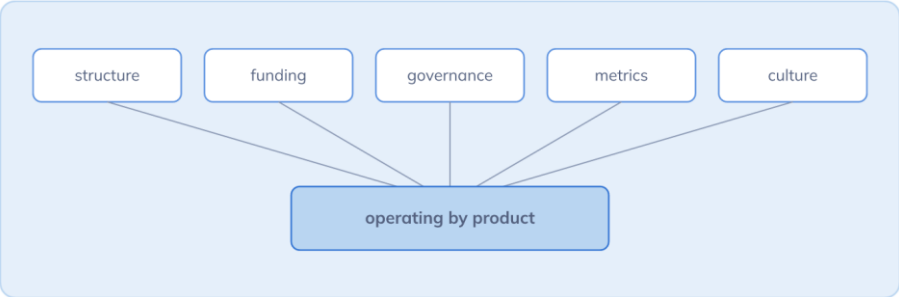
Scrum Manager
Uncovering better ways

Guide

Agile product operating model

In-depth development of the State of the art topics



Updated November 2026

Scrum Manager® · Skill Arena

About this document

This document is an in-depth development of the topics in the reference map (State of the art) for operating an organization by product (how product work is structured, funded, governed, and measured, and what culture sustains it) instead of managing it as a succession of projects, which is available at: Skill Arena.

Use it as reference material to keep your practice aligned with, and checked against, current professional knowledge.

It is complemented by the training and assessment platform in Skill Arena. In the Agile product operating model area you can take training tests to check and improve your level of knowledge and, if you wish, you can also obtain a diploma that provides curricular accreditation of professional competence and currency in this area.

State of knowledge

Knowledge about the product operating model evolves progressively. Its principles, such as the shift from project to product, empowered teams, and governance by outcomes, have been settled for more than a decade, but their synthesis and their nomenclature are being fixed now: the two reference formulations date from 2024, and Scrum.org reformulated its own in 2026 as a set of principles. Artificial intelligence makes building cheaper and does not change those principles; instead, it throws them into sharper relief. Throughout the document, the labels (ESTABLISHED, CONSOLIDATING, EMERGING) help identify the maturity of each concept:

ESTABLISHED settled consensus; knowledge taken as required.
CONSOLIDATING gaining adoption fast; not yet universal but already relevant.
EMERGING recent frontier; high relevance and high volatility.

Contents

Chapters of the guide, in the order of the State of the art.

Block A · Foundations and framing of the product operating model

1. What an operating model is and what “by product” means
2. From project to product: the paradigm shift
3. Outcomes over outputs
4. The two formulations and their divergence

Block B · The product as an operating unit and the portfolio

5. What a product is in this model
6. Taxonomy and topology of products
7. Product portfolio management
8. From strategic context to product strategy

Block C · Structure: empowered product teams, roles, and organizational design

9. Empowered product teams
10. Roles and competencies of the model
11. Product leadership
12. Organizational design: topologies, autonomy, and dependencies

Block D · Funding and governance by value streams

13. From project funding to product or value stream funding
14. Lean budgeting and Beyond Budgeting
15. Governance, product authority, and constraints

Block E · Outcome orientation: metrics and evidence-based governance

16. Objectives by outcomes: OKRs
17. Evidence-Based Management (EBM) and the four Key Value Areas
18. Delivery cadence as a condition of the model
19. Portfolio measurement and health of the model

Block F · Culture of product work

20. Product mindset, trust, and psychological safety
21. Learning, experimentation, and error management
22. The relationship with stakeholders and with “the business”

Block G · The transition to the product operating model

23. The transition: holistic nature and incremental journey
24. Leading the change, objections, and power dynamics
25. The boundary with scaling frameworks

BLOCK A · FOUNDATIONS AND FRAMING OF THE PRODUCT OPERATING MODEL

Chapter 1. What an operating model is and what “by product” means · ESTABLISHED

The product operating model is the set of decisions on structure, funding, governance, metrics, and culture with which an organization builds and operates products on an ongoing basis, and it is formulated as a set of principles.

Introduction

An operating model describes how a company organizes itself to do its work: how it groups people, how it allocates money, who decides what, how it measures success, and what culture sustains all of the above. For that model to be a “product” model means that the organization is structured around durable products and the stable teams that look after them, leaving behind organization by temporary projects or by isolated functions.

It is worth establishing from the outset what kind of object this is. The two reference formulations present it as a set of principles that each organization applies in its own way, and understanding it this way avoids the most common mistake, which is to look for a recipe that does not exist. In this chapter we will see which decisions the model orders, why it is formulated as principles, and where its territory ends in relation to neighboring subjects.

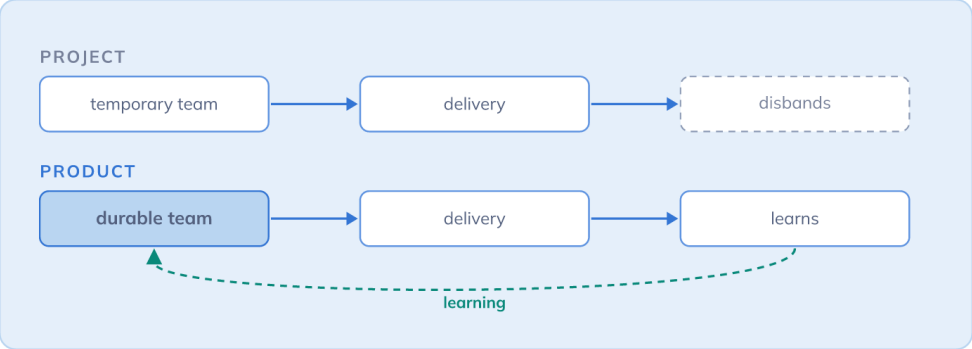


Figure 1. *From project to product: the project is set up, delivers, and disbands, while the product keeps a durable team that learns from every delivery.*

1.1 The five dimensions it orders

To walk through the operating model, this topic groups its decisions into five dimensions that must move together.

- **Structure.** How teams and people are grouped.
- **Funding.** How money is allocated.
- **Governance.** Who has the authority to decide what.
- **Metrics.** How value and progress are measured.
- **Culture.** What behaviors and values are expected.

This grouping is a synthesis of this topic, designed to order the journey through the following blocks. It should not be confused with the three dimensions with which Marty Cagan articulates his model or with the four key areas of the Scrum.org model, which we will see in chapter 4. The three ways of ordering the subject cover the same territory from different angles.

What is decisive is the coherence among the five. Changing only one, for example asking for empowered teams while keeping project funding, produces friction and often leads to the failure of the transition.

Central idea. A product operating model describes how the entire company is organized so that product work is possible. That is why it is, above all, the responsibility of leadership.

1.2 A set of principles

The two reference formulations agree on this point. Cagan insists that the model is a set of principles and warns against process addiction, which confuses following ceremonies with operating well. Scrum.org stresses that the model belongs to those who operate it and that it admits many valid instances.

Scrum.org made this explicit in 2026. In January it published the guiding principles of its Agile Product Operating Model (APOM), and in February Dave West presented the change as a move from a framework to a set of guiding principles. The document justifies it on the grounds that every situation is unique, so the model gives up offering a complete blueprint and proposes principles that each organization adapts to its starting point.

The practical consequence is that two companies can both operate by product with very different structures, cadences, and tools. What they share are the principles, and each organization decides the concrete form.

1.3 The boundary with the craft of product and with change management

This topic deals with the operating model, that is, how the company is organized. The craft of product (discovery, prioritization, writing OKRs, or user research) is a separate subject and appears here only in the place it occupies within the model.

Nor is it a course on organizational change management as a general discipline, and of the transition only what is specific to this model is covered. Scaling frameworks (SAFe, LeSS, Nexus) form another layer, that of its instantiation, as we will see in chapter 25.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain what an operating model is and what it means for it to be a product operating model.
- Name the five dimensions with which this topic orders the model and distinguish them from Cagan's three dimensions and from the four key areas of the APOM.
- Distinguish an operating model from a methodology or a prescribed process.
- Argue why the five dimensions must change coherently.
- Explain the APOM's move from framework to set of guiding principles in 2026 and what it implies for those who adopt it.
- Place the boundary of the model in relation to the craft of product, change management, and scaling frameworks.
- Recognize when an organization confuses following a process with operating by product.

Common mistakes and antipatterns

- **Looking for the recipe.** Treating the model as a framework with numbered steps to implement, instead of as principles that each organization adapts to its context.
- **Changing only one dimension.** Asking for empowered teams while keeping project funding, activity metrics, or stage-gate governance; the inconsistency cancels out the change.
- **Confusing craft and model.** Mixing product discovery or planning techniques with the design of the operating model, which operates at another level.
- **Mixing the ways of ordering the model.** Taking the five dimensions of this topic for Cagan's three or for the four areas of the APOM, and attributing to one formulation what belongs to another.

Further reading

· SVPG — The Product Operating Model

· Scrum.org — The Agile Product Operating Model

· Scrum.org — APOM Guiding Principles Have Published

BLOCK A · FOUNDATIONS AND FRAMING OF THE PRODUCT OPERATING MODEL

Chapter 2. From project to product: the paradigm shift ·
ESTABLISHED

The product model replaces the models of origin with a continuous and durable effort, and with it change who decides, how work is funded, and what teams are accountable for.

Introduction

Most organizations that adopt a product model come from one of three models of origin. All three share an assumption, that the work comes to an end, and the product model breaks it. A product is looked after and evolves for as long as it delivers value, and that change of frame carries everything else with it, from who decides to how the work is funded and what teams are accountable for.

In this chapter we will see the models of origin and their limits, what changes when moving to product, and how Mik Kersten gave this shift a name, first in *Project to Product* (2018) and then in *Output to Outcome* (2026).

2.1 The models of origin and their limits

Each model of origin has a reasonable internal logic, which is why it persists.

- **The model of technology in service of the business.** The technology department acts as an internal supplier that handles the requests of the business areas.
- **The project model.** Work is organized into initiatives with a scope, a budget, and a closing date, and teams are assembled for each one and disbanded when it ends.
- **The feature team model.** Teams are stable, but they receive lists of features to build and are accountable for delivering them.

All three clash with digital products that live for years and compete in fast-changing markets. The project model, for example, optimizes the delivery of a fixed scope on time and on budget, but penalizes learning, because any deviation from the plan is experienced as a failure even when it is exactly what the market was asking for.

The feature team model deserves a separate mention because it resembles the product model. Its teams are durable and often work with agile practices, but the solution reaches them already decided and their success is measured by what they deliver. As we will see in chapter 3, what is assigned to the team is what separates the two models.

2.2 What changes when moving to product

When moving to product, three things shift at once.

- **The unit of organization**, which moves from the temporary project to the persistent product.
- **Accountability**, which moves from complying with a plan to achieving outcomes.

- **The relationship with time**, which moves from a calendar with an end to a continuous cadence of delivery and learning.

Renaming projects as products without moving these three things is the most frequent failure of the transition, which we will return to in chapter 23.

Central idea. Moving to product is not about renaming the teams, but about changing how the work is funded, what people are accountable for, and the pace at which they learn. If all of that stays as it was in the project, the organization keeps operating by project even if it says otherwise.

2.3 From *Project to Product* to *Output to Outcome*

Mik Kersten gave this shift a name in *Project to Product* (2018), where he proposes organizing technology work around products and the value streams that serve them, and measuring it by the value that flows through them. The book popularized the expression that gives this chapter its title and has become a reference for the transition.

In *Output to Outcome* (2026) Kersten takes up the thesis again with artificial intelligence at its center. He argues that, when AI speeds up building, the bottleneck shifts to the operating model itself, that is, to how the organization decides what to build, how it funds it, and how it measures what it achieves.

AI thus makes the move from project to product more urgent. An organization that accelerates its way of working by projects produces faster, but still does not know whether what it produces moves any outcome.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Describe the three models of origin (technology in service of the business, project, and feature teams) and their internal logic.
- Explain the assumption they share and why it clashes with durable digital products.
- Distinguish a feature team from a product team by what is assigned to it and by what it is accountable for.
- Identify the three things that change when moving to product (unit, accountability, and relationship with time).
- Detect when an organization has renamed its projects as products without changing the substance.
- Explain Kersten's contribution in *Project to Product* and his thesis in *Output to Outcome* about the bottleneck in the age of AI.

Common mistakes and antipatterns

- **A product with a closing date.** Keeping the project mindset under the new name and planning the product as a single delivery with an end.

