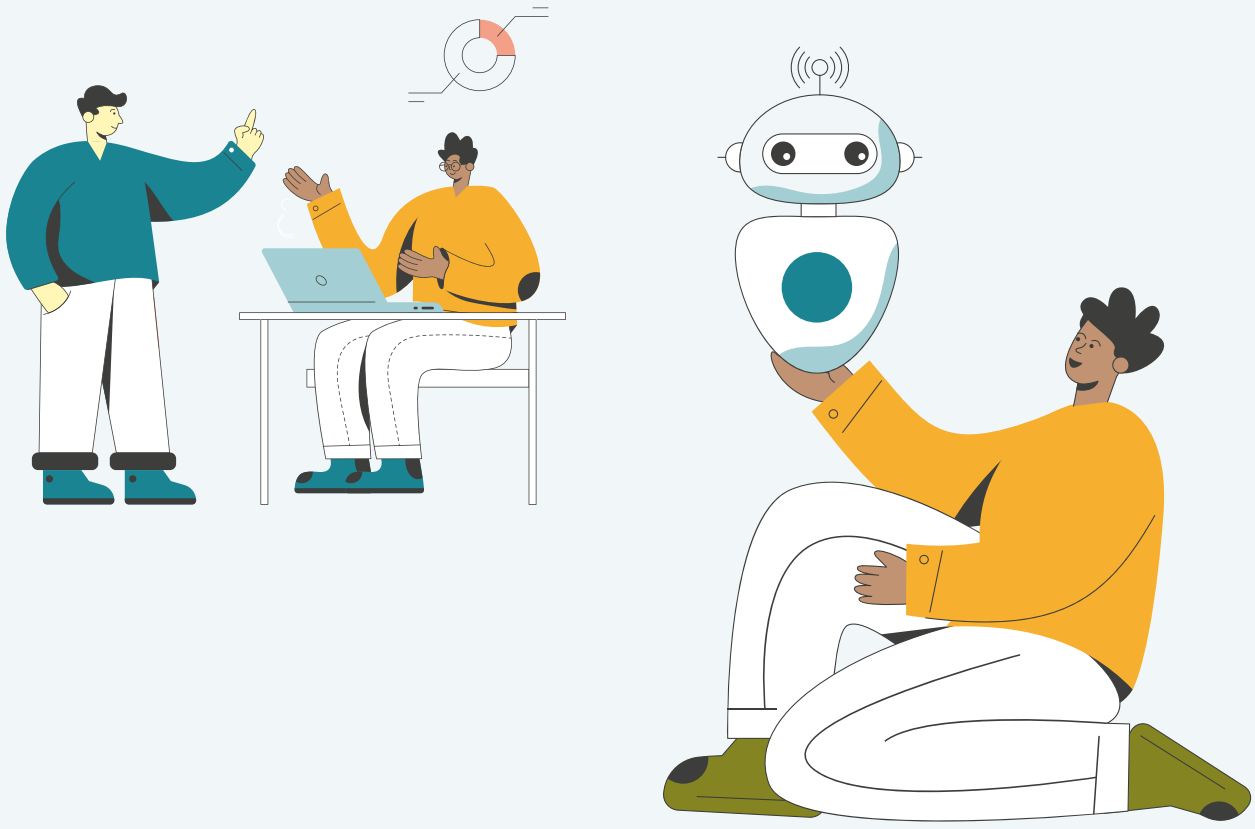
WHITE PAPER | NOVEMBER 2026

The Rise of the Product Builder

How AI is reshaping product management, product development and the organisations behind them.

From customer insight to proven value.

