

From PDLC to AI-DLC: Product management in the age of AI-driven development

Five shifts product leaders need to plan for as AI moves from a coding assistant to the engine of the product development lifecycle.

Product Guide Research • 29 October 2026 • productguide.io/research/product-management-pdlc-to-ai-dlc/

Executive summary

Software delivery is being rebuilt around AI. In 2025 AWS published the AI-Driven Development Life Cycle (AI-DLC), a method in which AI plans, proposes and builds while people set intent, make decisions and approve each step ^[1]. McKinsey argues that AI will bring five fundamental changes to the software product development life cycle (PDLC) ^[2], and Google's DORA research found that 90% of technology professionals now use AI in their work ^[3].

For product managers, the important change is not that engineering gets faster. It is that the constraint moves. When code is cheap to produce, the scarce resources become clear intent, sound judgement about what to build, and justified trust in what AI produces. The evidence on speed is mixed: in one controlled experiment developers completed a coding task 55.8% faster with an AI assistant ^[7], while a 2025 trial with experienced open-source developers found they were 19% slower, even though they believed they were faster ^[6].

This paper sets out five shifts and what product leaders should do about each: the move from phases to fast loops; product judgement becoming the bottleneck; the specification becoming the product; trust and governance designed in from the start; and an operating model in which AI amplifies existing strengths and weaknesses. It closes with a practical 90-day plan.

90%

of technology professionals use AI at work ^[3]

3%

of developers highly trust AI output ^[5]

19%

slower in a trial where developers expected to be 24% faster ^[6]

90%

of enterprise engineers forecast to use AI code assistants by 2028 ^[10]

Inside

1. From phases to loops
2. The bottleneck moves to product judgement
3. The specification becomes the product
4. Trust, quality and governance by design
5. AI amplifies the operating model you already have
6. A 90-day plan for product leaders

WHO IT IS FOR Heads of product, product directors and CPOs deciding how their teams will work as AI takes on more of the development lifecycle, and the engineering and design leaders who partner with them.

01 From phases to loops

What is the AI-driven development lifecycle (AI-DLC)?

The AI-DLC is a way of building software in which AI agents plan, propose and produce the work, while people provide intent, make decisions and approve each step. AWS describes three phases, Inception, Construction and Operations, run in short cycles it calls bolts, measured in hours or days rather than the weeks of a sprint ^[1].

The traditional PDLC moves work through handoffs: from discovery to requirements, design, build, test and release. Every handoff is a queue. The AI-DLC removes many of them. In AWS's Inception phase, AI turns business intent into requirements, stories and units of work, and the whole team validates its questions and proposals together in a session AWS calls Mob Elaboration. In Construction, AI proposes the architecture, domain models, code and tests, and the team settles technical decisions in Mob Construction. Epics give way to units of work, and sprints to bolts ^[1].

Key takeaways

- Map where your current lifecycle waits on handoffs. Those queues are where AI shortens cycle time most.
- Pilot short cycles on one contained product area before changing rituals everywhere.
- Keep a human approval at every phase boundary: AI proposes, people decide.

McKinsey makes a similar case from the product side. AI can join up fragmented customer feedback and product usage data and feed it into the development cycle, so that products are tied to customer value from the outset ^[2].

The common thread is a change of rhythm: from a sequence of phases to a fast loop of intent, generation, review and learning. Rituals designed for two-week sprints, such as long refinement sessions and big-batch planning, start to look like overhead.

02 The bottleneck moves to product judgement

Does AI make product managers more or less important?

More important, but for different work. When AI makes building faster, deciding what to build and why becomes the constraint. Andrew Ng has argued that product management is now the bottleneck in AI-assisted teams, and described one team proposing one product manager for every half an engineer, a ratio he called extreme but an early signal ^[9].

Teams have typically run with one product manager to around six engineers ^[9]. If engineering output rises sharply while discovery, prioritisation and stakeholder alignment move at the same pace, product decisions become the queue.

The speed gains are real but uneven. In a 2022 randomised experiment, developers using GitHub Copilot finished a defined programming task 55.8% faster than a control group ^[7]. Yet in a 2025 randomised trial of 16 experienced developers working on 246 real tasks in mature open-source

projects, METR found they took 19% longer with AI tools. Beforehand they expected to be 24% faster, and afterwards they believed they had been 20% faster ^[6]. Measure outcomes, not impressions: self-reported speed is not evidence.