- **The business asks, technology serves.** Keeping the client-supplier relationship of the model of origin, which prevents being jointly accountable for outcomes.
- **Success as a plan fulfilled.** Continuing to measure success by adherence to the plan instead of by the outcome achieved, which penalizes learning.
- **Speeding up the project model with AI.** Using the new tools to produce faster within the same model, which widens the bottleneck instead of resolving it.

Further reading

· SVPG — The Product Operating Model

· Scrum.org — Project to Product and the Agile Product Operating Model

· Mik Kersten — Project to Product

· Mik Kersten — Output to Outcome

BLOCK A · FOUNDATIONS AND FRAMING OF THE PRODUCT OPERATING MODEL

Chapter 3. Outcomes over outputs · ESTABLISHED

The cross-cutting guiding principle of the model consists of assigning teams problems to solve and outcomes to achieve, and it weighs more as producing outputs gets cheaper.

Introduction

An output is what is built (features, releases, code), and an outcome is the change that it brings about in the customer and in the business. The principle of outcomes over outputs establishes that teams are accountable for the latter. It is the criterion that best distinguishes the product model from a feature factory, where success is measured by the quantity delivered.

In this chapter we will see what it means to assign problems, why artificial intelligence reinforces the principle, and what, according to Marty Cagan, is the underlying tension that makes it so hard to sustain, the one that pits it against predictability.

Figure 2. *Governing toward the outcome: what the team delivers only counts when it changes someone's behavior and that moves an outcome.*

3.1 Assigning problems to solve

The operational consequence is that an empowered team is assigned a problem to solve or an outcome to achieve, and is trusted to discover the best solution. In the feature team

model, by contrast, the solution arrives already decided and all that is left for the team is to build it.

Assigning outcomes does not mean abandoning the team either. It means giving it context, constraints, and a measure of success, and leaving the decision about the how in its hands.

The litmus test. If a team can declare success after delivering everything planned without having moved any value metric, the organization is still measuring outputs.

3.2 Why AI reinforces the principle

As producing outputs becomes cheaper and faster, with AI tools as the most visible example, the main risk becomes quickly building what does not deliver value. The cheaper it is to build, the more it matters to govern toward the outcome, because the bottleneck shifts from the capacity to build to the capacity to decide what deserves to be built.

Marty Cagan analyzed this in July 2026, in an article on the so-called AI productivity paradox (teams that deliver faster without their outcomes improving). He argues that there is no such paradox, because most use AI to speed up their old way of working by projects, a model designed to produce outputs.

Strong teams, in his description, use AI to check earlier whether a solution deserves to be built, and they separate what is built to learn from what is built to earn revenue. This is the territory of discovery, which we will return to in chapter 21.

3.3 Outcomes and predictability

In October 2026 Cagan pinpointed where the underlying tension lies. Outcomes need outputs, so the choice that is really at stake is between outcomes and predictability. The focus on output would be the symptom, and its cause, a deeply rooted desire for predictability.

Committing to dates and deliverables gives a sense of control that outcomes cannot offer, because many product ideas, if not most, are not going to work. Pursuing outcomes requires humility and good discovery to find out which ones are worth it.

Predictability has legitimate uses. The problem arises when an organization puts it first, because then the outcome is left as a side effect that is hoped for from the promised features. According to Cagan, most companies implicitly choose this way, and it is up to leadership to decide what it rewards. As we will see in chapter 22, that preference is at the root of many objections to the model.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Distinguish output from outcome with examples from one's own context.

- Reword a request expressed as a list of features in terms of a problem and an outcome.
- Explain why measuring outputs leads to the feature factory.
- Argue why the cheaper building brought by AI increases the importance of the outcome.
- Explain the tension between outcomes and predictability and recognize when an organization in fact prioritizes predictability.
- Define an outcome-oriented measure of success for a given initiative.

Common mistakes and antipatterns

- **Governing with a feature list.** Directing the work with deliverables committed by date instead of with outcomes to achieve.
- **Taking speed for value.** Treating the quantity or speed of delivery as proof of success, with or without AI, without checking whether any value metric moved.
- **Decorative outcome.** Stating an outcome in the presentation and continuing to evaluate the team by the delivery of the planned scope.
- **Predictability as a hidden priority.** Declaring that outcomes are pursued while rewarding meeting dates in practice, so that the outcome is left as a hoped-for side effect of what was promised.

Further reading

· Marty Cagan (SVPG) — The AI Productivity Paradox

· Marty Cagan (SVPG) — Predictability vs Outcomes

· SVPG — The Product Operating Model: An Introduction

· Scrum Manager BoK — Outcome over output (in Spanish)

· Scrum Manager BoK — Feature factory (in Spanish)

BLOCK A · FOUNDATIONS AND FRAMING OF THE PRODUCT OPERATING MODEL

Chapter 4. The two formulations and their divergence ·

CONSOLIDATING

Cagan and SVPG's Product Operating Model and Scrum.org's Agile Product Operating Model share a common body and diverge in their anchoring, in the way they are ordered, and even in their name.

Introduction

The product operating model is not unified under a single label. Two reference formulations coexist that share a broad common body and diverge in emphasis, in anchoring, and in name. Knowing both avoids two symmetrical mistakes, that of believing there is a single canonical model and that of believing they are talking about different things.

In this chapter we will see how each formulation orders the model (Marty Cagan's in three dimensions and twenty principles, and Scrum.org's, since 2026, in four key areas and four core principles), what they share and where they diverge, and what Mik Kersten's reading contributes.

4.1 The Product Operating Model (Cagan and SVPG)

Marty Cagan and SVPG formulate it above all in *Transformed* (2024) as a model of principles, articulated in three dimensions.

- **How you build**, with continuous and reliable delivery.
- **How you solve problems**, with discovery and empowered teams.
- **How you decide which problems to solve**, with product strategy.

Its twenty principles are grouped into five concepts (culture, strategy, teams, discovery, and delivery), with four principles in each. Cagan deliberately avoids the adjective “agile” and maintains a critical stance toward process addiction.

4.2 The Agile Product Operating Model (Scrum.org)

Scrum.org formulates the APOM anchored in Scrum, in Nexus, and in Evidence-Based Management (EBM). In 2025 it described it with seven characteristics, which remain as its initial formulation (unique to each organization, holistic, evidence-based, with empowered teams, empirical, complete, and with change management built in).

Since January 2026 it has formulated it with guiding principles, in a document that defines the APOM as a principle-based approach that uses evidence to align the organization's strategy with technical execution. The model is ordered into four key areas.

- **Strategy**, which integrates business and technology objectives and measures them by outcomes.

- **People**, focused on empowered, cross-functional teams, with servant leadership and communities of practice.
- **Structure**, which replaces bureaucratic hurdles with governance, agile procurement, and team-owned processes.
- **Value Cycle**, which joins discovery, delivery, and support in a single continuous feedback loop.

On those areas rest four core principles.

- **Evidence-Based.** Decisions are driven by objective data and learning.
- **Empirical Agility.** Small, frequent releases are prioritized to accelerate feedback and reduce the cost of failure.
- **End-to-End Ownership.** The product team owns the whole lifecycle, from ideation to production support, to reduce handoffs and delays.
- **Outcome Focused.** Success is measured by the actual impact on customer value and business goals.

Each area is developed in a series of guiding principles, and each principle has an associated evidence metric that makes it possible to check whether the organization complies with it and to adapt the model according to real results.

4.3 Common body and divergences

The two formulations share the essentials: product over project, outcomes over outputs, empowered teams, evidence-based governance, and continuous delivery.

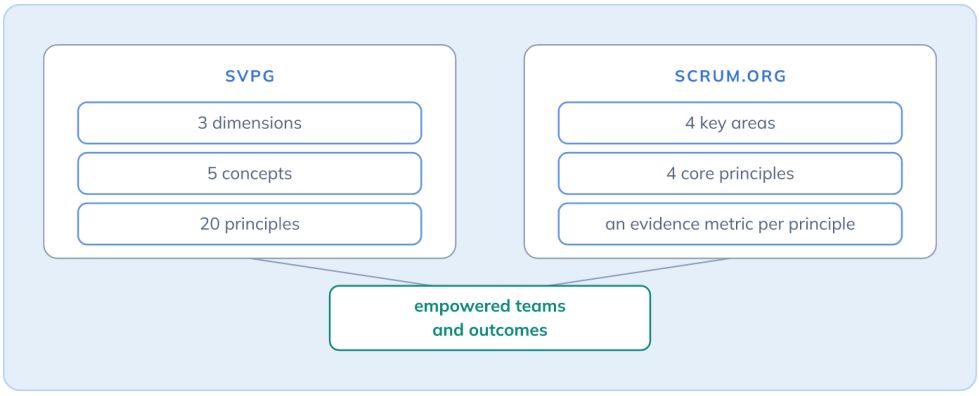


Figure 3. *Two formulations with a common body: sVPG orders the model in three dimensions and five concepts, and Scrum.org in four key areas and four core principles, on the same core of empowered teams and outcomes.*

They diverge in three respects. In anchoring, because the APOM rests on Scrum, Nexus, and EBM, while Cagan's model does not rest on any framework. In the weight of process, which Cagan distrusts. And in the label, because the adjective “agile” belongs to one of the formulations and not to the consensus. This topic covers the common body and points out the divergences where they appear.

Mik Kersten contributes a third reading, centered on value streams. In *Output to Outcome* (2026) he presents it as a continuation of *Project to Product* (2018), and it is

best read as an influential articulation of the same shift, without elevating it to a third formulation of the model.

Why it is consolidating. The explicit pairing of the two formulations is recent and the APOM is still evolving. In little more than a year it has gone from being described with seven characteristics to being formulated with key areas, core principles, and evidence metrics.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Describe the three dimensions of Cagan's Product Operating Model and the five concepts into which he groups its twenty principles.
- Describe Scrum.org's APOM with its four key areas and its four core principles, and explain the role of evidence metrics.
- Place the seven characteristics of 2025 as the initial formulation of the APOM.
- Identify the common body that the two formulations share.
- Explain the points of divergence, including that of the adjective “agile.”
- Place Kersten's reading, centered on value streams, in relation to the two formulations.
- Attribute a specific practice or statement to the formulation it comes from.

Common mistakes and antipatterns

- **A single canonical model.** Presenting one of the two formulations as the product operating model and ignoring the other.
- **Describing the APOM with its initial formulation.** Presenting it only with the seven characteristics of 2025, without the key areas, core principles, and evidence metrics of 2026.
- **Confusing the label with the consensus.** Treating the “agile” of the APOM as part of the common body when it is specific to that formulation.
- **Mixing without attributing.** Combining elements of the two formulations without indicating where each comes from, which loses the traceability of the divergences.
- **Counting three formulations.** Elevating Kersten's reading to a formulation of the model in its own right, when he himself presents it as a continuation of his 2018 work.

Further reading

· SVPG — The Product Operating Model: An Introduction

· Scrum.org — APOM Guiding Principles Have Published

· Scrum.org — APOM State of Play, Part 1: Fundamentals

· Mik Kersten — Output to Outcome

BLOCK B · THE PRODUCT AS AN OPERATING UNIT AND THE PORTFOLIO

Chapter 5. What a product is in this model · ESTABLISHED

A product is recognized by clear boundaries, identified users, and measurable value, and being a product is a matter of degree.

Introduction

If the model is organized by product, the first question is what counts as a product. The answer has consequences, because the boundaries of the teams, the allocation of funding, and the distribution of authority all depend on it. A product, in this model, has clear boundaries, recognizable users and stakeholders, and a value that can be measured.

In this chapter we will see how Scrum.org and the two formulations define it, why it is advisable to treat being a product as a matter of degree, and what distinguishes it from neighboring concepts.

5.1 The characteristics of a product

A product serves identifiable users and stakeholders, solves a problem or meets a need, has a reasonably sharp boundary with respect to other products, and generates value that can be observed and measured. These characteristics make it possible to assign it a durable team and hold that team accountable for outcomes, because without them the team would not know what it is accountable for.

The APOM guiding principles (2026) condense this into a brief definition. A product is a vehicle for delivering value, with clear boundaries, known stakeholders, and well-defined users or customers.

Figure 4. *A product as an operating unit: clear boundaries, recognized users and stakeholders, and a value that can be measured.*

The second installment of Scrum.org's series on the state of the model (Dave West, June 2025) extends the list. In addition to boundaries, users, and stakeholders, a product has these traits.

