International Journal of Computer Techniques–IJCT Volume 13 Issue 4, July-August - 2026

Open Access and Peer Review Journal ISSN 2394-2231 https://ijctjournal.org/

AI-Accelerated Engineering: How AI Can Transform the Product Owner Role

Toby Chembakassery Mathew, CPCU

Product and Strategy

New Jersey, United States

tobycm@gmail.com

Abstract: Generative AI is already changing software delivery in practice. Coding assistants, automated test generation, and AI-supported refactoring now let engineering teams move from intent to working software in a fraction of the time the same work took only a few years ago. The requirement definition side of delivery, owned by the Product Owner, has not kept pace. Problem discovery, business-rule analysis, prioritization, backlog elaboration, requirement capture, and outcome reporting are still performed largely by hand. The result is a velocity gap: an engineering function that can build faster than the product function can decide what to build. This paper argues that the Product Owner role is becoming the binding constraint on delivery throughput, and that the response is not to ask Product Owners to work harder but to apply AI to their own workflow. It presents a six-stage model of Product Owner work, identifies the dominant pain point and a corresponding AI intervention at each stage, and quantifies the resulting reallocation of effort. Under the model, roughly a third of a Product Owner's time shifts away from mechanical, low-value tasks toward the discovery and alignment work that AI cannot perform independently. Because AI also compresses the discovery cycle itself, the same shift lets a Product Owner carry more discovery in parallel, raising the throughput of validated work the delivery pipeline can sustain.

Keywords: Product Ownership, Generative AI, Agile Delivery, Product Discovery, Requirements Elicitation, Backlog Management, Software Delivery Velocity, Delivery Throughput, AI Augmentation., Product Owner.

1. Introduction

Generative AI has compressed the build step. Engineers now use coding assistants to scaffold services, generate tests, translate between languages, and refactor legacy code at a speed that was not available a few years ago; in one controlled study, developers completed a programming task roughly fifty-six percent faster with an AI assistant than without one [1]. Tasks that once occupied days of an engineer's week are increasingly measured in hours. The constraint on delivery is moving upstream, from building the capability to deciding what capability needs to be prioritized and how.

This is not an argument that Product Owners should work harder, nor that the role should be automated; it is an argument that the role should be re-tooled. A large share of Product Owner time goes to mechanical work: searching for a business rule buried in a document, pulling data to support a prioritization call, drafting backlog items, assembling a status report. None of this requires judgment, and all of it is work AI can now absorb. The opportunity is to move that time into the work that does require judgment and cannot be delegated to a model — deciding which problems are worth solving, validating them with real users and stakeholders, and keeping a team aligned on outcomes rather than output.

2. A Model of Product Owner Work

To reason about where AI helps, the role has to be decomposed into stages, each with a distinct kind of work, a distinct pain point, and a distinct intervention. This paper uses a six-stage model that follows a unit of product work from inception to outcome, consistent with the substance of the Product Owner role in agile practice [6], [7]:

1) **Discover and Validate:** identifying a problem worth solving and confirming it with users and stakeholders before committing to a solution.
2) **Analyze and Assess:** understanding how a proposed change interacts with existing business rules, systems, and constraints.
3) **Quantify and Prioritize:** using data to decide what to work on next and to justify the decision.
4) **Elaborate Backlog:** turning a validated, prioritized problem into well-formed, buildable backlog items.
5) **Enable and Align Team:** answering the team's questions, running refinement, and keeping stakeholders aligned.
6) **Measure Outcomes:** tracking whether delivered work produced the intended result, and reporting on it.

The stages are not strictly sequential. A Product Owner rarely sits in just one at a time; the more typical state is running discovery on one problem while supporting execution on another and prioritizing a third the same week. The order describes the path of a single piece of work, not the Product Owner's day, which is always several pieces at different stages at once.

3. Pain Points and AI Interventions by Stage

3.1 Discover and Validate

The pain point is that features are too often written before the underlying problem is validated. Under delivery pressure, a Product Owner moves quickly from a request to a specification, and stakeholder alignment arrives after build has started, when changing course is expensive. The upfront problem validation that would prevent this is the first thing sacrificed when time is short — yet confirming the problem before committing to a solution is long advocated in product practice [4], [5], and is precisely the discipline that mechanical workload crowds out.

The AI intervention is prototype-led discovery. Rather than describing a feature in prose and asking stakeholders to

imagine it, a Product Owner can turn a problem statement into a clickable prototype in hours, then work backward to the feature set. A working artifact surfaces disagreement earlier than a document does, because stakeholders react to something concrete instead of an abstraction. The same capability that builds the prototype also speeds authoring of the feature definition once the problem is confirmed, on the order of thirty to forty percent faster. This is one of two stages where AI should increase the Product Owner's share of time, from roughly a tenth of effort to a little over a quarter in the model. The time saved on mechanical work downstream is reinvested here, in validating the problem before the team builds against it.