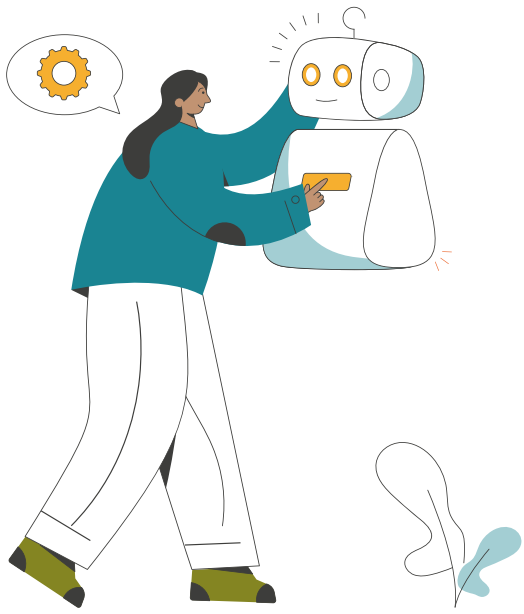


tarigo.co.uk

Contents

A practical perspective on how AI can connect customer understanding, commercial judgement and technical execution.

Executive summary		3
1.	Why the Product Builder is emerging	4
2.	Defining the Product Builder	5
3.	How product management and development change	6
4.	Faster building is not the same as faster value	8
5.	The organisational challenge: connecting the work	9
6.	A shared model from problem to proven value	10
7.	What this looks like in practice	12
8.	The boundary between a prototype and a product	13
9.	What changes for product leadership	14
10.	Building the capability, not just buying the tools	15
11.	A practical route to adoption	16
12.	Build the capability to create value	18



Executive summary

AI is changing more than the speed at which products are developed. It is changing who can build, how teams work together and where human judgement creates the greatest value. The emerging Product Builder represents this shift: a capability that connects customer understanding, commercial thinking, design and technical execution.

The boundaries between product management and product development are becoming more 2026 CPO Insights Report, drawing on interviews and surveys involving more than 1,500 Chief Product Officers, describes an emerging hybrid operator capable of moving across discovery, prototyping, development and delivery with fewer functional handoffs. It calls this operator the Product Builder.

The significance is not that every product manager will become a software engineer, or that every engineer will become a product manager. It is that organisations have an opportunity to reduce the distance between understanding a problem and testing a solution.

However, generating more software does not automatically create more value. A working prototype does not establish customer demand. Faster development does not guarantee adoption. Broader individual capability does not remove the need for specialist expertise, technical assurance or clear accountability.

This paper argues that the Product Builder should be understood at three connected levels: as an individual who works across traditional boundaries; as a capability distributed across a team; and as an organisational approach that connects people, technology and decisions around shared outcomes.

The opportunity is therefore larger than a change in job title. It is a redesign of how products move from customer problems to measurable results.

Realising that opportunity requires more than access to AI tools. Organisations need a shared way of working, clearer decision rights, stronger connections between discovery and delivery, and practical development of both product and technical judgement.

INDIVIDUAL	TEAM	ORGANISATION
Broader capability	**Complementary expertise**	**Connected decisions**

The goal is not simply to build more, with fewer people. It is to become better at deciding what deserves to be built, proving its value and delivering it responsibly.

Why the Product Builder is emerging

AI is reducing the effort required for some forms of creation, making it possible to reconsider how product work is divided between people and functions.

80%+

CPOs reporting materially faster development cycles. [1]

90%+

Technology professionals reporting AI use at work. [2]

Self-reported signals from different research populations not directly comparable productivity measures.

The *2026 CPO Insights Report* presents a clear signal from product leadership: more than 80% of the CPOs consulted said AI had materially accelerated their product development cycles. The report highlights prototyping, research, engineering and testing as areas of change. It also describes increased reliance on generalist product managers, rising from 15% to 28% in its year-on-year comparison. These are reported experiences and organisational signals, rather than controlled measurements of productivity. [1]

Google's 2025 DORA research provides a complementary perspective. Among nearly 5,000 technology professionals surveyed, 90% reported using AI at work and more than 80% believed it had improved their productivity. The research also emphasises that organisational benefits depend on the environment in which those tools are used. [2]

The implication is not simply that established tasks can be completed more quickly. It is that the distribution of those tasks can be reconsidered.