- An investor who pays for building and supporting it.
- A value that exceeds its cost.
- A forecast of how it will evolve, because it exists beyond the work done on it.

- A connection with the business strategy.

5.2 Being a product, a matter of degree

The same article proposes seeing product-ness as a scale. Some sets of capabilities have more product traits than others, and the lack of one does not disqualify something as a product, because adopting a product focus often helps to build the missing ones. When something does not fully fit the notion of product, the model is applied with nuances.

The boundary of the product is a practical decision that each organization makes according to what makes sense for it, and it is worth reviewing regularly. To assess it, West suggests asking whether the structure improves the teams' ability to deliver value frequently and whether it makes sense from the point of view of funding and control.

5.3 Product versus project, feature, service, or platform

It is worth distinguishing the product from neighboring concepts.

- **Project.** It is temporary, whereas the product is continuous.
- **Feature.** It is a part of the product and has no standing of its own as a product.
- **Service.** It can be a product if it has its own users and value.
- **Platform.** It is a particular type of internal product whose customers are other teams, as we will see in chapter 6.

The boundary matters. Drawing the boundaries of a product badly creates dependencies and overlaps between teams. Defining the product is, in practice, an organizational design decision, which we will cover in chapter 12.

What you should be able to do

Having mastered this chapter, a professional is able to:

- List the characteristics that define a product in this model.
- State the definition of product in the APOM guiding principles (a vehicle for delivering value) and its three traits.
- Assess whether a specific effort can be treated as a product, understanding being a product as a matter of degree.
- Distinguish a product from a project, a feature, a service, or a platform.
- Explain why the definition of product shapes team boundaries, funding, and the distribution of authority.

Common mistakes and antipatterns

- **A feature promoted to product.** Treating as an independent product what is a part of another, which multiplies teams and dependencies.

- **A product without measurable value.** Declaring something a product whose value cannot be observed, which makes it impossible to hold the team accountable for outcomes.
- **Blurred boundaries.** Leaving ambiguous boundaries between products, a source of overlaps and conflicts of authority.
- **Demanding the perfect product.** Refusing to treat something as a product because it lacks some trait, when being a product is a matter of degree and the product focus itself helps to complete it.

Further reading

· Scrum.org — APOM State of Play, Part 2: Product Definition

· Scrum.org — The Agile Product Operating Model

· Scrum.org — APOM Guiding Principles Have Published

· SVPG — The Product Operating Model

BLOCK B · THE PRODUCT AS AN OPERATING UNIT AND THE PORTFOLIO

Chapter 6. Taxonomy and topology of products · ESTABLISHED

Products can be customer-facing, internal, platforms, or enablers, differentiating or commodity, and each one can have its own operating model.

Introduction

Not all products are alike, and forcing a single mold on all of them is a frequent mistake. A simple taxonomy helps to reason about that diversity, because each type of product has different users, risks, and rhythms and can, therefore, operate differently.

In this chapter we will see the types of product, the axis that separates differentiating products from commodity ones, what Marty Cagan has clarified about internal products, and the principle that each product can have its own operating model.

6.1 Types of product

A first classification looks at whom the product serves and what need it meets.

- **Customer-facing.** They serve external users and compete in a market.
- **Internal.** They serve users within the organization itself, such as operations tools.
- **Platforms.** They are products whose customers are other teams, whose cognitive load they reduce by offering them capabilities as a service.
- **Enablers.** They address cross-cutting needs, such as security, data, or identity.

The distinction guides how the value of each is measured. That of a customer-facing product is observed in the market, and that of a platform, in what it makes easier for the teams that use it.

6.2 Differentiating products and commodity products

The APOM guiding principles (2026) add a second axis when dealing with strategy. Some products differentiate the organization from its competitors, and others offer commodity capabilities that any company needs. Being transparent about the role each product plays clarifies the decisions made about it.

The axis helps decide how to serve each product. A differentiating one justifies a dedicated team that discovers and learns, while a commodity capability can be handled with a shared platform. Scrum.org's series on the state of the model even raises whether some capabilities are better outsourced or bought.

6.3 Internal products

In April 2026 Cagan clarified how the model applies to internal products, and he distinguishes three kinds.

- **Customer-enabling tools**, used by the staff who work with customers.
- **Productivity tools**, which improve the organization's internal work.
- **Internal platform services**, which support much of the business and on which other teams build.

Internal products face the same four risks as commercial ones (value, usability, feasibility, and viability), although the bar for each one is usually lower. The internal user cannot leave for a competitor, some training or documentation is often enough, the demands of scale and performance are lower, and the product operates in a bounded part of the business.

Platform as a product. Treating the platform as a product, with its internal users, its value proposition, and its measurement, prevents it from turning into a standardization project disconnected from those who use it.

6.4 Each product, its own operating model

Both formulations recognize that each product can have its own operating model within a common framework. A mature, regulated product and one in an exploration phase do not need the same cadence, the same governance, or the same metrics.

The organization sets the common constraints (compliance, security, funding), and within them the model of each product is decided by those who operate it.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Classify products as customer-facing, internal, platforms, and enablers.
- Place a product on the differentiating or commodity axis and reason whether a dedicated team or a shared platform is more appropriate.

- Explain the three kinds of internal products that Cagan distinguishes and why the bar for their four risks is usually lower.
- Explain why each type of product can measure its value differently.
- Argue the principle that each product can have its own operating model within common constraints.

Common mistakes and antipatterns

- **A single mold.** Imposing the same cadence, the same governance, and the same metrics on all products regardless of their nature.
- **Platform as a project.** Managing the platform as a standardization initiative without treating the teams that use it as customers.
- **The invisible enabler.** Failing to recognize cross-cutting enablers as products, which leaves them with no owner and no measurement.
- **An internal product without risks.** Assuming that an internal product does not need discovery because its users cannot choose, when it faces the same risks as a commercial one, albeit with a lower bar.

Further reading

· Scrum.org — The Agile Product Operating Model

· Marty Cagan (SVPG) — Commercial vs Internal Products

· Scrum.org — Shifting to a Product-Based Operating Model with APOM: The Accenture Way (podcast)

· Scrum.org — APOM Guiding Principles Have Published

BLOCK B · THE PRODUCT AS AN OPERATING UNIT AND THE PORTFOLIO

Chapter 7. Product portfolio management · CONSOLIDATING

Portfolio management aligns, funds, and prioritizes the set of products, including the bets and the work that cuts across several of them, and leaves the planning of each product to the craft.

Introduction

Portfolio management operates one level above the individual product. It decides which products to invest in, how resources are distributed among them, and how efforts that cut across several products are coordinated. The roadmap of a specific product is a matter for the craft, and here we deal with the governance of the whole, that is, where the organization places its bets.

In this chapter we will see how a portfolio is governed by outcomes, the six principles that Scrum.org has proposed for it, and how work between products is coordinated without returning to project management.

7.1 From portfolio planning to bets

In a product model, the portfolio is governed with outcomes. Its planning starts by setting objectives, often through portfolio OKRs, and allocating persistent resources to the products best placed to contribute to them. Large initiatives are treated as bets, that is, as risky investments that are reviewed according to the evidence.

OKRs are the usual link between the portfolio and the teams. We will see their role within the model in chapter 16, and their technique is developed in the Skill Arena topic "OKRs and product strategy."

7.2 Six principles for the portfolio

Dave West, of Scrum.org, proposed six principles for managing the product portfolio in June 2025.

1. **Fund the products (teams), not the work.** Teams are funded for their value and strategic fit, and they are the ones who decide which work contributes most.
2. **Minimize cross-product initiatives that require portfolio management.** Leadership limits them to a small set so that teams keep room to decide and innovate.
3. **Discovery, delivery, and validation happen within the product teams.** The team that is going to build something is also the one that researches and tests it.
4. **Work is defined in terms of outcomes and measures.** Each initiative states the outcome it pursues and how it will be measured, and OKRs are one of the ways of doing so.
5. **Incentives, structure, and governance align with business and product strategy.** If they do not fit the business and product strategy, they can undermine an otherwise well-designed model.

6. **Transparency, cadence, and clear ownership are required.** Visibility, a regular review rhythm, and clear ownership within the product, across products, and in the portfolio make it possible to adapt.

The first is the one that clashes most with common practice, because it moves funding from one-off initiatives to durable teams, as we will see in chapter 13.

7.3 Work between products

Much of the value arises at the intersection of several products. Portfolio management coordinates that work by aligning common objectives and reducing dependencies, without creating coordination committees that take authority away from the teams. West's second principle points in the same direction, because the fewer cross-product initiatives that have to be governed from the portfolio, the more room each team keeps.

It is one of the practices that is being formalized the most and in which organizations vary the most, hence its consolidating status.

A portfolio by outcomes. If the portfolio is still governed as a list of projects that are funded and controlled by stage gates, the organization operates by project at scale, even if its teams are called product teams.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Distinguish portfolio management from the roadmap of a specific product.
- Explain how a portfolio is governed by outcomes and what a bet that is reviewable according to the evidence is.
- List Scrum.org's six portfolio principles and explain the one about funding the products (teams), not the work.
- Identify how to coordinate work between products without reintroducing project management.
- Recognize the role of OKRs as a link between the portfolio and the teams.

Common mistakes and antipatterns

- **The portfolio as a list of projects.** Continuing to govern the whole as a list of initiatives with scope and date, stage-gate control included.
- **Funding the work.** Allocating money to one-off initiatives instead of to products and their teams, which forces every piece of work to be negotiated and takes away from the teams the decision about what contributes most.
- **The coordination committee.** Coordinating work between products with layers of governance that take authority away from the teams.

- **Multiplying cross-product initiatives.** Launching many cross-cutting initiatives governed from the portfolio, which consume the teams' capacity and leave them little room for their own decisions.
- **The irreversible bet.** Treating large initiatives as closed commitments instead of as investments that are reviewable according to the evidence.

Further reading

· Scrum.org — APOM State of Play, Part 3: 6 Guiding Principles for Product Portfolio Management

· Scrum.org — The Agile Product Operating Model (APOM): an Evidence-Based Approach

· Scrum.org — The Agile Product Operating Model

· Skill Arena — OKRs and product strategy

BLOCK B · THE PRODUCT AS AN OPERATING UNIT AND THE PORTFOLIO

Chapter 8. From strategic context to product strategy ·

ESTABLISHED

Business strategy cascades into product strategy and reaches the teams as a context of vision, focus, and transparency that guides without dictating the solution.

Introduction

Empowered teams need a direction that allows them to decide for themselves. That direction arrives as strategic context, made up of a vision, a focus, insights, and transparency that allow each team to decide well without anyone dictating the solution.

In this chapter we will see the place of strategy in the model, how the two formulations position it, and what role insights and focus play. The techniques for formulating strategy belong to the craft and are developed in the Skill Arena topic “OKRs and product strategy.”

8.1 Strategy as context

Business strategy is translated into product strategy, and this reaches the teams in the form of problems to solve and outcomes to achieve, framed by a clear vision and focus. Focus matters as much as vision, because a strategy that wants everything guides nothing.

Transparency is what allows context to replace control. When teams see the logic behind the decisions, and not just the decisions, they can adapt their work to what they discover without losing their way.

Central idea. The clearer the strategic context, the less need there is for direct control. Where context is missing, the organization tends to reintroduce micromanagement to make up for it.

8.2 Strategy in the two formulations

For Cagan, deciding which problems to solve is one of the three dimensions of the model and rests on product strategy, as we saw in chapter 4. The APOM guiding principles (2026) devote one of their four key areas to strategy and define product strategy as the intersection of business strategy and technology strategy, together with the constraints that govern them.

The APOM asks that each product have a clear description of its value, its business, its technology, and its operations, and that the role it plays in the organization's strategy be transparent. The principles of this area include the following, each with its evidence metric.

- **Start with why.** Product strategy must have a purpose understood by everyone.
- **Customer first.** Strategic options are filtered by the value they deliver to customers and users.
- **Minimal dependencies.** Dependencies are inevitable, but they are made transparent and kept to a minimum.
- **Measurement drives progress.** Each strategic initiative has its outcome metrics built in, and the evidence is the percentage of initiatives that have them.

8.3 Insights and focus

Insights about the market, data, technology, and customers are the material from which strategy is built, and they are shared with the teams so that they can decide with sound judgment. Cascading the strategy consists of sharing understanding. When a team understands the why, it can adapt the how to what it discovers, which is the behavior the model seeks.

