

Become a Product Owner by actually doing the work.

The full curriculum: 10 stages, 28 modules, 98 lessons. Self-paced. Every module closes with the Loop — draft a real artifact, get it AI-graded against a rubric, then defend it against an AI stakeholder. No multiple-choice, no talking-head video: the artifacts you ship become your portfolio, your capstone, and your certificate.

STAGE 1 / FOUNDATIONS

Zero. What the job is, why it exists, and how value flows through a product team.

01 — The Product Owner Role

The role, drawn against everyone next to it.

Ships: Role-definition 1-pager

- Owner vs PM, BA & Scrum Master — Distinguish the PO's accountabilities from adjacent roles and name what only the PO owns.
- Agile, Scrum & the value mindset — Explain why agile exists and how a PO maximizes value inside a Scrum or Kanban team.
- A day, a sprint, a backlog — Map the PO's cadence across the ceremonies and the lifecycle of a single backlog item.
- PO anti-patterns — Spot the ticket-monkey and proxy-PO traps and describe how a real PO operates instead.

02 — Product Thinking & Strategy

Aim before you build — outcomes, not output.

Ships: One-page product strategy

- Outcomes over outputs — Reframe a feature request as the user or business outcome it is meant to drive.
 - Vision, mission & strategy — Articulate a product vision and connect it to a coherent, defensible strategy.
 - How products make money — Identify a product's business model and the levers that move its economics.
 - The product lifecycle — Place a product on its lifecycle curve and adjust priorities to that stage.
-

STAGE 2 / DISCOVERY

Find the right problem before committing a single sprint to a solution.

03 — Customer Discovery & Research

Replace opinion with evidence about real users.

Ships: 5 interview scripts + synthesis

- Why discovery beats opinion — Justify continuous discovery over feature-factory, opinion-led decision-making.
- Jobs-to-be-Done — Frame a customer need as a job-to-be-done rather than a requested feature.
- Running great interviews — Write non-leading questions and run a session that surfaces real behavior, not flattery.
- Synthesizing research — Turn raw interview notes into patterns, insights, and usable personas.
- Quant + qual together — Combine analytics and survey signal with interview insight to size a problem.

04 — Problem & Opportunity Framing

Name the problem so sharply the solution gets obvious.

Ships: Opportunity solution tree

- Sharp problem statements — Write a problem statement naming user, pain, and context without smuggling in a solution.
 - Opportunity solution trees — Map opportunities to a desired outcome and branch candidate solutions beneath them.
 - Hypotheses & assumptions — Express a product bet as a testable hypothesis and surface its riskiest assumption.
 - Choosing where to play — Decide which single opportunity to pursue first against one clear outcome.
-

STAGE 3 / DEFINITION

Turn a chosen problem into work an engineer can build without guessing.

05 — The Product Backlog

A living plan — not a wish-list dump.

Ships: A refined, healthy backlog

- What a backlog really is — Define the backlog as a prioritized, living plan rather than a graveyard of ideas.
- Backlog hygiene & DEEP — Assess a backlog against the DEEP criteria and clean it back to health.
- Refinement that works — Facilitate a refinement session that leaves items ready and shared by the team.
- Definition of Ready — Write a Definition of Ready the team will actually honor.

06 — User Stories & Acceptance Criteria

Small, vertical, testable slices of value.

Ships: Story map + AC set

- Anatomy of a story (INVEST) — Write a user story that passes every letter of the INVEST test.
- Story splitting — Split an epic into vertically-sliced stories that each deliver value independently.
- Acceptance criteria & Gherkin — Write acceptance criteria as observable behavior QA can verify, not aspiration.
- Story mapping — Build a story map that exposes the backbone and a thin walking-skeleton release.

07 — Writing PRDs & Specs

Documents engineers actually read.

Ships: A defensible PRD

- Why most PRDs get ignored — Diagnose the four failure modes that get a spec skimmed and ignored.
- PRD anatomy — Structure a PRD around its five load-bearing sections.
- Edge cases & non-functionals — Surface edge cases, error states, and NFRs before engineering trips over them.
- Writing for the engineer's eye — Write a spec an engineer can build from without booking a meeting to decode it.

08 — UX & Design Collaboration

Partner with design without driving the pen.