A product manager may contribute a working prototype rather than only a written description. An engineer may explore alternative customer workflows rather than wait for a fully specified solution. A designer may make an interaction testable without requiring a separate implementation cycle.

In the model proposed here, these contributions bring different disciplines into the same conversation earlier. Ideas become tangible while they are still open to challenge.

This does not make collaboration a new invention, nor suggest that effective product teams previously operated as assembly lines. It creates an opportunity to extend collaborative working: fewer ideas need to remain abstract while a team waits for the next specialist to become available.

The important question is therefore not: *How much faster can each function complete its existing tasks?*

What becomes possible when the distance between thinking, making and learning is reduced?

Defining the Product Builder

The Product Builder is not simply a product manager with a coding tool. It is the capability to connect customer problems, business intent and technical possibility and turn them into products that create measurable value.

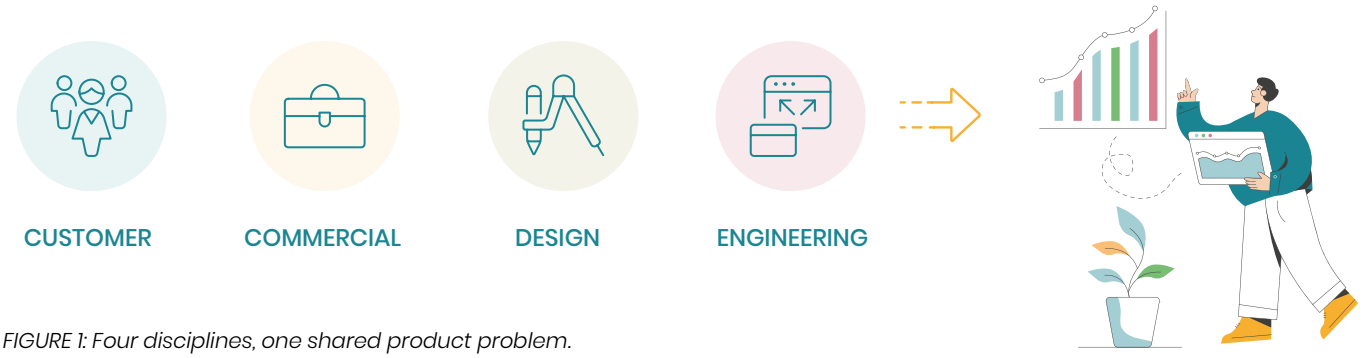


FIGURE 1: Four disciplines, one shared product problem.

For this paper, we define Product Builder capability as:

The ability to understand a meaningful problem, shape and test possible solutions, coordinate human and AI contributions, and take responsibility for learning whether the resulting product creates value.

This definition places judgement at the centre. Producing an interface or generating code is part of the capability, not its defining purpose.

A Product Builder must be able to distinguish between a request and a problem, an attractive idea and a viable opportunity, a convincing demonstration and a dependable product.

The capability can exist at different levels.

At the individual level, a professional works beyond the traditional boundaries of their discipline. A product manager develops technical fluency; an engineer develops stronger customer and commercial judgement.

At the team level, people combine complementary strengths around a shared problem. They do not all become equally efficient at everything, but they understand enough of one another's work to make better decisions together.

At the organisational level, leadership provides the structures, platforms, standards and incentives that allow this way of working to be repeated safely.

Products That Count predicts that the traditional product manager role will be replaced by the Product Builder by 2030. This paper does not adopt that prediction as an established outcome. The underlying organisational change matters more than whether a particular title survives. [1]

Some organisations may introduce Product Builder roles. Others may retain their existing titles while changing responsibilities and ways of working.

How product management and development change

The convergence runs in both directions: product professionals move closer to building, while development professionals move closer to deciding what is worth building.