Focus completes the context. A focused strategy allows teams to prioritize and to give up what does not contribute, and OKRs, which we will see in chapter 16, are a common way of expressing it.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain how business strategy is translated into product strategy and reaches the teams.
- Distinguish leading by context from leading by instructions.
- Explain the definition of product strategy in the APOM guiding principles as the intersection of business strategy and technology strategy.

- Place strategy in the two formulations, as a dimension in Cagan and as a key area in the APOM.
- Argue why focus is an essential part of strategy.
- Describe the role of insights and transparency in empowerment.

Common mistakes and antipatterns

- **A strategy that wants everything.** Stating a strategy without focus, which does not allow teams to prioritize or to say no.
- **Cascading tasks, not context.** Handing out a list of work instead of the understanding that allows the team to decide the how.
- **Decisions without a why.** Communicating decisions without their logic, which forces teams to execute without being able to adapt.
- **Product strategy without technology.** Formulating it only from the business side, without the technology perspective or the constraints that govern both.

Further reading

· SVPG — The Product Operating Model: An Introduction

· SVPG — The Product Operating Model

· Scrum.org — APOM Guiding Principles Have Published

· Scrum Manager BoK — Product strategy (in Spanish)

· Skill Arena — OKRs and product strategy

BLOCK C · STRUCTURE: EMPOWERED PRODUCT TEAMS, ROLES, AND ORGANIZATIONAL DESIGN

Chapter 9. Empowered product teams · ESTABLISHED

The cross-functional, durable team that is assigned problems and is accountable for outcomes is the centerpiece of the model, and Cagan places it at its center.

Introduction

The empowered product team is the centerpiece of the product operating model, and Marty Cagan places it at the center of his model. It is cross-functional, because it brings together the skills needed to discover and deliver; durable, because it persists beyond any particular initiative; and accountable for an outcome that can be observed in users or in the business.

In this chapter we will see what each of those traits means, what makes empowerment real, and why the APOM guiding principles (2026) add end-to-end ownership, under which the team is responsible for the product throughout its entire lifecycle.

Figure 5. *The empowered product team: the cross-functional team is assigned a problem to solve and is accountable for the outcome.*

9.1 Cross-functional and durable

Cross-functional means that the team has within it, or within easy reach, the product, design, and engineering skills it needs to take an idea from the problem to delivered value without depending on handoffs to other teams. SVPG sums up the ideal in five attributes: product teams are cross-functional, durable, informed by the strategic context, empowered, and accountable for outcomes.

Durable means that the team persists. Over time it builds up knowledge of the product, the domain, and the users, a capital that is lost every time a temporary team is disbanded when its project closes. That shared memory shortens each new learning cycle and allows the team to recognize sooner which ideas deserve to be tested.

9.2 Empowerment and accountability

Empowering a team means assigning it a problem and responsibility for the outcome, and trusting it with the decision about the solution. SVPG includes this among the

principles of product teams, which are given problems to solve and are empowered to find the best solution.

Empowerment is real when autonomy and accountability go together. Autonomy without accountability leaves the team without direction, and accountability without autonomy reproduces the trap of the feature team model, where the outcome is demanded of those who could not decide how to achieve it. A sense of ownership, experiencing the product as one's own, is the cultural consequence of that balance.

Central idea. Without empowered teams, the other elements of the model (outcomes over outputs, evidence-based governance, a learning culture) have nothing to rest on. That is why both reference formulations include them as a principle and treat them as the heart of the model.

9.3 End-to-end ownership

The APOM guiding principles, published by Scrum.org in 2026, count end-to-end ownership among their four core principles. According to that formulation, the product team owns the entire lifecycle, from ideation through production support, to minimize handoffs and delays.

The APOM puts this into practice in its Value Cycle, which brings together discovery, delivery, and support. Ideally the same team sustains all three, because that way it deepens its knowledge of the product and receives firsthand the signals of its real use. Handing production problems over to a separate support team lengthens the feedback loop and weakens the incentive to build well from the start.

As we will see in chapter 18, that ownership rests on a frequent and reliable delivery cadence, without which the team cannot learn from what it puts in users' hands.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Define an empowered product team by its traits: cross-functional, durable, and accountable for outcomes.
- Explain why the team's permanence preserves knowledge that temporary teams lose.
- Distinguish real empowerment from autonomy without accountability and from accountability without autonomy.
- Explain why the empowered team sits at the center of the model and which other elements depend on it.
- Describe end-to-end ownership and its effect on handoffs and feedback.
- Recognize when a “product team” is a feature team under another name.

Common mistakes and antipatterns

- **Feature team in disguise.** Calling a team “empowered” while it continues to be handed already decided solutions to build.
- **Autonomy without a measure.** Giving the team freedom without a clear measure of success, which turns empowerment into a lack of direction.
- **Ephemeral teams.** Setting up and disbanding teams for each initiative, with the loss of the accumulated knowledge of the product and the domain.
- **Build it and move on.** Handing the product over to a separate support team and washing one's hands of how it behaves in production.

Further reading

· SVPG — The Product Operating Model: An Introduction

· Scrum.org — Project to Product and the Agile Product Operating Model

· Scrum.org — APOM Guiding Principles Have Published

· SVPG — The Product Operating Model

BLOCK C · STRUCTURE: EMPOWERED PRODUCT TEAMS, ROLES, AND ORGANIZATIONAL DESIGN

Chapter 10. Roles and competencies of the model · ESTABLISHED

The model names a set of typical roles, but what matters is the competencies the team brings together, including the ability to use product tools with judgment, AI among them.

Introduction

A product team is defined by the competencies it brings together. The model names a set of typical roles (the product manager, the product designer, engineering with its tech lead, and, above them, the product leader), and its guiding principle is that what matters is the competencies the team needs, whatever the title of whoever provides them.

In this chapter we will see the place of each role in the model, why it is the operating model that defines the roles, and how AI reorders the weight of the competencies. The detailed craft of each role is a separate subject.

10.1 The roles and the four risks

Marty Cagan divides the team's responsibilities around the four risks that every solution must overcome before it is built. In SVPG's introduction to the product operating model, a solution has to be:

- **Valuable.** The customer buys it or chooses to use it.
- **Usable.** The user can figure out how to use it.
- **Feasible.** Engineering can build it with the team's time, technology, data, and skills.
- **Viable.** It works for the business, within its marketing, sales, or finance constraints.

According to SVPG, the product manager is responsible for the value and viability of what the team builds and is accountable for the outcomes; the product designer, for usability and the customer experience; and the tech lead, for the feasibility of the solution and its delivery. The product leader provides the strategic context in which they all decide, as we will see in chapter 11.

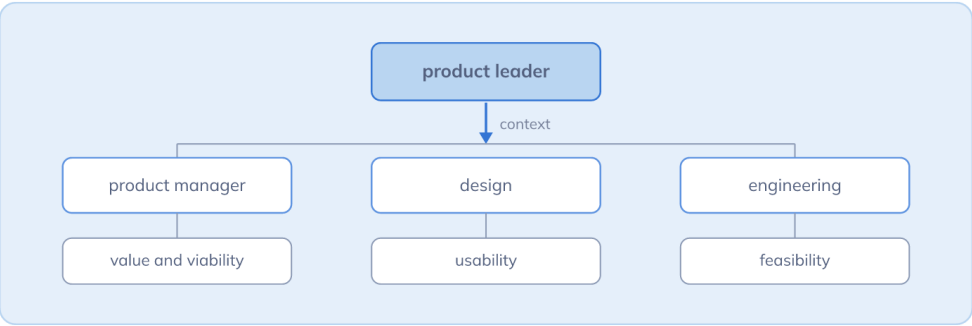


Figure 6. *The roles of the product team: the product manager is responsible for value and viability, design for usability, and engineering for feasibility, while the product leader provides the context.*

In small teams one person may cover more than one role, and in large ones a role may be shared among several people. What cannot be missing is the competencies, because a

team that lacks one will decide badly no matter how many titles it has covered. Cagan calls those who actively shape the product and tackle those risks product creators, and adds that their title hardly matters (May 2025).

10.2 The operating model defines the roles

In a November 2026 article on AI and the future of the product role, Cagan argues that it is the operating model with which a company creates products that determines the roles and responsibilities of its people. That is why the same title designates very different jobs in an empowered team and in a feature team.

From that premise he analyzes the effect of AI. As long as the bottleneck moves away from delivery, the focus remains on discovery, and that will keep increasing the demand for people with the knowledge and skills of the product manager of an empowered team. At the same time, he considers it clear that fewer feature team product managers will be needed, although he admits that nobody knows how many.

10.3 Competencies and using tools with judgment

In August 2026 Cagan proposed a fresh definition of the product role based on that of the analyst Benedict Evans, which describes three competencies of effective product people: seeing the underlying problem behind the pain the customer expresses, discovering a solution that customers value, and knowing the company well enough for that solution to be viable. Cagan picks up Evans's observation that AI changes the thresholds, but not the problem.

Among the competencies of product work is the ability to use the available tools with judgment, and today AI is the one that most reorders the weight of each competency. Part of the mechanical work becomes cheaper and judgment gains value: deciding which problem deserves attention, interpreting the evidence, and validating what the tool produces. AI comes in here as a context that shifts the emphasis among the competencies, without giving rise to a separate role.

Central idea. The question that matters is whether the team brings together the competencies to discover and deliver value autonomously, and it is answered by looking at the work the team does, whatever the distribution of job titles.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Name the typical roles of the product team and the main responsibility of each.
- Assign the four risks of a solution (value, usability, feasibility, and viability) to the role responsible for each.
- Explain why the model gives priority to competencies over titles.
- Relate the operating model to the actual content of a role and explain how AI shifts the demand for product managers.

- Assess whether a team brings together the necessary competencies, regardless of which positions it has filled.
- Describe how AI reorders the weight of the competencies without creating a separate role.

Common mistakes and antipatterns

- **Filling titles, not competencies.** Considering a team complete because it has the planned positions, even though it lacks a key competency.
- **AI as a role.** Treating the use of AI as a separate function rather than as a cross-cutting competency of the team.
- **Watertight roles.** Turning roles into silos that hand work off to one another, which breaks the team's cross-functionality.
- **Changing the title without changing the model.** Renaming the managers of feature teams as product managers without giving them problems to solve or responsibility for the outcome.

Further reading

· Marty Cagan (SVPG) — AI and the Future of the Product Role

· Marty Cagan (SVPG) — A Fresh Definition of The Product Role

· SVPG — The Product Operating Model

· SVPG — The Product Operating Model: An Introduction

· Marty Cagan (SVPG) — The Era of the Product Creator

BLOCK C · STRUCTURE: EMPOWERED PRODUCT TEAMS, ROLES, AND ORGANIZATIONAL DESIGN

Chapter 11. Product leadership · ESTABLISHED

The product leader leads by context and coaches people, senior leadership sustains the model, and leadership competence matters more the cheaper it becomes to build.

Introduction

If teams decide how to solve the problems they are assigned, leaders are left with a different and demanding task, which consists of providing context and developing people. Product leadership is exercised through coaching and leadership by context. The leader sets the direction and the constraints, helps their people grow, and removes obstacles, and senior leadership, all the way up to the CEO, has its own role in sustaining the model.

In this chapter we will see what leadership by context and coaching consist of, why Cagan argues that competent leadership gains importance as building becomes cheaper, and what role falls to senior leadership.

11.1 Coaching and leadership by context

Leading by context means giving each team the information, objectives, and constraints it needs to decide well, and trusting it to decide. Coaching is the other half of the job and consists of developing people's skills so that this trust is justified.

A leader who hands out solutions brings the feature team model back in through the back door, however good their intentions. The context they provide includes the product vision and strategy, as we saw in chapter 8, and the objectives the team must achieve.

Health signal. The best indicator of the health of product leadership is what it hands to the teams. When it hands them problems, expected outcomes, and context, the model works; when it hands them solutions and closed plans, it suffocates the model.

11.2 Experts lead experts

In September 2026 Cagan analyzed a widespread trend in the industry, that of having fewer managers and more people who build. Many engineering managers have gone back to individual contributor jobs, and one often hears that managers and other leaders will be much less important. Cagan considers this a necessary but temporary state.