Product work is changing too. Low-judgement tasks, such as writing up notes, drafting stories and grooming backlogs, are increasingly automated. What remains is the hard part: choosing problems, making trade-offs, saying no, and judging whether something is valuable, usable, feasible and viable.

Key takeaways

- Spend product time on discovery and decision quality, not documentation volume.
- Track cycle time from idea to validated outcome, not tickets closed.
- Revisit team ratios on purpose so engineering capacity never outruns product direction.

03 The specification becomes the product

What does a product manager write in an AI-DLC?

Specifications that both people and AI agents can act on: clear intent, users, constraints, non-goals, acceptance criteria and examples. In spec-driven development the specification is the primary artefact and code is one expression of it, an approach GitHub open-sourced as Spec Kit in September 2025 ^[8].

AI agents do what they are asked, quickly. Vague intent now produces plausible but wrong software faster than ever. That raises the value of precise problem statements, explicit non-goals, edge cases and testable acceptance criteria. These are the craft skills of a good requirements document, now written for a machine reader as well as a human one.

GitHub's Spec Kit, released with support for agents including GitHub Copilot, Claude Code and Gemini CLI, encodes the workflow: write the specification, plan, break the plan into tasks, then implement ^[8].

Key takeaways

- Standardise a specification template: intent, users, constraints, non-goals and acceptance criteria.
- Keep specifications in version control alongside the code they produce.
- Prototype early to learn; specify precisely to build.

AWS's AI-DLC likewise starts every piece of work by turning intent into requirements and units of work before any code is generated ^[1].

Prototypes and specifications reinforce each other. A prototype tests desirability with real users early; the specification captures what was learned in a form agents can build from. Specifications stop being handoff documents. They become living inputs to an automated system, versioned next to the code and reviewed with the same care.

04 Trust, quality and governance by design

How should product teams govern AI-generated work?

By building review, evaluation and accountability into the lifecycle instead of adding them at the end. Developers use AI widely but trust it little: in Stack Overflow's 2025 survey, 46% distrust the accuracy of AI tools against 33% who trust it, and only 3% highly trust the output ^[5].

Low trust is rational. AI output can look finished while hiding defects, security issues or a misunderstood requirement. DORA's 2025 research highlights rework as an increasingly important signal of delivery health in AI-assisted teams ^[3]. Speed that creates rework is not speed.

Governance is also becoming a legal duty. Under the EU AI Act, obligations for general-purpose AI models have applied since 2 August 2025, with obligations for high-risk systems phasing in after

that ^[12]. The NIST AI Risk Management Framework offers a voluntary structure built on four functions: govern, map, measure and manage ^[11].

For product teams this means concrete practice: approval gates at each phase, evaluation sets that test AI-built features against expected behaviour, a named owner for every AI-generated change, and requirements that state risk level and regulatory context from the start.

Key takeaways

- Define who approves what at each stage, and record it.
- Treat evaluations and acceptance tests as part of the specification.
- Classify features by risk early, including regulatory exposure.

05 AI amplifies the operating model you already have

What operating model helps product organisations benefit from AI?

One with strong foundations. DORA's 2025 research, drawing on nearly 5,000 technology professionals, found that AI acts as an amplifier: it magnifies the strengths of high-performing organisations and the dysfunctions of struggling ones ^[3].

DORA's AI Capabilities Model names seven capabilities that increase the benefit of AI, including a clear and communicated AI stance, healthy data ecosystems, a strong user-centric focus, strong version control, working in small batches and quality internal platforms ^[4]. DORA reports that teams with a strong user focus become more effective when they adopt AI, while teams with a weak one can see performance decline ^[4]. That is as much a product management finding as an engineering one.

Adoption is moving fast. Gartner forecasts that 90% of enterprise software engineers will use AI code assistants by 2028, up from less than 14% in early 2024 ^[10]. Product organisations that wait for the tools to settle will find their engineering partners have already changed how they work.

The product manager's skill profile shifts accordingly: writing for agents, prototyping with AI tools, designing evaluations, reading data directly and coaching teams through new rituals. The durable skills of customer insight, strategy and judgement matter more, not less.

Key takeaways

- Publish a clear AI stance: what is encouraged, what is not, and why.
- Fix the foundations first: data quality, small batches, version control and platforms.
- Build product skills in prototyping, specification and evaluation.

