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.

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