3.2 Analyze and Assess

The pain point is hidden business rules. In any organization with a non-trivial history, the rules that govern how the product behaves are scattered across documents, prior backlog items, source code, and the memories of long-tenured staff. A Product Owner assessing the impact of a change spends a substantial share of time simply reconstructing what the current rules are and where a change will ripple — pure overhead, necessary but adding no value of its own.

The AI intervention is on-demand access to those rules. A retrieval system indexed over the organization's content repositories, backlog history, and codebase lets a Product Owner ask a question in natural language and receive an answer grounded in the actual artifacts, with references back to the source. The rule lookup that used to require knowing whom to ask and which document to open becomes a query, cutting impact-analysis time by roughly half. The model reduces this stage from about thirty percent to sixteen; analysis remains the Product Owner's responsibility, but reconstructing its inputs does not.

3.3 Quantify and Prioritize

The pain point is access to data. Sound prioritization rests on evidence — key indicators, the size of the affected population, the cost of the current workaround — and that evidence is frequently locked behind a request to an analytics team or a manual pull. The friction is high enough that prioritization often proceeds on intuition, because assembling the data would cost more than the decision seems to justify. Each manual pull can consume several hours per feature.

The AI intervention is data-led prioritization through direct, conversational access to the organization's analytics and business-intelligence layer. When a Product Owner can ask for the relevant figures and receive them in the flow of work, evidence stops being a luxury reserved for the largest decisions and becomes a routine input to ordinary ones. The share of time spent here falls modestly in the model, from fifteen percent to ten — not because prioritization matters less, but because the labor of feeding it shrinks.

3.4 Elaborate Backlog

The pain point is that backlog elaboration is slow and error-prone. Turning a validated problem into a complete set of well-formed items — with acceptance criteria, edge cases, and non-functional considerations — is painstaking by hand, and missed scenarios resurface later as defects or mid-sprint surprises, far more expensive than catching them during elaboration.

The AI intervention is assisted backlog authoring grounded in personas aligned to the Product Owner's product area. A model primed with the relevant personas, prior items, and domain context can draft items and propose the edge cases a human is prone to overlook, leaving the Product Owner to review, correct, and decide rather than write from a blank page. The model moves this from a quarter of Product Owner time to fifteen percent; the model drafts, the Product Owner ratifies what is true and in scope.

3.5 Enable and Align Team

Two pain points converge here. Requirement elicitation is conversational, and every interview, workshop, and refinement session leaves a clerical tail — notes to capture, requirements to write up, decisions to record, and action items to assign and follow up on. Alongside it sits repetitive clarification: during delivery a Product Owner fields a steady stream of the same questions, and refinement is consumed by re-explaining what was already specified somewhere. Both are interrupt-driven and pull attention from the forward-looking parts of the role.

The AI intervention is an on-demand team assistant grounded in the team's own backlog, decisions, and documentation, paired with automatic meeting capture. It answers routine requirement questions in real time without the Product Owner in the loop, and it transcribes conversations, extracts decisions and action items, and drafts refinement summaries, demo scripts, and stakeholder updates that would otherwise be assembled by hand; detecting and summarizing action items from meeting dialogue is a well-studied capability [9]. Two to four hours per sprint typically come back. This is the second stage where AI should increase the Product Owner's share of time, from fifteen percent to a little over a quarter: the assistant takes the repetitive question-and-answer and the documentation, and the reclaimed time goes into alignment for getting stakeholders to a shared understanding and enabling the team on the reasoning behind the work.

3.6 Measure Outcomes

The pain point is that status reflects activity, not progress. Conventional reporting shows what was done — items closed, points burned — rather than whether the work moved the outcome it was meant to move, and producing even that activity-based report is manual effort that falls on the Product Owner.

The AI intervention is outcome visibility through integration with the team's work-tracking and delivery systems, generating outcome-oriented indicators automatically. When reporting is generated rather than hand-assembled, the Product Owner stops producing status and the conversation shifts from output to outcome. This is the smallest stage by time and stays small, from five percent to three in the model. The value is less the hours saved than

the shift in focus to whether the work is moving the business objectives it was meant to move.

4. The Reallocation of Effort

Taken together, the six interventions change the shape of the role, not just its speed. Four of the six stages — analysis, prioritization, elaboration, and measurement — are mechanical work that AI compresses, so the share of time they consume falls. This frees capacity for Product Owners to create more impact, aligned with the organization's objectives. Table 1 summarizes the reallocation.

Table 1: Reallocation of Product Owner effort under AI augmentation (share of a standard working week)

Stage	Current Allocation	AI-Enabled Allocation	Shift in Share
Discover and Validate	10%	28%	+18%
Analyze and Assess	30%	16%	−14%
Quantify and Prioritize	15%	10%	−5%
Elaborate Backlog	25%	15%	−10%
Enable and Align Team	15%	28%	+13%
Measure Outcomes	5%	3%	−2%

The downward shifts total thirty-one percent of Product Owner time pulled out of mechanical work; the upward shifts move the same thirty-one percent into discovery and alignment. For a standard week, that is approximately twelve hours — not time saved but time moved, redirected from work AI does well to work only a person can do. A large part of it comes from one recognizable source: the elimination of manual documentation around requirement elicitation, the notes, write-ups, and action-item tracking that every stakeholder conversation used to generate by hand.

