarding Optimisation Plan.

Map the activation journey, find drop-offs from data, propose experiments and set success targets.

Map the activation journey, find drop-offs from data, propose experiments and set success targets for each stage.

Develop an AI Customer Support Assistant.

Scope intents, knowledge sources, escalation to humans, quality and safety evaluation, and deflection and satisfaction metrics.

Scope the intents, knowledge sources and escalation to humans. Define quality and safety evaluation, and track deflection and satisfaction.

Design a RAG-Based Enterprise Knowledge Assistant.

Specify sources, permissions, chunking and retrieval design, citation behaviour, evaluation set and monitoring.

Specify sources, permissions, chunking and retrieval design, citation behaviour, the evaluation set and monitoring.

Build an AI Document Intelligence Product.

Extract and validate fields from documents, handle low-confidence cases with review, and measure accuracy and time saved.

Extract and validate fields from documents, route low-confidence cases to human review, and measure accuracy and time saved.

Design an AI Recommendation Engine.

Choose signals and approach, cold-start handling, diversity and fairness, offline and online evaluation, and business metrics.

Choose the signals and approach, handle cold start for new users and items, ensure diversity and fairness, and define offline and online evaluation with business metrics.

Create an AI Customer Complaint Management System.

Classify, route and summarise complaints, set priorities and SLAs, include human review and compliance tracking.

Classify, route and summarise complaints, set priorities and service-level targets, and include human review and compliance tracking.

Build a Multi-Agent Business Automation Workflow.

Define agent roles, tools, permissions, hand-offs, approval points, failure handling and cost limits.

Define agent roles, tools, permissions, hand-offs, approval points, failure handling and cost limits.

Develop an AI Sales Assistant.

Support prospecting, call preparation and follow-up with CRM grounding, guardrails and pipeline-impact metrics.

Support prospecting, call preparation and follow-up with CRM grounding and guardrails. Track pipeline impact, not just usage.

Create an AI-Powered Product Analytics Assistant.

Let users ask data questions in natural language with safe SQL generation, verification and clear lineage of answers.

Let users ask data questions in plain language, generate safe queries, verify results and show where each answer came from.

Summary of terms

Term	Meaning
Build an E-commerce Product Strategy.	Define target shoppers, positioning, growth levers (search, recommendations, checkout) and a prioritised roadmap with metrics.
Design a Fintech Payment Product.	Cover onboarding and KYC, payment flow, fraud and risk controls, compliance, reliability targets and merchant or consumer value.

Build a Product Analytics Dashboard.	Choose a north star and input metrics, define events, build funnel and retention views, and document how each metric drives a decision.
Create a Customer Onboarding Optimisation Plan.	Map the activation journey, find drop-offs from data, propose experiments and set success targets.
Develop an AI Customer Support Assistant.	Scope intents, knowledge sources, escalation to humans, quality and safety evaluation, and deflection and satisfaction metrics.
Design a RAG-Based Enterprise Knowledge Assistant.	Specify sources, permissions, chunking and retrieval design, citation behaviour, evaluation set and monitoring.
Build an AI Document Intelligence Product.	Extract and validate fields from documents, handle low-confidence cases with review, and measure accuracy and time saved.
Design an AI Recommendation Engine.	Choose signals and approach, cold-start handling, diversity and fairness, offline and online evaluation, and business metrics.
Create an AI Customer Complaint Management System.	Classify, route and summarise complaints, set priorities and SLAs, include human review and compliance tracking.
Build a Multi-Agent Business Automation Workflow.	Define agent roles, tools, permissions, hand-offs, approval points, failure handling and cost limits.
Develop an AI Sales Assistant.	Support prospecting, call preparation and follow-up with CRM grounding, guardrails and pipeline-impact metrics.
Create an AI-Powered Product Analytics Assistant.	Let users ask data questions in natural language with safe SQL generation, verification and clear lineage of answers.

Practice exercise: Pick one capstone and complete every item of the deliverable checklist. Present it in ten minutes.

