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.

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