A credible Product Builder model must change both sides of the relationship. Asking product managers to prototype while engineers continue receiving predetermined requirements is not meaningful convergence.

Neither is asking engineers to take greater ownership without giving them access to customers, business context or decision-making.

Product management moves closer to tangible evidence

Product managers use prototypes, workflow simulations and small experiments to make assumptions testable.

Their contribution becomes less dependent on describing an imagined solution in exhaustive detail before anybody can experience it. They can help a team explore what the solution might actually feel like, where the difficult decisions lie and which assumptions deserve attention.

However, a prototype should remain an invitation to challenge, not an instruction to implement. The product manager's responsibility is not to persuade engineering to reproduce an AI-generated demonstration. It is to help establish what has been learned, what remains uncertain and what would justify further investment.

Strategic thinking, customer understanding and commercial judgement remain central. They are exercised closer to the act of making.

Development moves closer to the customer problem

Engineers should participate in framing opportunities and exploring alternatives, not only assessing whether a proposed feature is feasible. Their technical understanding can change the product direction. They may identify a simpler way to achieve the desired outcome, expose a hidden dependency or show that an apparently modest feature carries substantial operational cost.

The development role also needs to regain responsibility beyond initial code production. DORA's March 2026 analysis highlights the trade-off between faster generation and the work required to audit, verify and integrate outputs. This reinforces the distinction between creating an initial implementation and making it dependable. [3]

The Product Builder model should therefore increase the visibility of engineering judgement, not treat it as a residual activity after AI has 'done the building'.

How product management and development change

Design and research remain essential

Broader participation in design does not make every contributor an expert in usability, accessibility or research. The proposed shift is from specialists being the only people permitted to contribute, towards specialists helping the wider team make better decisions while retaining depth in their discipline.

A useful principle is broader contribution, with explicit responsibility. More people can explore a solution; the team must still know who is accountable for evaluating its quality.

Dimension	A handoff-dependent approach	A Product Builder approach
Starting point	A feature moves between functions	A team works on a shared customer problem
Product contribution	Describe the intended solution	Frame, prototype, test and make trade-offs
Engineering contribution	Implement an agreed specification	Shape the opportunity and deliver a dependable solution
Evidence	Reviewed at scheduled checkpoints	Used continuously to guide decisions
Accountability	Completion of functional outputs	Shared outcomes, with named decision owners

These are contrasting operating patterns, not a claim that all existing teams follow the first column.

Faster building is not the same as faster value

The central risk is mistaking production speed for progress. The Product Builder must shorten the journey to evidence and outcomes, not merely the journey to a demonstration.

The CPO report argues that constraints are shifting towards go-to-market execution, customer adoption and organisational alignment. In its comparison of internal challenges, the share identifying speed to market as their leading concern rises from 14% in 2025 to 22% in 2026. This indicates changing leadership priorities; it does not establish that engineering constraints have disappeared in every organisation.[1]

The distinction matters.

A team might produce a feature quickly but struggle to explain its value. Customers might understand the benefit but find migration difficult. A product might attract users while creating an unsustainable support burden.

In each case, software production is only one part of the value equation.

Evidence on development productivity also argues against a universal improvement. METR's 2025 randomised study involved 16 experienced open-source developers completing 246 tasks in familiar repositories. With the early-2025 AI tools studied, tasks took 19% longer when AI was permitted. This was a specific setting, not a finding about all development work.[4]

In February 2026, METR reported that newer tools likely provided greater acceleration, but selection and measurement problems prevented a reliable estimate of the improvement.[5]

The practical conclusion is to measure the actual workflow rather than import a headline productivity multiplier.

For Product Builders, speed should answer three different questions:

How quickly can we learn whether the opportunity is real? How quickly can we deliver a dependable solution? How quickly can customers achieve the intended benefit?