Self-check

1. Explain Build an E-commerce Product Strategy. in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain Build an AI Document Intelligence Product. in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Create an AI-Powered Product Analytics Assistant. in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- 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.



AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Interview Preparation and Career Readiness

Preparing for traditional PM and AI PM interviews, with practice, portfolio and skill-gap work. The full question bank is in the Interview guide on this site.

Chapter 26 of 26 · 22 topics · Edition 2026 · skillnavora.in

CHAPTER 26

Interview Preparation and Career Readiness

Preparing for traditional PM and AI PM interviews, with practice, portfolio and skill-gap work. The full question bank is in the Interview guide on this site.

Product Management Interview Fundamentals

Know the loop: screens, product sense, execution, analytics, technical, strategy and behavioural rounds.

Know the typical loop: screening, product sense, execution, analytics, technical, strategy and behavioural rounds. Prepare a story bank that covers each.

Associate Product Manager Interviews

Emphasise structured thinking, curiosity, user empathy and learning ability over experience.

Emphasise structured thinking, curiosity, user empathy and learning ability over experience. Show how you learned from a project.

Product Manager Interviews

Show end-to-end ownership, prioritisation, metrics and influence with examples.

Show end-to-end ownership, prioritisation, metrics and influence with examples and numbers.

Product Owner Interviews

Show backlog management, story writing, Scrum fluency and stakeholder handling.

Show backlog management, story writing, Scrum fluency and stakeholder handling.

Product Analyst Interviews

Expect SQL, metrics definition, experiment analysis and data storytelling.

Expect SQL, metric definition, experiment analysis and data storytelling. Practise explaining a result to a non-technical audience.

AI Product Manager Interviews

Expect AI fundamentals, evaluation, cost and risk trade-offs, and product sense for AI features.

Expect AI fundamentals, evaluation, cost and risk trade-offs, and product sense for AI features. Use the interview guide on this site to practise.

Product Sense Interviews

Structure: user, problem, solutions, prioritise, metrics.

Use a clear structure: the user, the problem, possible solutions, prioritisation and metrics.

Product Design Interviews

Design a product for a user group, with flows, trade-offs and validation.

Design a product for a user group, covering flows, trade-offs and validation with users.

Product Strategy Interviews

Market analysis, positioning, moats, and a recommendation with risks.

Cover market analysis, positioning, competitive moats and a recommendation with its risks.

Technical Product Interviews

Explain systems, APIs and trade-offs clearly and discuss feasibility.

Explain systems, APIs and trade-offs clearly and discuss feasibility with an engineer.

AI/ML Technical Interviews

Explain model types, metrics, LLM concepts, prompting, RAG and fine-tuning choices.

Explain model types, metrics, LLM concepts, prompting, retrieval and when to fine-tune.

RAG and Agentic AI Interviews

Design a RAG or agent system, discuss evaluation, failure modes, permissions and cost.

Design a retrieval or agent system and discuss evaluation, failure modes, permissions and cost.

Product Analytics Interviews

Define metrics, diagnose a drop, design an experiment and read a funnel.

Define metrics, diagnose a sudden drop, design an experiment and read a funnel.

Product Case Studies

Practise end-to-end cases from problem framing to rollout and measurement.

Practise end-to-end cases from problem framing through rollout and measurement. Time yourself and state assumptions aloud.

Business Case Interviews

Estimate market size, profitability and ROI with clear assumptions.

Estimate market size, profitability and return on investment with clear assumptions and sensitivity checks.

Behavioural Interviews

Prepare STAR stories on conflict, failure, influence and leadership, each with a number.

Prepare STAR stories (situation, task, action, result) on conflict, failure, influence and leadership, each with a measurable result.

Stakeholder Management Interviews

Show how you handle disagreement, ambiguity and competing priorities.

Show how you handle disagreement, ambiguity and competing priorities, with a specific example.

