

20+ Practical AI Prompts Every Product Owner Should Use in Scrum Projects

Diwakar Singh

🌟 1. Backlog Refinement & Grooming

Prompt 1: User Story Refinement

“Refine the following user story to make it more clear, INVEST-compliant, and ready for development. Suggest acceptance criteria using Gherkin format.

User Story: As a user, I want to reset my password so that I can access my account if I forget it.”

✅ *Example Output:* Clear story, INVEST criteria validation, “Given–When–Then” acceptance criteria, edge cases.

Prompt 2: Split Large Epics into Features and Stories

“Break down the following epic into features and user stories. Make sure each story is small enough to be completed in a sprint and follows INVEST principles.

Epic: Implement a digital onboarding flow for new banking customers.”

✅ *Use this before refinement sessions to prep backlog in a structured way.*

Prompt 3: Story Point Estimation Helper

“Give me complexity factors, edge cases, and potential unknowns to consider when estimating this story:

Story: Enable OTP-based login for existing customers.”

✅ *Great for facilitating estimation discussions with the team.*

2. Sprint Planning

Prompt 4: Sprint Goal Suggestions

“Based on the following set of prioritized stories, propose 3 crisp, outcome-driven sprint goals.

Stories: [List stories with brief description].

Keep goals SMART, value-focused, and not just a task list.”

✅ *Helps POs communicate sprint vision clearly to the team & stakeholders.*

Prompt 5: Capacity & Risk Check

“Given the following team capacity (in story points) and list of candidate stories, help me check if this sprint plan is realistic. Highlight potential risks, dependencies, or stories that might spill over.”

✅ *Ideal for pre-planning checks before Sprint Planning meetings.*

3. Product Discovery & Prioritization

Prompt 6: Problem Statement Refinement

“Refine this problem statement to make it clear, specific, and measurable, focusing on user pain points and business value.
Problem: ‘Customers are abandoning the checkout flow.’”

 *Use before discovery workshops to sharpen problem framing.*

Prompt 7: Prioritization Using WSJF or RICE

“Apply the WSJF and RICE frameworks to prioritize the following list of features. Provide a clear ranking and reasoning for each.
Features: [List features with basic info like business value, effort, reach, etc.]”

 *Supports data-driven prioritization in roadmap discussions.*

Prompt 8: Persona-Based Story Generation

“Generate 3 user stories from the perspective of a [Persona: e.g., small business owner] who wants to use [Product Feature]. Keep stories realistic and aligned with user pain points.”

 *Helps enrich backlog with user-centric stories.*

4. Stakeholder Communication & Alignment

Prompt 9: Executive Summary of Sprint Outcomes

“Summarize the following sprint outcomes in an executive-friendly format. Include key achievements, metrics, and blockers. Keep it concise and suitable for a steering committee update.

Outcomes: [Paste sprint notes or key deliverables].”

 *Saves time in crafting crisp stakeholder updates.*

Prompt 10: Translate Technical Jargon to Business Language

“Explain the following technical implementation in clear business language for non-technical stakeholders.

[Paste technical explanation from dev team].”

 *Great for bridging tech-business gaps.*

Prompt 11: Stakeholder Workshop Agenda

“Draft a 60-minute stakeholder workshop agenda to align on feature prioritization for Q4 roadmap. Include time blocks, activities, and expected outcomes.”

 *Helps POs run structured, purposeful workshops.*

5. Sprint Reviews & Retrospectives

Prompt 12: Sprint Review Talking Points

“Create talking points for a sprint review that summarize the value delivered, demo key stories, and gather stakeholder feedback. Stories: [List of delivered stories].”

 *Helps Product Owners confidently lead reviews.*

Prompt 13: Retrospective Prompt Generator

“Generate thought-provoking retrospective questions focused on improving backlog quality, stakeholder alignment, and release predictability for the PO & team.”

 *Keeps retros fresh and targeted to PO’s role.*

6. Acceptance Criteria & Definition of Done

Prompt 14: Generate Acceptance Criteria in Gherkin

“For the following user story, write detailed acceptance criteria using the Given–When–Then format, covering positive and negative scenarios.

Story: As a user, I want to receive email notifications for every transaction.”

 *Ensures clarity for Dev & QA teams.*

Prompt 15: DoD Checklist Generator

“Based on this team’s context [e.g., web application, 2-week sprints, security checks], draft a Definition of Done checklist that includes development, testing, documentation, and review requirements.”

✓ *Standardizes quality gates across stories.*

7. Roadmap & Release Planning

Prompt 16: Product Roadmap Drafting

“Create a high-level 2-quarter product roadmap for the following initiatives. Include key themes, goals, and tentative releases. Initiatives: [List].”

✓ *Good starting point before fine-tuning with leadership.*

Prompt 17: Release Notes Creation

“Generate user-friendly release notes for version X.Y.Z based on these completed stories. Make them clear, non-technical, and suitable for end-users. [List stories or changelog].”

✓ *Saves hours before product launches.*

8. Continuous Discovery & Feedback Loops

Prompt 18: Feedback Synthesis

“Summarize the following user feedback into key themes, pain points, and opportunities. Suggest 3 potential features or improvements.

[Paste raw feedback].”

 *Turns noisy feedback into actionable insights.*

Prompt 19: Hypothesis-to-Story Conversion

“Convert this product hypothesis into 2–3 user stories and related acceptance criteria.

Hypothesis: If we add social login, new user signup conversion will increase by 15%.”

 *Great for structured experimentation.*

Prompt 20: Daily Summary Draft

“Summarize yesterday’s dev updates into a 3-line daily standup summary for the PO to share with stakeholders. Focus on progress, blockers, and priorities.”

✓ *Useful when working with distributed teams or async communication.*