These questions also change how go-to-market work is approached. Positioning, onboarding, pricing and customer readiness should influence product decisions from the beginning, rather than becoming the next queue after development.

A feature shipped but not adopted is not a completed value-creation process.

The organisational challenge: connecting the work

Giving every function access to AI does not create a coherent product organisation. The opportunity lies in connecting their work around common evidence, goals and decisions.



FIGURE 2: From separate AI-generated outputs to a connected body of product work.

Consider a team in which a product manager generates requirements, a designer generates interfaces, an engineer generates code and a marketer generates launch messaging.

Each person may produce their output more quickly. Yet the outputs may still be based on different assumptions about the customer, the problem and the intended value.

The result could be faster activity without stronger alignment.

We describe the difference between isolated AI use and a connected approach as the orchestration gap.

Closing this gap does not require a complex network of autonomous agents. It begins with shared context: the customer problem, supporting evidence, desired outcome, known constraints and decisions already made.

DORA central finding that AI amplifies existing organisational strengths and weaknesses supports this emphasis on the wider working environment. Its research does not suggest that tool adoption alone guarantees better organisational performance.[2]

In a Product Builder organisation, people and AI tools should contribute to a connected body of work rather than create competing versions of reality.

A changed customer insight should be visible in the solution being explored. A technical limitation should influence the promise made to customers. A failed experiment should affect investment decisions, not disappear into a presentation archive.

This requires traceability, but not necessarily more documentation. The useful record is the one that explains what the team believes, why it believes it and what would change its mind.

The practical foundations are also important. DORA's AI Capabilities Model identifies factors including clear organisational guidance on AI use, healthy data, access to relevant internal context, version control, small batches, user focus and quality internal platforms. These provide a useful complement to the shared product workflow proposed here.[6]

The organisational goal is not to maximise the number of AI-generated outputs. It is to improve the coherence and quality of the decisions those outputs support.

A shared model from problem to proven value

Product Builders need a common way of working that connects discovery, delivery and adoption without turning them into disconnected stages.

An organisation does not need to discard its existing product lifecycle. It needs to make the decisions within that lifecycle more explicit and connect them through evidence.

The following model describes five questions that a Product Builder team should be able to answer.

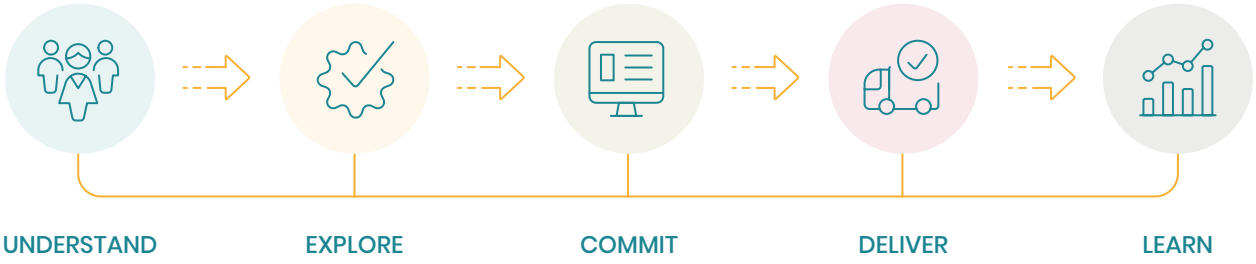


FIGURE 3: Evidence flows back through the journey; these are not one-way stage gates.

Core question	Work undertaken together	Evidence needed to progress
Which problem deserves attention?	Understand the customer, strategic relevance and commercial significance	Evidence of a meaningful problem and a clear reason to act
What might solve it?	Explore customer workflows, alternative approaches and technical possibilities	Testable hypotheses, options and known constraints
What should we commit to?	Test the riskiest assumptions and choose a proportionate next step	Customer evidence, feasibility findings and explicit trade-offs
Can we deliver it responsibly?	Build, verify, integrate and prepare the product for real use	Evidence of quality, operational readiness and appropriate assurance
Is it creating value?	Support adoption, observe behaviour and evaluate results	Customer outcomes, commercial performance and evidence for the next decision

This is a learning loop, not a waterfall. Findings from delivery may change the solution. Adoption problems may reveal that the original problem was misunderstood. A technical experiment may show that the team should stop before conducting a larger customer trial.