His argument is that, when building becomes cheaper, the responsibilities of leadership weigh even more, and competent leadership becomes an ever greater competitive advantage. With more building capacity within reach of every team, someone has to make sure that this capacity is directed at the right problems and that people have the judgment to make the most of it. The title of his article, experts lead

experts, sums up the idea that whoever leads expert professionals must in turn master the craft.

Cagan has also proposed (February 2026) that product people use foundation models, configured with their company's context, as a personal product coach, because very few have a manager who is willing and able to coach them. For leaders, he recommends combining that tool with a good human leadership coach, and warns that using it requires challenging its answers and looking in them for criticism and mistakes.

11.3 The role of senior leadership

The operating model is a company-wide decision that no department can make on its own. That is why the sponsorship of senior leadership, and often of the CEO, is a condition for it to work. They are the ones who can change funding, governance, and incentives, the dimensions that a team does not control and that we will cover in chapters 13 to 15.

Without that sponsorship, teams are empowered on paper and trapped by a structure that does not change. Leadership's commitment shows in concrete decisions about budget, authority, and measurement, which are what make empowerment credible to the teams.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain what leading by context consists of and why it makes the team's empowerment possible.
- Describe the role of coaching in developing product people.
- Argue why competent leadership gains value when building becomes cheaper.
- Assess the use of AI as a personal product coach and its limits compared with human coaching.
- Argue why the sponsorship of senior leadership is a condition of the model.
- Recognize when a leader brings back the feature team model by handing out solutions.

Common mistakes and antipatterns

- **The leader who provides solutions.** Replacing context with detailed instructions, which empties the team's empowerment of meaning.
- **Empowerment without sponsorship.** Asking for empowered teams without leadership changing funding, governance, and incentives.
- **Delegate and disappear.** Confusing leading by context with absence, and leaving teams with neither direction nor development.

- **Doing without leadership because building is cheap.** Cutting managers in the belief that they are superfluous when there is more capacity to build, which is precisely when directing it is most needed.

Further reading

· Marty Cagan (SVPG) — Experts Lead Experts

· SVPG — The Product Operating Model

· Scrum.org — Shifting to a Product-Based Operating Model with APOM: The Accenture Way (podcast)

· Marty Cagan (SVPG) — Product Coaching and AI

BLOCK C · STRUCTURE: EMPOWERED PRODUCT TEAMS, ROLES, AND ORGANIZATIONAL DESIGN

Chapter 12. Organizational design: topologies, autonomy, and dependencies · ESTABLISHED

Teams are grouped around products and value streams with few team types and interaction modes, a bounded cognitive load, and as few dependencies as possible.

Introduction

Once the products and teams are defined, what remains is to decide how teams are grouped and how they interact. The model's organizational design relies on established frameworks, with *Team Topologies* as the common reference, to group teams around products and value streams, limit their cognitive load, and minimize the dependencies that slow down flow.

In this chapter we will see the team types and their interaction modes, what the second edition of the book (2025) adds, and the balance between autonomy and alignment. We deal with the structural role of these frameworks, without going into teaching them in detail.

12.1 Team types and interaction modes

Team Topologies, by Matthew Skelton and Manuel Pais, proposes four team types and three interaction modes. The types are the stream-aligned team, the platform team, the enabling team, and the complicated-subsystem team; the interaction modes are collaboration, X-as-a-Service, and facilitating.

The stream-aligned team is the primary type, because it delivers value end to end on a stream of work. The others exist to put it in the best possible conditions and reduce its cognitive load. The value of the framework lies in restricting variety, since four types and three interaction modes prevent the proliferation of custom team shapes.

12.2 What the second edition clarifies

The second edition of *Team Topologies* (IT Revolution, September 2025) keeps the four team types and the three interaction modes, and clarifies three ideas with direct consequences for the product operating model.

- **Cognitive load as a design principle.** What a team can effectively sustain in domain, technology, and coordination is limited, and team design starts from that limit.
- **Platform grouping.** A platform can be made up of several teams that offer internal services to the stream-aligned teams.
- **Value stream grouping.** Several teams working on the same value stream form a multi-team unit, which connects with the teams of teams of the next section.

12.3 Autonomy, alignment, and dependencies

Design seeks a balance between autonomy, by which each team decides and delivers on its own, and alignment, by which they all move toward the same strategy. The main lever is to minimize dependencies, because the fewer the waits between teams, the better value flows.

When the architecture does not allow full decoupling, the answer is teams of teams that share a product or a value stream, as we will see in chapter 25 when we discuss the boundary with scaling frameworks.

Conway's Law. Systems end up mirroring the communication structure of the organization that builds them, as the Scrum Manager BoK notes. Designing the teams therefore amounts to designing the system, and whoever ignores this ends up fighting against their own organization.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Describe the four team types and the three interaction modes of *Team Topologies*.
- Explain why the stream-aligned team is the primary type.
- Apply cognitive load as a criterion for bounding a team's scope.
- Distinguish platform grouping from value stream grouping, and relate both to teams of teams.
- Relate team design to product architecture through Conway's Law.
- Identify dependencies between teams and propose design changes that reduce them.

Common mistakes and antipatterns

- **Hybrids and custom types.** Inventing one's own team types instead of using the framework's four, which brings back the confusion the framework avoids.

- **A platform that controls.** Turning the platform into a control body instead of a service that reduces the cognitive load of the stream-aligned teams.
- **Unmanaged dependencies.** Tolerating permanent dependencies between teams as if they were inevitable, without redesigning to reduce them.
- **Ignoring cognitive load.** Assigning a team more domains, technologies, or services than it can sustain, and then blaming people for the slowness of the system.

Further reading

· Team Topologies — Key Concepts

· Matthew Skelton and Manuel Pais — Team Topologies, 2nd Edition

· Scrum Manager BoK — Conway's Law (in Spanish)

· Scrum Manager BoK — Structural agility (in Spanish)

BLOCK D · FUNDING AND GOVERNANCE BY VALUE STREAMS

Chapter 13. From project funding to product or value stream funding · CONSOLIDATING

The alternative consistent with the model is to fund products and value streams persistently, and to review the investment based on the evidence of outcomes.

Introduction

Funding is the dimension where most transitions run aground. An organization can declare empowered teams and durable products, but if money is still allocated by project, with scope, budget, and date, the financial structure drags it back to the project model. Funding by product or by value stream means persistently resourcing the teams that look after a product and reviewing that allocation based on their outcomes.

In this chapter we will see why project funding clashes with the model, what the alternative consists of, and why its adoption is still partial.

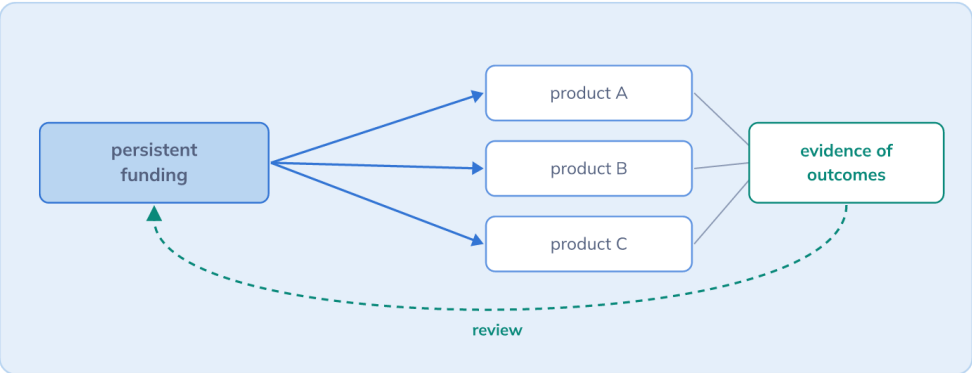


Figure 7. *Funding products instead of projects: money flows persistently to the products and their teams, and is reviewed against the evidence of outcomes.*

13.1 Why project funding clashes with the model

Project funding presupposes that the work has a known scope and an end. Three effects contrary to the product model follow from that premise.

- It forces people to estimate up front what can only be discovered by doing.
- It penalizes learning, because any change of course counts as a budget deviation.
- It disbands the teams when the project ends, and with them the accumulated knowledge of the product and the domain.

All three contradict the basic assumption of the product as a continuous and durable effort, which we saw in chapter 2.

13.2 Funding products and value streams

The alternative is to fund persistent teams and review the investment periodically according to the evidence of value, which we will study in chapter 17. Large initiatives are treated as bets that are confirmed or corrected with what is being learned.

The third installment of Scrum.org's “APOM State of Play” series, devoted to product portfolio management, includes this idea among its six principles with the formula “fund the products (teams), not the work.” Money is allocated to durable capabilities, and the team decides, within the strategy, which work to spend it on. We cover portfolio management in chapter 7.

The falling marginal cost of building, with AI among its causes, reinforces this logic. If building is ever cheaper, what is scarce and valuable is deciding well where to keep investing, and persistent funding with evidence-based review makes that decision easier, while project funding gets in its way.

Why it is consolidating. The principle is widely advocated and its adoption is still partial. Scrum.org's survey on the APOM, published in 2026 with responses from more than 130 organizations, shows silos and structural impediments that limit the teams' real autonomy, and no adoption data have appeared that would allow it to be considered established.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why project funding contradicts the assumption of the product as a continuous effort.
- Describe what funding products or value streams persistently consists of.
- Explain the principle of funding the products and their teams, and its consequences for who decides the work.
- Link the review of the investment to the evidence of value and decouple it from the milestones of the plan.

- Argue why cheaper building reinforces persistent funding.
- Recognize when the financial structure drags the organization back to the project model.

Common mistakes and antipatterns

- **A product funded like a project.** Resourcing “products” with scope-and-date budgets, which brings the project logic back in by way of money.
- **Reviewing by milestones, not by evidence.** Releasing or cutting funding according to progress against the plan rather than according to demonstrated value.
- **Reorganizing without refunding.** Changing the structure of the teams while leaving the budget allocation model untouched.
- **Funding the work under another name.** Approving funds initiative by initiative within each product, so that the team still cannot decide which work deserves the investment.

Further reading

· Scrum.org — Unlocking the APOM: Addressing the Core Barriers to Value Delivery

· Scrum.org — APOM State of Play, Part 3: 6 Guiding Principles for Product Portfolio Management

· SVPG — The Product Operating Model

BLOCK D · FUNDING AND GOVERNANCE BY VALUE STREAMS

Chapter 14. Lean budgeting and Beyond Budgeting ·

CONSOLIDATING

Budgeting in a way consistent with the model means allocating incrementally, over short and revisable horizons, with the principles of Beyond Budgeting and lean budgeting as the conceptual framework.

Introduction

If persistent products are funded, a way of budgeting is needed that is consistent with uncertainty and with the continuous cadence of product work. The principles of lean budgeting and Beyond Budgeting offer that conceptual framework, which consists of allocating incrementally, over short and revisable horizons, and abandoning the closed, negotiated annual budget.

In this chapter we will see the principles of Beyond Budgeting, how they apply to the budget of product teams, and why the topic is still consolidating. We deal with the principles at the level of concept, without going into the mechanics of any particular framework.

14.1 The Beyond Budgeting principles

Beyond Budgeting proposes separating what the annual budget mixes together (targets, forecasts, and resource allocation) and managing each adaptively, with relative and ambitious goals, a lean and unbiased forecasting process, and resources made available as they are needed.

The principles document of the Beyond Budgeting Institute lists twelve, six leadership principles and six management process principles.

- **Leadership principles.** Purpose, values, transparency, organization, autonomy, and customers.
- **Management process principles.** Targets, forecasts, resource allocation, performance evaluation, rewards, and coordination.

Each principle sets an adaptive practice against the usual one. Targets, for example, should be directional, ambitious, and relative, without fixed figures cascaded from above; management processes are organized dynamically around business rhythms and events, not around the calendar year only; and resources are made available as needed, with a cost-conscious mindset and without detailed annual allocations.

14.2 Lean budgeting applied to the product

Applied to the product model, these principles mean resourcing persistent teams, allocating funds over short horizons that are reviewed against the evidence, and preventing the budget from becoming a covert scope commitment.

The transition can be gradual. It is enough for allocation to stop relying on closed annual plans and to be guided instead by outcomes and learning. The evidence-based approach that Scrum.org associates with the APOM provides the value information against which that allocation is reviewed, as we will see in chapter 17.