A 90-day plan for product leaders

WEEKS 1-4

Baseline

Measure idea-to-outcome cycle time, rework and current AI use. Publish an AI stance for product and engineering.

WEEKS 5-8

Pilot

Run one product area in AI-DLC style: short cycles, joint elaboration sessions with AI, and specification-first work with approval gates.

WEEKS 9-12

Scale what worked

Standardise the specification template and gates, train product managers in prototyping and evaluation, and revisit team shape.

Questions and answers

What is the difference between the PDLC and the AI-DLC?

The product development lifecycle (PDL) moves work through sequential phases and handoffs between specialist roles. The AI-driven development lifecycle (AI-DLC) runs short loops in which AI proposes requirements, designs, code and tests, and people validate and approve them, so handoffs and waiting time shrink ^[1].

What is a bolt in the AI-DLC?

A bolt is the AI-DLC's unit of delivery cadence. AWS describes bolts as shorter, more intense cycles measured in hours or days, replacing sprints measured in weeks ^[1].

Questions and answers, continued

What is spec-driven development?

Spec-driven development treats the written specification as the primary artefact and generated code as one expression of it. Teams write the specification, plan, break it into tasks, and let AI agents implement each task. GitHub open-sourced a toolkit for this, Spec Kit, in September 2025 ^[8].

Will AI replace product managers?

The evidence points the other way. As AI speeds up building, deciding what to build becomes the constraint, which raises the value of product judgement ^[9]. Administrative product tasks are being automated; discovery, prioritisation and trade-offs are not.

How should we measure AI's impact on product development?

Measure delivered outcomes rather than perceived speed. Useful measures are cycle time from idea to validated outcome, rework rate and customer impact. A 2025 trial found developers felt 20% faster while actually being 19% slower ^[6], and DORA flags rework as an important signal ^[3].

References

1. Raja SP, Amazon Web Services. AI-Driven Development Life Cycle: Reimagining Software Engineering. AWS DevOps & Developer Productivity Blog, July 2025. <https://aws.amazon.com/blogs/devops/ai-driven-development-life-cycle/>
2. McKinsey & Company. How an AI-enabled software product development life cycle will fuel innovation. McKinsey & Company, February 2025. <https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/how-an-ai-enabled-software-product-development-life-cycle-will-fuel-innovation>
3. DORA, Google Cloud. State of AI-assisted Software Development 2025. DORA, September 2025. <https://dora.dev/research/ai/>
4. DORA, Google Cloud. Introducing the DORA AI Capabilities Model: 7 keys to succeeding in AI-assisted software development. Google Research, 2025. <https://research.google/pubs/introducing-the-dora-ai-capabilities-model-7-keys-to-succeeding-in-ai-assisted-software-development/>
5. Stack Overflow. 2025 Developer Survey: AI. Stack Overflow, July 2025. <https://survey.stackoverflow.co/2025/ai>
6. METR. Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. METR, July 2025. <https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/>
7. S. Peng, E. Kalliamvakou, P. Cihon and M. Demirer. The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. arXiv:2302.06590, February 2023. <https://arxiv.org/abs/2302.06590>
8. GitHub. Spec Kit: toolkit for spec-driven development. GitHub (open source), September 2025. <https://github.com/github/spec-kit>
9. Andrew Ng. Remarks on product management as the bottleneck in AI-assisted teams, as reported. Business Insider via AOL, 2025. <https://www.aol.com/andrew-ng-says-real-bottleneck-051543175.html>
10. Gartner. Magic Quadrant for AI Code Assistants (forecast as published in the Gartner webinar summary). Gartner, 2025. <https://www.gartner.com/en/webinar/674243>
11. National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1. NIST, January 2023. <https://www.nist.gov/itl/ai-risk-management-framework>
12. European Commission. AI Act: regulatory framework for artificial intelligence. European Commission, 2024–2025. <https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai>

About this paper

This paper draws on publicly available sources published between 2023 and 2025: vendor methodologies, independent research and industry surveys. Figures are reported as their sources published them. Forecasts and vendor-reported results are identified as such. The paper is independent and not sponsored.

About Product Guide

Free product management templates, frameworks and guides, organised by the work you're doing, with a community of product people around them. Read more and subscribe at productguide.io/join.html.