At each point, AI can contribute analysis, alternatives or implementation work. Humans remain accountable for the decisions made using those contributions.

Three disciplines that keep the model honest

Three disciplines make the model practical.

1

Distinguish assumptions from evidence

A plausible customer persona is not a customer interview. A generated market estimate is not verified market data. A positive reaction to a prototype is not proof of willingness to pay.

The team should preserve those distinctions in its working materials.

2

Make decision rights explicit

Shared ownership should not mean that everybody is responsible and nobody can decide.

The team needs named owners for product direction, technical acceptance, risk approval and release decisions. The allocation will vary by context, but it must be understood before disagreements arise.

3

Match assurance to consequence

A disposable prototype using dummy data and a production service handling sensitive information should not have identical approval paths. The aim is to allow low-risk exploration to move quickly while ensuring that consequential changes receive appropriate scrutiny.

Fewer handoffs should mean less unnecessary translation not less challenge, review or accountability.

What this looks like in practice

The Product Builder approach changes how a team reaches a decision, not just how quickly it produces an interface.

The following is an illustrative scenario, not a reported customer case study.

A business software provider wants to improve customer onboarding. Too many new customers need assistance before they can complete their first useful task.

A feature-led response might begin with a proposal to redesign the onboarding screens. Requirements would describe the revised journey, followed by design, development and launch.

A Product Builder team starts with the outcome: help more customers complete that first useful task without avoidable support.

The product manager, engineer and designer examine customer conversations, observed behaviour and support requests together. Suppose this reveals that the main difficulty is not the interface. Customers do not know which information they need before they begin.

The team explores two options: a preparatory checklist and an in-product assistant. AI helps create testable versions of both, but actual customers provide the evidence about whether either approach helps.

Technical exploration considers integration, permissions, data quality and ongoing support. Commercial exploration considers whether the added complexity is justified.

The team may discover that a straightforward checklist solves most of the problem. A more sophisticated AI feature might not be necessary.

The chosen approach is then built and released with appropriate testing, monitoring and ownership. The team measures successful onboarding, time to first value and support demand. It also checks whether any improvement has merely shifted work elsewhere.

The distinctive feature is not that one person performs every task. It is that customer, commercial, design and technical considerations remain connected throughout.



The builder's achievement is the improved customer outcome – not the sophistication of the technology used to produce it.

The boundary between a prototype and a product

Lower barriers to creation make it more important to distinguish what has been demonstrated from what has been proven.

A prototype can help establish whether people understand an interaction. A technical experiment can explore whether an integration is feasible. Neither, by itself, establishes that a service is ready for sustained use.

DORA's 2026 analysis identifies a gap between rapid prototyping and production integration, where edge cases, internal systems and verification can absorb the apparent gains from initial creation. [3]

The Product Builder model should make this boundary visible.

A useful distinction is between exploratory work, created to answer a question, and operational work, accepted for use under defined conditions.

Before exploratory work becomes operational, the team should know who will maintain it, how failures will be detected, what data it can access, how changes will be reviewed and how the service can be recovered.

Where AI is part of the customer-facing product, evaluation should also address the consequences of incorrect or inconsistent outputs. Where AI is only used to help build the product, the resulting software still needs to meet the organisation's normal expectations.

This is why the Product Builder should not be presented as a universal substitute for specialists.

Architecture, security, accessibility, reliability and domain knowledge are not obstacles to the model. They are capabilities that make it sustainable.

Organisations should make that expertise accessible through supported platforms, reusable components and clear review paths. The alternative is to make every small team rediscover the same difficult lessons.