Why it is consolidating. The principles are mature, since Beyond Budgeting was formulated more than two decades ago, and their adoption in product organizations remains uneven. The consolidating status reflects adoption, not conceptual novelty.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain which three things the annual budget mixes together and why it pays to separate them.
- List the twelve Beyond Budgeting principles in their two groups, leadership and management processes.
- Translate those principles into the budgeting of persistent product teams.
- Distinguish incremental, revisable allocation from budgeting by annual scope.
- Argue that the topic's consolidating status is due to its uneven adoption, with principles that are already mature.

Common mistakes and antipatterns

- **An annual budget in disguise.** Calling a closed annual budget “product funding” under another name.
- **A fixed target as both goal and reward.** Tying rewards to negotiated fixed figures, which encourages caution and haggling instead of value.
- **All or nothing.** Postponing any change because “you can't abolish the budget,” instead of adopting the principles incrementally.

Further reading

· Beyond Budgeting Institute — The 12 Beyond Budgeting principles (PDF)

· Scrum.org — The Agile Product Operating Model (APOM): an Evidence-Based Approach

BLOCK D · FUNDING AND GOVERNANCE BY VALUE STREAMS

Chapter 15. Governance, product authority, and constraints ·
CONSOLIDATING

Once the strategy is set, the model distributes product authority to the teams and governs by outcomes, with clear fundamental constraints and without the inherited ones that no longer protect against anything.

Introduction

Empowering teams immediately raises the question of governance, which is who decides what and where the limits lie. The model distributes product authority so that, once the strategy is set, the teams decide within clear constraints, with lightweight governance that verifies outcomes. It is the dimension where the transition is hardest, because it touches power.

In this chapter we will see how authority is distributed, what the APOM guiding principles (2026) propose for governance and compliance, and how to tell a fundamental constraint from an inherited one.

15.1 Distributing authority and lightweight governance

Authority over product decisions moves down to the teams once the strategy and the constraints are clear. With that, the role of general management and of finance changes, from approving plans and exercising stage-gate control to setting the context, ensuring the legitimate constraints, and reviewing outcomes.

Lightweight governance is still governance. It verifies with evidence what has been achieved and stops requiring approval for every step, so that control is exercised over the outcomes and autonomy is exercised over the path.

15.2 Governance in the APOM guiding principles

The governance and compliance chapter of the APOM guiding principles starts from a tension present in almost every product organization, where governance is seen as a brake that adds steps to delivery processes that are already complex. Its answer is to build governance into the operating model and treat it like any other requirement or constraint, described in terms of impact, outcomes, and value. It sets this out in three principles.

- **Outcome-based governance.** Governance and compliance requirements are defined as clear outcome goals, and the teams autonomously choose and adapt the most effective solution to meet them.
- **Shift governance left.** Controls are built into the discovery and delivery flow as automated checks and transparent quality gates, owned by the teams themselves, instead of waiting for sequential reviews at the end of the cycle.

- **Frequently inspect governance outcomes.** The empirical process is applied to governance itself, whose effectiveness is reviewed continuously so that structure and processes are adapted when evidence of friction or failure appears.

The APOM also separates the governance process, which is mandatory and controlled, from the work process, which belongs to those who do the work and helps them collaborate, streamline flow, and speed up delivery.

The same logic extends to procurement of products and services from third parties. To replace centralized, sequential purchasing, the APOM proposes agile, collaborative, and incremental procurement, with suppliers involved from discovery onward, contracts treated as living agreements that allow for change, and short cycles that deliver value early.

15.3 Fundamental and inherited constraints

Not all constraints have the same origin. Fundamental constraints, such as compliance, security, regulation, or real risks, are legitimate and must be respected. Inherited ones exist only because things have always been done that way, in the form of approval processes, templates, and controls that no longer protect against anything.

Treating an inherited constraint as fundamental is one of the main sources of friction and false impossibility in the transition. When governance is described in terms of outcomes, as the APOM proposes, each control has to justify the risk it reduces, and those that cannot justify it become visible.

Central idea. Distributing authority changes who decides, and that is why this transition runs into power dynamics, as we will see in chapter 24. Telling a fundamental constraint from an inherited one is the tool for gaining autonomy without taking on real risks.

Why it is consolidating. Distributing authority is the hardest part of the transition, and governance that is outcome-based and shifted left, as the APOM guiding principles call for, is a practice that is still settling in.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain how product authority is distributed once the strategy has been set.
- Describe the shift in the role of general management and finance toward lightweight governance.
- Explain the three APOM governance principles: outcome-based governance, shifting governance left, and frequently inspecting its outcomes.
- Reword a compliance requirement as an outcome goal that the team can address autonomously.
- Tell a fundamental constraint apart from an unjustified inherited one.
- Identify where power dynamics hold back the distribution of authority.

Common mistakes and antipatterns

- **Governance by approvals.** Keeping committees and approval gates that control every step instead of verifying outcomes.
- **An inherited constraint treated as law.** Treating inherited processes and controls as fundamental constraints, which blocks autonomy without reducing any real risk.
- **Empowering without ceding authority.** Announcing empowerment while decision-making authority remains concentrated at the top.
- **Governance at the end of the cycle.** Leaving compliance checks for a review after delivery, which delays value and uncovers problems when they are most expensive to fix.

Further reading

· Scrum.org — APOM Guiding Principles Have Published

· Scrum.org — APOM State of Play, Part 1: Fundamentals

· SVPG — The Product Operating Model

BLOCK E · OUTCOME ORIENTATION: METRICS AND EVIDENCE-BASED GOVERNANCE

Chapter 16. Objectives by outcomes: OKRs · ESTABLISHED

OKRs align the portfolio and the teams around outcomes, and give them focus and context without dictating the solution.

Introduction

OKRs (objectives and key results) are the most widespread mechanism for aligning an organization around outcomes. An objective expresses a qualitative direction, and the key results make it measurable. In the product model they connect the portfolio with the teams, because they translate the strategy into outcomes to achieve and give each team a measure of success without dictating the solution.

In this chapter we will look at the role of OKRs within the operating model, how portfolio OKRs fit together with those of each team, and how they differ from performance indicators. The technique of writing them is a subject in its own right, developed in the Skill Arena topic “OKRs and product strategy”.

16.1 OKRs as focus and context

Used well, an OKR is a tool for focus, because it forces a choice about what really matters over a short horizon and a renunciation of everything else. It is also context: it tells the teams the outcome being sought and leaves them to decide how to achieve it.

Used badly, it degenerates in two ways. The first is a task list that looks like a set of objectives, in which the key results are deliverables (“launch feature X”) and the OKR goes back to measuring output. The second is an instrument of top-down control, which reintroduces the micromanagement the model meant to leave behind.

Central idea. A key result describes a change in the customer or in the business. When it describes a deliverable, the OKR has stopped measuring outcomes, even if it keeps its format.

It is the same principle of outcomes over outputs that we saw in chapter 3, applied to the instrument with which the organization sets its goals.

16.2 Portfolio and team OKRs

In the portfolio, OKRs steer the bets and the work that cuts across several products, as we saw in chapter 7. In the team, they provide the measure of success for the problem it has been assigned. Scrum.org's APOM uses them as the link between the two levels, so that the portfolio sets the outcomes it pursues and each team formulates its own to contribute to them.

The key is vertical coherence without imposition. Team OKRs contribute to portfolio OKRs, but they are formulated with the participation of the team itself, which is closest to

the problem. If they reach the team already worked out, the OKR becomes a work order shaped like an objective.

16.3 OKRs and performance indicators

Nor are OKRs the same as key performance indicators (KPIs). A KPI monitors the steady operation of something that already exists, such as the availability of a service or the cost per customer, and it is tracked continuously. An OKR expresses the change sought over a specific horizon and is renewed when that change is achieved or stops being a priority. The two coexist, and a deteriorating KPI can inspire the objective of the next cycle.

How to write good objectives and key results, how to review them during the cycle, and how to link them with product strategy are covered in depth in the Skill Arena topic “OKRs and product strategy”. The OKR page of the Scrum Manager BoK provides the reference definition.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the role of OKRs as a mechanism for focus and context within the product operating model.
- Distinguish a key result that describes a change in the customer or in the business from a disguised deliverable.
- Describe how portfolio OKRs and those of each team fit together without top-down imposition.
- Differentiate an OKR from a key performance indicator.
- Recognize when OKRs degenerate into a task list or an instrument of control.
- Separate the role of OKRs in the operating model from the technique of writing them, which is a subject in its own right.

Common mistakes and antipatterns

- **OKRs as a task list.** Turning key results into committed deliverables loses the outcome orientation.
- **Top-down OKRs.** Imposing already worked-out OKRs on the teams turns them into an instrument of control and strips them of their value as context.
- **Too many objectives.** A long list of OKRs dilutes focus until it amounts to having none.
- **Confusing OKRs with KPIs.** Calling OKRs the indicators that monitor business as usual leaves the change being sought unexpressed.

Further reading

· Skill Arena — OKRs and product strategy

· Scrum Manager BoK — OKR (in Spanish)

· Scrum.org — The Agile Product Operating Model (APOM): an Evidence-Based Approach

· Scrum.org — The Agile Product Operating Model

BLOCK E · OUTCOME ORIENTATION: METRICS AND EVIDENCE-BASED GOVERNANCE

Chapter 17. Evidence-Based Management (EBM) and the four Key Value Areas · ESTABLISHED

Evidence-Based Management measures value in four Key Value Areas and treats management decisions as hypotheses to be tested against data.

Introduction

Evidence-Based Management (EBM) is Scrum.org's framework for governing by the value delivered. It proposes measuring what matters in four Key Value Areas and treating management decisions as hypotheses to be tested against evidence. It is a settled standard: its current guide dates from 2024 and has not changed since.

In this chapter we will look at the four areas, how empiricism is brought to the level of management, and why EBM is the evidence anchor of the Agile Product Operating Model (APOM).

17.1 The four Key Value Areas

EBM organizes measurement into four areas. Two describe market value, and the other two describe the organization's ability to deliver it:

- **Current Value.** The value the product delivers today to its customers and to the organization.
- **Unrealized Value.** The potential value not yet captured in the market, that is, the gap between what the product gives today and what it could give.
- **Time-to-Market.** The ability to deliver new value quickly and frequently, and to learn from each delivery.
- **Ability to Innovate.** The organization's ability to create new value, which is lost when effort is consumed by maintenance, technical debt, and friction.

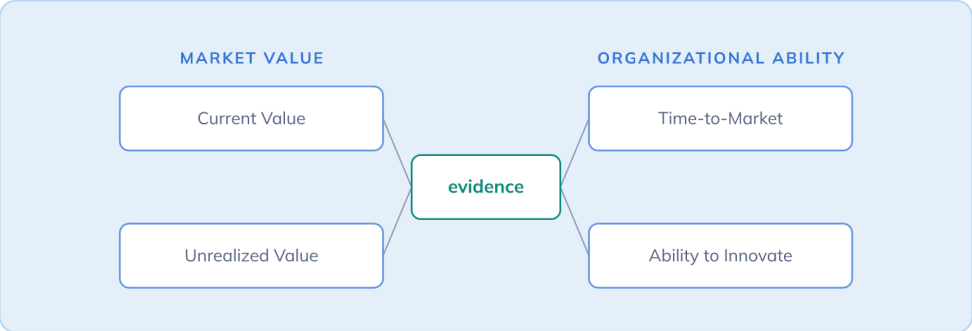


Figure 8. *The four Key Value Areas of EBM: two measure market value and two the organization's ability, and all of them are inspected with evidence.*

Four lenses at once. None of the four areas is enough on its own. Good Current Value with a slow Time-to-Market or a weak Ability to Innovate describes an organization living off past gains, and evidence-based governance looks at all four together.

17.2 Empiricism brought to management

Activity and the volume of output (how many features, how many points, how many releases) do not, on their own, tell you whether value is being delivered or whether the organization will be able to keep doing so. EBM shifts the focus to the four areas and treats each management decision as an experiment with four steps:

1. Formulate a value hypothesis and the measure that will test it.
2. Act, with the smallest step that allows learning.
3. Measure the effect on the Key Value Areas.
4. Adapt the decision to what the evidence says, whether that means continuing, correcting course, or abandoning.

The EBM Guide also arranges goals on three levels (a Strategic Goal, Intermediate Goals, and Immediate Tactical Goals), so that each experiment connects to a larger goal. It is the same empiricism as in product work, brought to the level of management and the portfolio.