Ships: Annotated prototype review

- How designers think — Describe the design process and where a PO adds value versus where they should step back.
- Reading wireframes & prototypes — Critique a prototype against the problem and AC instead of personal taste.
- Accessibility basics — Identify the accessibility floor every feature must clear before it ships.
- Content & microcopy — Spot UX copy that will generate support tickets and rewrite it.

STAGE 4 / PRIORITIZATION & PLANNING

Decide what to build, in what order, and what to leave behind.

09 — Prioritization

Saying no is the highest-leverage thing you do.

Ships: Scored, sequenced backlog

- Why prioritization is the job — Explain why deciding what not to build is a PO's core act, not an afterthought.
- RICE, WSJF & MoSCoW — Select and apply the right scoring framework for a given decision.
- Kano & value vs effort — Classify features by Kano type and plot value against effort to find quick wins.
- Cost of delay — Use cost of delay to sequence work when every stakeholder calls their ask "urgent."

10 — Roadmapping

A roadmap of outcomes, not a list of dates.

Ships: Now / next / later roadmap

- Roadmaps != release plans — Build a now/next/later roadmap organized by outcomes rather than fixed dates.
- Themes & bets — Group work into themes that visibly ladder up to product strategy.
- Communicating a roadmap — Present a roadmap that sets expectations without over-committing to deadlines.

11 — Estimation & Sprint Planning

Forecast honestly; plan a sprint with a point.

Ships: A planned sprint + goal

- Estimation & story points — Run relative estimation and explain why points beat hour-based estimates.

- Velocity & capacity — Forecast realistically from velocity without weaponizing it against the team.
- Sprint goals & planning — Write a sprint goal and assemble a sprint that genuinely serves it.

12 — Release & MVP Planning

Ship the smallest thing that learns the most.

Ships: Release plan

- MVP, MMP & slicing — Define an MVP that tests the riskiest assumption, not just the smallest build.
 - Release strategies — Choose a release approach — big-bang, phased, or flagged — for the risk at hand.
 - Sequencing a release — Sequence stories into a releasable increment built around a walking skeleton.
-

STAGE 5 / DELIVERY

Get the work built — well — alongside engineering and the team.

13 — Working with Engineering

Trade-offs, debt, and trust with the people who build.

Ships: Refinement minutes + trade-off log

- Speaking engineer — Hold a credible trade-off conversation without pretending to be the architect.
- Tech debt & non-functionals — Make the case for investing in tech debt in business terms leadership accepts.
- Definition of Done — Negotiate a Definition of Done that protects quality under delivery pressure.
- Scope / time / quality trade-offs — Facilitate a scope–time–quality trade-off and document the decision and its reason.

14 — Ceremonies & Facilitation

Make the Scrum events earn their hour.

Ships: Facilitated review + retro plan

- The PO across the ceremonies — State what the PO does and does not own in each Scrum event.
- Running a sprint review — Run a review that gathers real feedback instead of performing a demo.
- Retrospectives that change things — Facilitate a retro that produces one concrete, owned improvement.

15 — Dependencies & Risk

See the blockers before they block you.

Ships: Dependency & risk map

- Mapping dependencies — Surface cross-team dependencies before they harden into blockers.
 - Managing risk — Identify, rank, and mitigate delivery risks early instead of reactively.
 - Unblocking & escalation — Decide when to unblock yourself versus escalate — and escalate well.
-

STAGE 6 / STAKEHOLDERS & COMMUNICATION

Influence, align, and say no without authority over anyone.

16 — Stakeholder Management

Keep the room aligned and surprise-free.

Ships: Stakeholder map + 3 scripts

- Mapping stakeholders — Map stakeholders by power and interest and tailor your approach to each quadrant.
- Managing up — Keep leadership aligned, informed, and free of unwelcome surprises.
- Saying no with a yes-and — Decline a request while protecting both the relationship and the roadmap.
- Building buy-in — Secure buy-in for a decision before the meeting that is meant to ratify it.

17 — Communication & Storytelling

Make people care about the problem you chose.

Ships: Product pitch

- Structuring a product narrative — Structure a pitch around problem, insight, and bet rather than a feature list.
- Written communication — Write an update that a busy executive actually reads to the end.
- Demos & presentations — Deliver a demo that lands the value, not just the clicks.

18 — Negotiation & Hard Conversations

Trade well; disagree without damage.