Build broadly. Verify appropriately. Operate deliberately.

What changes for product leadership

As teams become more capable of exploring and delivering options, leadership must become more disciplined about where effort is directed and what earns further investment.

The CPO report describes a shift towards strategy, innovation and portfolio decisions. In its survey comparison, the share selecting portfolio management among their three main areas of focus rises from 13% to 24%. The report characterises this direction as the emergence of the 'Chief Product Investor'.

This is a change in reported priorities, not a measurement of the proportion of working hours spent on each activity. [1]

The useful idea is that product leadership should manage a portfolio of opportunities, not simply a larger queue of features.

In the proposed model, leaders establish which customer and business problems matter, provide proportionate resources for exploration, and increase commitment when evidence supports it.

They must also create permission to stop.

When a team can build something quickly, continuing can feel easier than challenging the original premise. A healthy investment approach rewards useful negative findings: discovering that an opportunity is weak before it consumes substantial resources is valuable work.

Leadership should also resist assuming that greater individual capability makes every organisation suitable for the same team size. A tightly scoped new product and a complex established platform have different dependencies and responsibilities.

The operating model should be designed around the work and its risks not around an arbitrary target for how many people AI ought to replace.

Finally, product and engineering leadership need to agree the model together. A product-led transformation imposed on engineering, or an engineering-led AI rollout disconnected from customer outcomes, would reproduce the separation the Product Builder is intended to address.



Manage a portfolio of opportunities, not simply a larger queue of features.

Building the capability, not just buying the tools

Product Builder capability requires deliberate development of judgement across disciplines. Tool familiarity is necessary, but insufficient.

A product professional needs enough technical understanding to recognise important constraints and challenge an implementation intelligently. An engineer needs enough customer and commercial understanding to question whether the proposed solution is worthwhile.

Neither needs to become an expert in every discipline.

The development goal is a combination of depth, working fluency and awareness of limits.

Teams should practise framing problems, designing experiments, evaluating evidence and making tradeoffs alongside AI-assisted creation. They should also practise recognising when specialist help is required.

DORA's 2026 discussion raises an important concern about learning: assistance can help people perform unfamiliar work while also allowing them to bypass some of the struggle through which expertise develops. This is a risk to manage, not a reason to reject assistance.[3]

One proposed response is structured simulation.

A team can work through a realistic product challenge using dummy data and simulated events. A stakeholder changes direction. Customer evidence contradicts the initial idea. A technical dependency appears. The commercial case weakens.

The purpose is to practise decisions and receive feedback before those decisions carry live consequences.

Simulation should then be followed by coached application to a real customer problem. Its role is to develop capability not to manufacture evidence that real customers will adopt a product.

Assessment should examine what people can explain and apply: the reasoning behind a decision, the quality of an experiment, the interpretation of evidence and the handling of uncertainty. A polished AI generated output should not, on its own, be treated as proof of competence.

The transition also needs an honest people narrative. Organisations should explain what is changing, what support is available and how responsibilities may evolve. Participation is unlikely to be meaningful where development is presented as support but experienced as an undisclosed selection exercise.



Practise decisions before they carry live consequences.

A practical route to adoption

The most useful starting point is a small, intact team working on a real problem within clear boundaries not an organisation-wide change of job titles.



DAYS 1 - 30
ESTABLISH

DAYS 31 - 60
APPLY

DAYS 61 - 90
EVALUATE

A 90-day pilot can test the working model. It should not assume that every product can safely reach production within that period.

Establish the conditions

In the first month, select a bounded customer problem with a credible opportunity to learn. Keep an existing team together where practical, with access to the relevant product, engineering, design and commercial expertise.

Agree the intended outcome, current baseline, decision rights and permitted uses of AI. Map where time is currently spent, including waiting, review and rework.

Provide a safe environment for experimentation and identify the specialist support the team will need.