17.3 The evidence anchor of APOM

EBM is the evidence anchor of APOM, which Scrum.org defines as a principles-based, evidence-backed approach. Since January 2026, its guiding principles have associated an evidence metric with each principle, and some point directly to EBM, such as the Current Value trend they propose for the principle of outcomes over outputs.

In this way, the logic of EBM extends from the product to the operating model itself, which is also governed with evidence. As we will see in chapter 19, those metrics serve to measure the health of the model.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Name and explain the four Key Value Areas of EBM.
- Group the four areas into market value and organizational capability.
- Explain why activity metrics do not, on their own, report on the delivery of value.
- Describe the empirical management cycle: hypothesis, action, measurement, and adaptation.
- Argue why the four areas must be looked at together.
- Relate EBM to APOM and to the evidence metrics of its guiding principles.

Common mistakes and antipatterns

- **Measuring activity.** Governing with production metrics, such as velocity or the number of releases, as if they reported on the value delivered.
- **Looking at a single area.** Optimizing Current Value while ignoring Time-to-Market or the Ability to Innovate leads to living off past gains.
- **Metrics without hypotheses.** Collecting indicators without a value hypothesis that gives meaning to how they are read and to the decision made with them.
- **Measuring without deciding.** A dashboard of the four areas that nobody uses to change decisions is activity under another name.

Further reading

· Scrum.org — Online Evidence-Based Management Guide

· Scrum.org — Evidence-Based Management

· Scrum.org — APOM Guiding Principles Have Published

· Scrum.org — The Agile Product Operating Model (APOM): an Evidence-Based Approach

BLOCK E · OUTCOME ORIENTATION: METRICS AND EVIDENCE-BASED GOVERNANCE

Chapter 18. Delivery cadence as a condition of the model · ESTABLISHED

Delivering in small, frequent, and reliable steps makes empirical learning and governance by outcomes possible, although on its own it does not guarantee them.

Introduction

The product model rests on the ability to deliver value in small, frequent, and reliable steps. Without it there is no empirical learning and no governance by outcomes, because if delivery takes months, the loop of hypothesis and evidence breaks. That is why delivery cadence is treated here as a condition of the model, and its technique is left to engineering.

In this chapter we will look at the cadence the sources propose, the conditions that make it reliable, how it is measured, and why delivering faster, with or without AI, is not enough to get better outcomes.

18.1 Small, frequent, and reliable

The introduction to the Product Operating Model that SVPG published in February 2023 sets the benchmark at releasing at least every two weeks, and ideally continuously through continuous integration and continuous deployment (CI/CD). The figure is indicative and expresses a principle with three components:

- **Small batches**, which reduce the risk of each change.
- **Frequency**, which shortens the learning loop.
- **Reliability**, which makes each release a routine event, free of surprises.

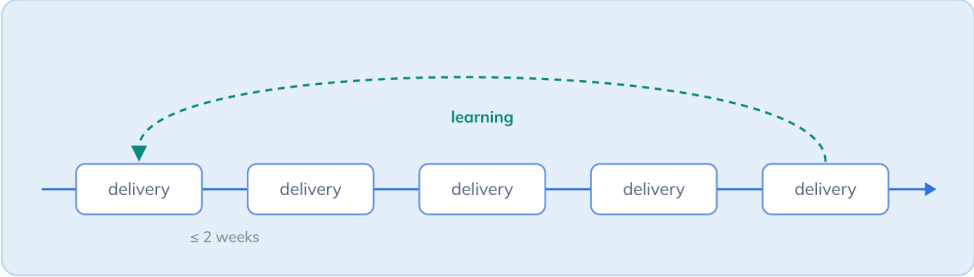


Figure 9. *Delivery cadence: small, frequent deliveries that close a learning loop about real value.*

An organization that can only release twice a year cannot operate by product, however much it reorganizes its teams.

18.2 Instrumentation, observability, and phased deployment

The same SVPG introduction lists the conditions that make the cadence reliable:

- **Instrumentation**, to know whether and how the product is used.
- **Monitoring and observability**, to detect problems early.

- **Phased deployment**, with tools that allow A/B testing and other practices that limit the impact of each change.

The model treats them as a condition, because without them the cadence is not reliable. Their detailed technique (continuous integration and deployment, automated testing, observability) belongs to engineering and falls outside the scope of the topic.

18.3 How delivery is measured

The DORA research program offers the most widely used reference for measuring software delivery performance. It proposes five metrics grouped into two families:

- **Throughput.** Change lead time, deployment frequency, and failed deployment recovery time.
- **Instability.** Change fail rate and deployment rework rate, which measures the proportion of deployments that are unplanned and made to address an incident in production.

For the operating model, these metrics show whether the condition is met. They say nothing about the value of what is delivered, which is measured with the EBM Key Value Areas covered in chapter 17.

18.4 A necessary but not sufficient condition

AI has made this nuance more visible. The 2025 DORA report, devoted to AI-assisted software development, finds that AI improves delivery throughput and also increases instability. The added speed comes at a cost when reliability does not keep pace.

Marty Cagan supplies the other half of the argument (SVPG, July 2026). He observes that most organizations use AI to accelerate their old project-based way of working, and he points out that the problem with the project model was never its slowness, because it is designed to produce output. AI makes building easier, but the hardest part is still knowing what to build, so delivering faster without better discovery does not improve outcomes.

Central idea. Without a frequent and reliable cadence, empiricism, evidence-based governance, and reviewable funding stay on paper. With it, value still depends on deciding well what to build and on measuring what changes.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why delivery cadence is a condition of the model and why it is not enough on its own.
- Justify the value of small batches, frequency, and reliability, and place the benchmark of releasing at least every two weeks.
- Relate cadence to the empirical learning loop and to evidence-based governance.

- Name the conditions that sustain a reliable cadence (instrumentation, observability, and phased deployment) and place their technique in engineering.
- Identify the five DORA metrics and distinguish the throughput metrics from the instability metrics.
- Explain why accelerating delivery with AI, without improving discovery or reliability, does not improve outcomes.

Common mistakes and antipatterns

- **Reorganizing without speeding up delivery.** Creating product teams that keep releasing infrequently breaks the learning loop.
- **Delivery as a big event.** Treating each deployment as a high-risk event discourages releasing often.
- **Cadence without observability.** Releasing fast without instrumentation or observability turns speed into a risk that nobody measures.
- **Using AI to accelerate business as usual.** Producing faster without improving discovery multiplies output and instability without bringing outcomes any closer.
- **Taking DORA metrics for value metrics.** They measure the ability to deliver; whether what is delivered adds value is checked with other measures.

Further reading

· SVPG — The Product Operating Model: An Introduction

· DORA — DORA's software delivery performance metrics

· Marty Cagan (SVPG) — The AI Productivity Paradox

· SVPG — The Product Operating Model

BLOCK E · OUTCOME ORIENTATION: METRICS AND EVIDENCE-BASED GOVERNANCE

Chapter 19. Portfolio measurement and health of the model · CONSOLIDATING

Measuring the portfolio and the health of the model makes it possible to tell an organization that truly operates by product from one that has only changed its vocabulary.

Introduction

Beyond measuring the value of each product, the model needs to know whether the organization itself is truly operating by product or has only changed its vocabulary. Measuring the health of the model and of the portfolio is a more recent and less standardized practice than EBM, and that is why it is consolidating: there is agreement that it is needed, but not yet on how to do it.

In this chapter we will look at what portfolio measurement adds to the Key Value Areas, which signals distinguish real change from cosmetic change, what APOM proposes with its evidence metrics, and what precautions measuring the model itself requires.

19.1 Portfolio metrics

At the portfolio level, what matters is how investment is distributed among products, what outcomes each bet produces, and whether the whole is advancing toward the strategic objectives. The four Key Value Areas (chapter 17) offer a starting point that can be aggregated, and portfolio measurement adds questions of its own:

- **Exploitation and exploration.** The balance between making the most of what exists and exploring what is new.
- **Dependencies.** How much work cuts across several products and requires coordination from the portfolio.
- **Reallocation.** The actual speed at which resources move when the evidence calls for it.

19.2 Signs that an organization truly operates by product

The most useful signal is a set of indications, because no single number captures the model. Four questions serve as a test:

- whether teams are assigned problems or solutions;
- whether funding is persistent or by project;
- whether investment is reviewed on evidence or on milestones;
- whether the teams decide or the committees approve.

An organization can have the whole product vocabulary and still operate by project in practice. Measuring the health of the model consists of detecting that gap between rhetoric and practice so that it can be closed.

Rhetoric and practice. The characteristic risk of transitions is cosmetic change, with new names and old behaviors. Measuring the health of the model serves above all to make it visible.

19.3 An evidence metric for each principle

The APOM guiding principles, published by Scrum.org in January 2026, are the most concrete proposal to date. They associate one or more evidence metrics with each principle to check whether it is being applied, and so offer a practical way to measure and adapt the operating model itself. Here are some examples:

- the percentage of discovery work with explicit hypotheses, for the principle of continuous discovery;
- the percentage of decisions that teams make without escalating them, for that of empowered teams;
- the percentage of strategic initiatives explicitly tied to a business objective or a customer outcome, for that of starting with why;
- the stability of product teams over time, for that of separating talent management from work management;
- the EBM Current Value trend, for that of outcomes over outputs.

The proposal is recent and has not yet become a reference instrument. Scrum.org's APOM survey (2026, with more than 130 organizations) lists measurement among the open challenges, and the commercial maturity assessments that exist, such as Planview's, do not constitute a standard. That is why measuring the health of the model is consolidating.

19.4 Measuring without distorting what is measured

The APOM guiding principles add a caution about flow metrics. They help locate bottlenecks and impediments, and they should stay in the hands of the product teams, without serving as indicators for individual targets.

Another principle extends the caution to all process metrics. Their legitimate use is transparent diagnosis and the continuous improvement of flow, and using them to rank the performance of individuals or teams encourages local optimization and the manipulation of data.

The same applies to the health signals of the model. Turned into a punitive dashboard, they invite people to massage the indicators, and the gap between rhetoric and practice is hidden just when it was meant to be made visible.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why the health of the model needs to be measured in addition to the value of each product.
- Describe the questions of its own that portfolio measurement adds to the EBM Key Value Areas.
- Identify the signs that an organization operates by product and distinguish them from cosmetic change.
- Give examples of evidence metrics from the APOM guiding principles and relate each one to the principle it tests.
- Justify that flow and process metrics be used only by the product teams, without linking them to individual targets or rankings.
- Argue why measuring the health of the model is not very standardized and is consolidating.

Common mistakes and antipatterns

- **Cosmetic change.** Taking the transition for done because the vocabulary has been adopted, without measuring whether behaviors changed.
- **Only product metrics.** Measuring the value of each product without measuring the health of the model makes it impossible to see whether the organization operates by product.
- **Flow metrics as individual targets.** Using them to evaluate or rank individuals and teams encourages local optimization and the massaging of data.
- **Taking a commercial assessment for a standard.** Presenting a vendor's maturity model as a recognized measure of the health of the model ignores that the practice is not yet standardized.

Further reading

· Scrum.org — APOM Guiding Principles Have Published

· Scrum.org — Unlocking the APOM: Addressing the Core Barriers to Value Delivery

· Scrum.org — Evidence-Based Management

BLOCK F · CULTURE OF PRODUCT WORK

Chapter 20. Product mindset, trust, and psychological safety

ESTABLISHED

Product culture puts outcomes ahead of dates and rests on the why, on transparency with stakeholders, and on the trust that makes autonomy possible.

Introduction

Culture is the dimension that decides whether the rest of the model takes root or stays on paper. A product mindset puts outcomes ahead of meeting dates, learning ahead of certainty, and ownership of the problem ahead of carrying out orders. That mindset needs a foundation of trust and psychological safety, without which empowerment remains nominal.

In this chapter we will look at how that culture is built from the why and from transparency, what role trust and psychological safety play, and how they are reflected in the guiding principles with which Scrum.org has formulated its Agile Product Operating Model (APOM) since 2026.