Ships: Negotiation prep + roleplay

- Principled negotiation — Separate positions from interests to find a trade better than either opening ask.
 - Difficult conversations — Hold a hard conversation that preserves trust on the other side of it.
 - Influence without authority — Move a decision forward when you don't manage the people who own it.
-

STAGE 7 / DATA & MEASUREMENT

Know whether the thing you shipped actually worked.

19 — Product Metrics

Measure the outcome, not the activity.

Ships: Metrics tree + north star

- Outputs, outcomes & a north star — Choose a north-star metric and explain why it beats vanity metrics.
- Metric frameworks (AARRR, HEART) — Apply a metrics framework to map a product's funnel and experience.
- Building a metrics tree — Decompose a north star into the input metrics a team can actually move.
- Avoiding metric traps — Spot gaming, bad proxies, and Goodhart's-law failures before they mislead you.

20 — Experimentation

Turn opinions into tests with answers.

Ships: Experiment design

- Hypothesis-driven development — Turn a product bet into a falsifiable, measurable experiment.
- A/B testing basics — Design an A/B test and read significance without fooling yourself.
- Feature flags & rollouts — Plan a safe, measurable rollout using feature flags.
- Beyond A/B — Pick the right validation method — fake door, concierge, beta — for the bet.

21 — Data-Informed Decisions

Evidence and judgment, not one or the other.

Ships: Decision memo

- Data-informed, not data-driven — Balance evidence, judgment, and strategy when the data is ambiguous.
 - Analytics for POs — Pull and interpret the analytics a PO needs without waiting on a data team.
 - Writing a decision memo — Document a decision so it survives turnover and later second-guessing.
-

STAGE 8 / LAUNCH & GROWTH

Get it into the world, then keep making it better.

22 — Go-to-Market & Launch

Land the launch across product, support, and sales.

Ships: Launch plan + comms

- Launch readiness — Build a launch checklist covering product, support, and go-to-market.
- Working with marketing & sales — Equip go-to-market teams to sell and support what you shipped.
- Launch comms & enablement — Draft the internal and external communications a launch needs.

23 — Iteration & Continuous Discovery

Close the loop and decide what's next.

Ships: Retro + v2 priorities

- Closing the loop — Compare launch results to the original hypothesis and decide the next move.
 - Feedback channels — Build durable channels that keep customer feedback flowing after launch.
 - Planning v2 — Translate post-launch learning into a prioritized v2 backlog.
-

STAGE 9 / BUSINESS ACUMEN

Speak the language of the people who fund the work.

24 — Product Economics

Tie features to money — both directions.

Ships: Business case

- Unit economics & pricing — Read a product's unit economics and the levers behind its margins.
- Building a business case — Build a business case that ties a feature to revenue or cost.
- ROI & opportunity cost — Argue for an investment using ROI and the opportunity cost of not doing it.

25 — Domain, Legal & Compliance

Know which questions you must never skip.

Ships: Compliance-aware spec note

- Privacy & data (GDPR-aware) — Flag the privacy obligations a feature triggers before it is designed.
- Security & risk awareness — Recognize the security questions a PO is responsible for raising.
- Working in regulated domains — Adapt the PO playbook for a compliance-heavy domain.

26 — Scaling Product

Coordinate beyond a single team.

Ships: Scaled-team operating note

- Multiple teams & dependencies — Coordinate one backlog and its dependencies across several teams.
 - Scaling frameworks overview — Compare SAFe and LeSS and the trade-offs each scaling model imposes.
 - Portfolio alignment — Align team-level work to portfolio-level strategy and funding.
-

STAGE 10 / HERO — CAREER & CAPSTONE

Turn everything you built into a credential and a next step.

27 — Career & Growth

Grow influence well past your job title.

Ships: Growth plan

- PO vs PM & the ladder — Map the PO-to-PM path and the skills each rung actually demands.
- Leading without managing — Grow influence and seniority as an individual contributor.
- Building your product brand — Build a track record and network that compounds over a career.

28 — Interview & Portfolio

Prove it — and land the role.

Ships: Capstone case-study deck

- The PO interview loop — Map a typical PO interview loop and what each round is really testing.
 - Case & product-sense questions — Structure a confident answer to a product-sense or estimation prompt.
 - Your portfolio & capstone — Assemble your shipped artifacts into one coherent case-study narrative.
-

One price, one year of access: \$149 founding (was \$199), 14-day refund. Join at backlog.academy.