The initial question is whether the team has the conditions to work differently not how many tools it has adopted.

A practical route to adoption

Apply the model

During the second month, work through the shared problem-to-value model.

Make assumptions explicit. Use AI where it helps create or assess options. Test with real customers where appropriate. Review the evidence jointly rather than asking each function to present its own separate account of progress.

Observe the operating model as carefully as the product idea. Where does work wait? Which decisions are unclear? What context is missing? Which reviews prevent defects, and which merely repeat earlier conversations?

Evaluate before expanding

In the third month, assess whether the approach has improved learning, delivery or customer value without unacceptable costs elsewhere.

The scorecard should cover learning quality and speed, delivery performance and reliability, and customer and commercial outcomes. Include total effort, review time, rework and ongoing operating costs not just the speed of initial creation.

Where outcomes have not yet had time to emerge, distinguish leading evidence from realised benefits.

The decision may be to expand, adapt or stop the pilot. A useful result is a clearer understanding of the conditions under which the approach works.

Scaling should follow demonstrated capability and repeatable practices, not enthusiasm for an impressive demonstration.



Build the capability to create value

The rise of the Product Builder is best understood as a shift in capability and organisational design not simply a forecast about the disappearance of a profession.

AI provides an opportunity to bring customer understanding, solution exploration and technical execution closer together. What organisations do with that opportunity will determine whether they become more effective or simply more productive at generating work.

The weak version of the Product Builder idea asks people to do more jobs with more tools.

The stronger version gives teams the context, capability and authority to solve problems together. It combines broad participation with specialist depth, faster experimentation with stronger evidence, and greater autonomy with clear accountability.

Product management does not become irrelevant in that model. Its judgement becomes more closely connected to creation.

Engineering does not become a commodity. Its expertise becomes more closely connected to customer and commercial decisions.

Leadership does not disappear into automation. It becomes responsible for designing a system in which good decisions can be made repeatedly.

The question is not whether every organisation should employ someone called a Product Builder.

It is whether the organisation can connect insight, judgement and execution well enough to turn its increased ability to build into an increased ability to create value.

The future is not simply about people who can build more. It is about people and teams who know what is worth building and can prove it.



Research and references

This paper combines external research with a proposed operating model. Reported experiences, experimental results and forecasts are treated as different forms of evidence.

[1] Products That Count, in partnership with Mighty Capital. (2026)
The 2026 CPO Insights Report: From Builder to Investor The New Product Mandate.
Supplied research report. Key passages: printed pp. 1 3, 7 11 and 15. Leadership survey findings are self-reported; forecasts about roles are not treated as established outcomes.

[2] Salva, R. J. / Google. (23 September 2025)
How are developers using AI? Inside our 2025 DORA report.
Research summary covering AI adoption, perceived productivity and the importance of the organisational environment.

[3] Baolin, J. and Harvey, N. / DORA. (10 March 2026)
Balancing AI tensions: Moving from AI adoption to effective SDLC use.
Qualitative analysis of 1,110 Google engineer responses. Used here for verification overhead, production integration and skill development considerations.

[4] Becker, J., Rush, N., Barnes, B. and Rein, D. / METR. (10 July 2025)
Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity.
Randomised study of 16 experienced developers and 246 tasks. Its result applies to the tools, work and participants studied not all software development.

[5] Becker, J. et al. / METR. (24 February 2026)
We are Changing our Developer Productivity Experiment Design.
Follow-up explaining why selection effects and measurement difficulties limit reliable estimates of the improvement from newer tools.

[6] DORA / Google Cloud. (2025)
DORA AI Capabilities Model.
Companion model describing seven technical and organisational capabilities that support effective AI-assisted development.

Tarigo Product Management Training
Howley Park Business Village,
Olympus House 2,
Pullan Way,
Morley,
Leeds LS27 0BZ

info@tarigo.co.uk
0113 450 8688

tarigo.co.uk