Product Execution Interviews

Show how you plan launches, handle risks and debug a metric drop.

Show how you plan launches, manage risks and diagnose a metric drop.

Product Leadership Interviews

Show vision, team building, strategy and organisational influence.

Show vision, team building, strategy and organisational influence, with examples at the right scope.

Resume and Portfolio Development

Quantified outcomes, clear scope, and 2-3 capstone case studies with artefacts.

Write quantified outcomes and clear scope. Build two or three case studies from your capstone projects with artefacts.

Mock Interviews

Timed practice with feedback on structure, clarity and depth, recorded where possible.

Practise under time limits with feedback on structure, clarity and depth. Record the session and review it.

Job Description-Based Skill Gap Analysis

Compare a target job's requirements with your skills and build a learning plan; see the industry research page for skill frequencies.

Compare a target job's requirements with your skills and build a learning plan. The industry research page shows which skills employers name most often.

Summary of terms

Term	Meaning
Product Management Interview Fundamentals	Know the loop: screens, product sense, execution, analytics, technical, strategy and behavioural rounds.
Associate Product Manager Interviews	Emphasise structured thinking, curiosity, user empathy and learning ability over experience.
Product Manager Interviews	Show end-to-end ownership, prioritisation, metrics and influence with examples.
Product Owner Interviews	Show backlog management, story writing, Scrum fluency and stakeholder handling.
Product Analyst Interviews	Expect SQL, metrics definition, experiment analysis and data storytelling.

AI Product Manager Interviews	Expect AI fundamentals, evaluation, cost and risk trade-offs, and product sense for AI features.
Product Sense Interviews	Structure: user, problem, solutions, prioritise, metrics.
Product Design Interviews	Design a product for a user group, with flows, trade-offs and validation.
Product Strategy Interviews	Market analysis, positioning, moats, and a recommendation with risks.
Technical Product Interviews	Explain systems, APIs and trade-offs clearly and discuss feasibility.
AI/ML Technical Interviews	Explain model types, metrics, LLM concepts, prompting, RAG and fine-tuning choices.
RAG and Agentic AI Interviews	Design a RAG or agent system, discuss evaluation, failure modes, permissions and cost.
Product Analytics Interviews	Define metrics, diagnose a drop, design an experiment and read a funnel.
Product Case Studies	Practise end-to-end cases from problem framing to rollout and measurement.
Business Case Interviews	Estimate market size, profitability and ROI with clear assumptions.
Behavioural Interviews	Prepare STAR stories on conflict, failure, influence and leadership, each with a number.
Stakeholder Management Interviews	Show how you handle disagreement, ambiguity and competing priorities.
Product Execution Interviews	Show how you plan launches, handle risks and debug a metric drop.
Product Leadership Interviews	Show vision, team building, strategy and organisational influence.
Resume and Portfolio Development	Quantified outcomes, clear scope, and 2-3 capstone case studies with artefacts.
Mock Interviews	Timed practice with feedback on structure, clarity and depth, recorded where possible.
Job Description-Based Skill Gap Analysis	Compare a target job's requirements with your skills and build a learning plan; see the industry research page for skill frequencies.

Practice exercise: Take a real job description, extract the required skills, mark your gaps, and run two timed mock interviews on the weakest ones.

Self-check

1. Explain Product Management Interview Fundamentals in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain RAG and Agentic AI Interviews in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain Job Description-Based Skill Gap Analysis in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- AI Product Management (original programme) ：The programme you already have: PM Master Handbook, PRD and BRD templates, interview prep.

- Interactive PM Handbook : 23 modules: product core, AI tools lab, frameworks, case studies and interview prep.
- Top 100 interview questions : Practice questions with answers.
- AI PM interview guide (27 questions) : Model answers tied to job-description research.
- AI PM industry research : Courses, job records and skills employers name most.
- Practice tests : Timed tests by programme.
- Quiz zone : Short quizzes.
- Interview prep hub : Questions, tests and quizzes.
- 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