5. From Reclaimed Time to Throughput

The table counts time moved, but the effect is larger, because AI also compresses the discovery cycle itself. A problem becomes a testable prototype in hours, and a validated problem becomes a draft backlog quickly. A Product Owner gets more hours for discovery and a shorter cycle to spend them on, which means discovery can run on several items at once instead of one at a time — more product threads and more projects in parallel.

The gain is throughput, not just personal speed. A delivery pipeline can only move as fast as its slowest stage [2]; when the Product Owner was single-threaded, the amount of validated, well-defined work was capped by how fast one person could work through it in sequence, and adding engineering capacity downstream did not help, since more people do not make a constrained pipeline move faster [3]. Running discovery in parallel lifts that cap: more validated work reaches engineering each week. Two limits apply. The added work has to be more discovery, not the mechanical work AI just removed, or the bottleneck returns at higher volume.

6. Implementation Considerations

The model assumes AI capability and integration that is available but not automatic. Several conditions determine whether the reallocation is realized in practice or remains a slide.

First, the interventions depend on access to the organization's own information. On-demand rule lookup, data-led prioritization, and grounded authoring derive their value entirely from the AI being connected to the real artifacts — content repositories, the analytics layer, the work-tracking system, the codebase. A model without that grounding produces fluent, confident, frequently wrong output; grounding generation in retrieved source material is what mitigates this [8]. Connecting these sources, and governing what the model may see, is the precondition for everything above.

Second, the Product Owner remains accountable for every output the AI touches. The model drafts a backlog item; the Product Owner decides whether it is correct and in scope. AI output is a first draft to be reviewed, never an answer to be shipped.

Third, the reclaimed time must be defended. The reallocation produces better outcomes only if the twelve hours actually move to discovery and alignment. The default response to a more efficient Product Owner is to assign more scope, which converts the saving back into mechanical work at higher volume and forfeits the benefit. The productive use of the capacity is more discovery run in parallel; the failure mode is absorbing more mechanical and documentation load, which simply recreates the bottleneck.

Finally, the model is a direction, not a precise prescription. The specific percentages reflect one view of the role and will vary by team, domain, and tooling maturity. The durable claim is directional: AI compresses the mechanical stages and frees time for the human.

7. Conclusion

The acceleration of engineering by AI has relocated the bottleneck in software delivery. An engineering function that ships in days is throttled by a product function that still validates and defines in weeks, and no amount of additional engineering capacity resolves the mismatch. The constraint is upstream.

The answer is not to ask the Product Owner to absorb more mechanical work, nor to hand the role to a model. It is

to give the Product Owner the same tools that reshaped engineering — for rule lookup, data access, backlog drafting, and status reporting and to put the freed time into the parts of the role that need judgment. The model puts that shift at about a third of a Product Owner's week, roughly twelve hours, moving from mechanical work into discovery, validation, and alignment. That keeps the definition side of delivery in step with the execution side, and it lets a Product Owner spend more of the week on work that moves the organization's objectives rather than on overhead.

References

[1] Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. arXiv preprint arXiv:2302.06590.

[2] Goldratt, E. M., & Cox, J. (1984). The Goal: A Process of Ongoing Improvement. Great Barrington, MA: North River Press.

[3] Brooks, F. P. (1995). The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition. Boston, MA: Addison-Wesley.

[4] Cagan, M. (2018). Inspired: How to Create Tech Products Customers Love, 2nd Edition. Hoboken, NJ: John Wiley & Sons.

[5] Torres, T. (2021). Continuous Discovery Habits: Discover Products that Create Customer Value and Business Value. Product Talk LLC.

[6] Schwaber, K., & Sutherland, J. (2020). The Scrum Guide: The Definitive Guide to Scrum — The Rules of the Game.

[7] Pichler, R. (2010). Agile Product Management with Scrum: Creating Products that Customers Love. Boston, MA: Addison-Wesley.

[8] Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems, 33. arXiv:2005.11401.

[9] Purver, M., Dowding, J., Niekrasz, J., Ehlen, P., Noorbaloochi, S., & Peters, S. (2007). Detecting and Summarizing Action Items in Multi-Party Dialogue. In Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pp. 18–25. Association for Computational Linguistics.

Author Profile

Toby Chembakassery Mathew is responsible for Product and Strategy within the Specialty division of a commercial insurance carrier. He holds the Chartered Property Casualty Underwriter (CPCU) designation through The Institutes. His work focuses on applying artificial intelligence to product ownership, underwriting workflow, and the modernization of complex, regulated software products, and he writes on the practical adoption of AI in enterprise delivery.