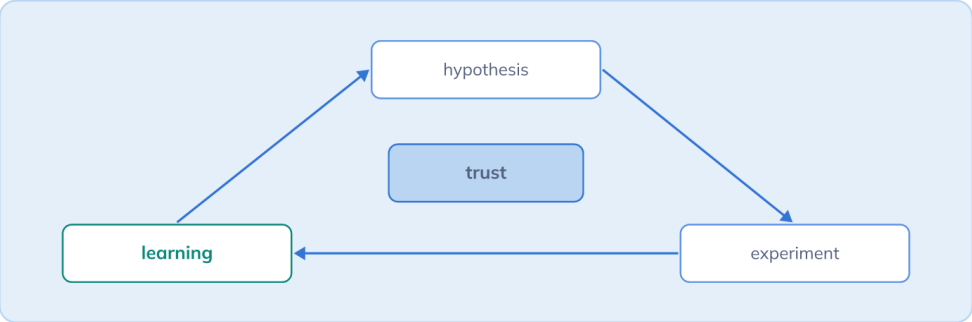


Figure 10. *An empirical culture: hypothesis, experiment, and learning in a continuous cycle sustained by trust.*

20.1 Starting with why, and transparency

A product culture starts by explaining the why. Teams that understand the reason for what they do make better decisions and take ownership of the outcome, because they can judge each option against the end it pursues. Transparency, which consists of sharing information, data, and the reasoning behind decisions, makes that why possible and allows context to take the place of control.

The relationship also works in reverse. An opaque organization cannot empower, because empowering requires putting within the team's reach what it needs to decide. If the strategy, the constraints, or the usage data stay with management, the team decides blindly or pushes the decision back up, and the declared autonomy stays on the org chart.

Transparency also extends to stakeholders. Showing them the hypotheses, the progress, and what has been learned, including what has not worked, is the basis of the collaboration between product and the business that we will see in chapter 22.

20.2 Trust and psychological safety

Autonomy rests on trust in two directions. Management trusts that the team will decide well if it has the right context, and the team trusts that it can say what it thinks, raise doubts, and admit mistakes without reprisals. That second condition is psychological safety, and the learning covered in chapter 21 depends on it, because where admitting a mistake is punished, mistakes are hidden and the organization stops learning from them.

The Scrum Manager BoK covers the concept on its page on psychological safety, linked at the end of the chapter, which is worth reading as a complement to this section.

20.3 Culture in the APOM guiding principles

Product culture and psychological safety as the basis of empowerment are a firm consensus in the literature, and Scrum.org's formulation makes this explicit. The APOM guiding principles, whose first version was published in January 2026, include trust and empowerment, and collaboration and transparency, as principles in their own right.

- Trust and empowerment consist of delegating decision-making authority and full ownership to small, cross-functional teams, aligned directly with product outcomes and with transparent strategic objectives.
- Collaboration and transparency consist of making all objectives, constraints, assumptions, and learnings visible, so that decisions are made collaboratively across teams and inspection and adaptation are fast.

Each guiding principle has an associated evidence metric, so culture too can be inspected with data, as we saw when discussing the health of the model in chapter 19.

Culture as the foundation. An organization can copy the structure, funding, and metrics of another that operates by product and still fail if it lacks the culture that sustains them. Culture is not the decoration of the model but its foundation.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Characterize a product mindset as opposed to a project mindset.
- Explain why starting with why improves the quality of decisions and ownership of the outcome.
- Relate transparency, including with stakeholders, to the real possibility of empowering a team.
- Argue why psychological safety is a condition of organizational learning.
- Identify the APOM guiding principles that cover trust and empowerment, and collaboration and transparency.
- Recognize the symptoms of an opaque or punitive culture that empties empowerment of content.

Common mistakes and antipatterns

- **Empowering without transparency.** Asking teams to decide while withholding the information and reasoning they need to do so.
- **Punishing mistakes.** Penalizing honest failures teaches people to hide them and shuts down learning.
- **A culture of dates.** Rewarding adherence to the schedule above the outcome reinstalls the project mindset.
- **Declared trust, practiced control.** Proclaiming the autonomy of the teams while keeping a prior approval for every significant decision.

Further reading

· SVPG — The Product Operating Model

· Scrum.org — APOM Guiding Principles Have Published

· Scrum Manager BoK — Psychological safety (in Spanish)

· Scrum.org — APOM State of Play, Part 1: Fundamentals

BLOCK F · CULTURE OF PRODUCT WORK

Chapter 21. Learning, experimentation, and error management

· ESTABLISHED

An empirical culture makes discovery the norm and experimentation a core competency, and treats cheap, early mistakes as information.

Introduction

The product model starts from the premise that nobody knows in advance what will work, and so it discovers before committing. The culture that sustains it treats learning as the norm and early, cheap mistakes as valuable information. It is the cultural counterpart of Evidence-Based Management and of delivery cadence, which we saw in chapters 17 and 18.

In this chapter we will look at discovery as the usual way of working, the place of experimentation when artificial intelligence makes building cheaper, and how to manage mistakes so that the organization learns from them.

21.1 Discovery as the norm

Discovery means validating ideas at a controlled risk before investing in building them at scale, whether by talking to users, testing prototypes, or launching small experiments. In a product culture it is a usual way of working that continues even when there is a rush. Blindly building what someone assumed is precisely what cheaper building makes more dangerous.

Discovery and delivery coexist in the same team and move forward in parallel, the approach the Scrum Manager BoK describes as dual-track discovery and delivery. The APOM guiding principles (2026) carry it into the culture with a direct formula, treat all work as an experiment, and call for measuring the impact of each release against the defined outcomes to feed continuous learning cycles.

21.2 Experimentation as a core competency

In October 2026 Marty Cagan clarified the place of experimentation in the model. Organizations that work by project also experiment, mainly to optimize what already exists. In the product model, experimentation stops being an optimization function and becomes the core competency of an organization trying to discover solutions worth building.

Cagan also stresses that people learn by building. The emphasis is on learning fast, and to that end teams build many prototypes for the sole purpose of testing ideas before building the product. Artificial intelligence makes those prototypes cheaper and faster, and in doing so it shifts the difficulty, because the hard part becomes deciding what to build.

Hence a warning about method. Cagan picks up an idea from Tom Chi: understanding why something worked matters much more than knowing whether it worked, because only the explanation allows what was learned to carry over to the next decision. For products that incorporate AI, he proposes dividing the tools, with evals (systematic assessments of the quality of the responses) to assess quality early and A/B tests focused on establishing value.

21.3 Error management and continuous improvement

Tolerating controlled failure consists of designing so that mistakes are small, early, and reversible, and of extracting from each one the learning it holds. Continuous improvement closes the cycle, because every experiment, whether it succeeds or fails, leaves the organization knowing something it did not know before.

All of this rests on the psychological safety we saw in chapter 20. If admitting a mistake carries a personal cost, the mistake is hidden and the learning is lost. That is why the analysis of a failure aims to understand what happened and what to change in the system, without looking for culprits, and the APOM principles also propose recognizing the willingness to experiment, even when what is learned forces a change of course.

A cheap mistake is information. Mistakes are certain to happen. What an organization can decide is whether they will be small, early, and reversible, and whether it will learn from them or hide them.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why discovery is a usual way of working that continues even under pressure.
- Explain the role of experimentation as a core competency of the product model and differentiate it from optimization experiments.
- Argue why, when AI makes building cheaper, the difficulty shifts to deciding what to build.
- Distinguish between knowing whether something worked and understanding why it worked, and justify why the latter better guides the next decision.
- Assign evals and A/B tests their respective roles in validating a product that uses AI.
- Design experiments so that mistakes are small, early, and reversible.
- Relate error management to psychological safety and continuous improvement.

Common mistakes and antipatterns

- **Skipping discovery in a hurry.** Treating validation as a luxury that is dropped under pressure and building blindly.

- **Experimenting only to optimize.** Reducing experimentation to fine-tuning what already exists, without using it to decide what is worth building.
- **Stopping at whether it worked.** Closing an experiment with the test result without understanding why it came about.
- **Heroic failure.** Confusing tolerance of mistakes with taking large, irreversible risks from which nothing is learned.
- **Blame-seeking failure analysis.** Turning the examination of a failure into an allocation of blame guarantees that the next mistake will be hidden.

Further reading

· Marty Cagan (SVPG) — Experimentation, AI and the Product Model

· Scrum Manager BoK — Product discovery (in Spanish)

· Scrum Manager BoK — Dual-track discovery & delivery (in Spanish)

· Scrum.org — APOM Guiding Principles Have Published

· SVPG — The Product Operating Model: An Introduction

· Scrum.org — Evidence-Based Management

BLOCK F · CULTURE OF PRODUCT WORK

Chapter 22. The relationship with stakeholders and with “the business” · ESTABLISHED

Technology and the business solve problems together, and cultural objections are handled knowing that many of them stem from the demand for predictability.

Introduction

The product model breaks the customer–supplier relationship between “the business” and “technology.” When technology works as an internal service, the business asks and technology serves; in the product model, both solve problems together and share responsibility for the outcome. Changing that relationship is one of the deepest cultural changes and one of those that arouse the most resistance.

In this chapter we will look at what collaboration between product and the business consists of, why the demand for predictability lies at the root of many objections, and how those objections are handled day to day.

22.1 Beyond “the business asks, technology serves”

The critique of technology as an internal service is central to Marty Cagan's formulation. When technology merely executes requests, nobody owns the outcome, and what was requested gets built even if it does not solve the problem. The feature teams of companies that work by project are organized, in Cagan's words, to serve the business.

Real collaboration means that product teams take part in understanding the problem and in deciding the solution. Stakeholders contribute the context and constraints of their area, whether legal, financial, commercial, or brand-related, and leave the design of the solution to the team. A closed specification deprives the team of exactly the room it needs to discover what works.

22.2 Predictability, the root of many objections

This change clashes with entrenched expectations: stakeholders used to commissioning features, areas that measure their power by the volume of their pending requests, and executives who expect commitments to scope and date.

In October 2026 Cagan put a name to what lies behind the last of those expectations. If you ask executives and stakeholders whether they want predictability or outcomes, they will most likely answer that they need both. Yet most companies in fact choose predictability, even if implicitly, and treat outcomes as the expected consequence of delivering what was promised.

Cagan also observes that predictability commitments are frequently missed, that executives, stakeholders, and customers know it, and that the volume of output then becomes the fallback measure of progress. This closes the circle that brings the organization back to the project model.

Recognizing the demand for predictability as the root of the objection to abandoning scope and date commitments changes the conversation. It makes it possible to treat the objection as a legitimate need to know what to expect and when, which the product model meets with transparency about progress toward outcomes, with frequent releases, and with commitments reserved for what truly requires them.

22.3 Handling cultural objections

Handling these objections is part of the cultural work. It requires listening to them, distinguishing legitimate concern from resistance to losing power, and showing results that make the new relationship credible. We will look at the catalog of objections to the transition, with their answers, in chapter 24.

Shared responsibility. There is a simple indicator that the customer–supplier relationship has been left behind. When something does not work, the organization asks what it has learned, not who asked for it.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain what technology as an internal service consists of and why it prevents ownership of the outcome.
- Describe the collaborative relationship between product and the business in the product model.
- Distinguish between contributing context and constraints and handing over closed specifications.
- Explain why the demand for predictability lies at the root of the objection to abandoning scope and date commitments.
- Identify typical cultural objections and separate legitimate concern from resistance to the shift in power.
- Use the shared responsibility question to diagnose the relationship between product and the business.

Common mistakes and antipatterns

- **A business that dictates solutions.** Keeping stakeholders who commission specific features instead of posing problems and outcomes.
- **A product organization that only executes.** Accepting the role of internal supplier and building what was requested without answering for the outcome.
- **Requests as power.** Measuring an area's influence by the volume of its pending requests, which encourages asking for more instead of achieving more.
- **Asking for outcomes and rewarding predictability.** Declaring that outcomes are wanted while evaluating teams on meeting the committed scope and date.

Further reading

· Marty Cagan (SVPG) — Predictability vs Outcomes

· SVPG — The Product Operating Model

· Scrum Manager BoK — Cultural agility (in Spanish)

· SVPG — The Product Operating Model: An Introduction

BLOCK G · THE TRANSITION TO THE PRODUCT OPERATING MODEL

Chapter 23. The transition: holistic nature and incremental journey · ESTABLISHED

Relabeling what already exists as product fails; the transition moves structure, funding, governance, metrics, and culture at the same time, and advances product by product.

Introduction

The transition to a product model is holistic and incremental. It is holistic because the five dimensions of the model (structure, funding, governance, metrics, and culture) have to move together, and changing only one of them creates friction. It is incremental because it advances product by product and at different speeds, in a continuous journey that does not end in a final state.

In this chapter we will see why relabeling without changing the substance fails, how the journey is sequenced, and why artificial intelligence makes the transition more urgent.

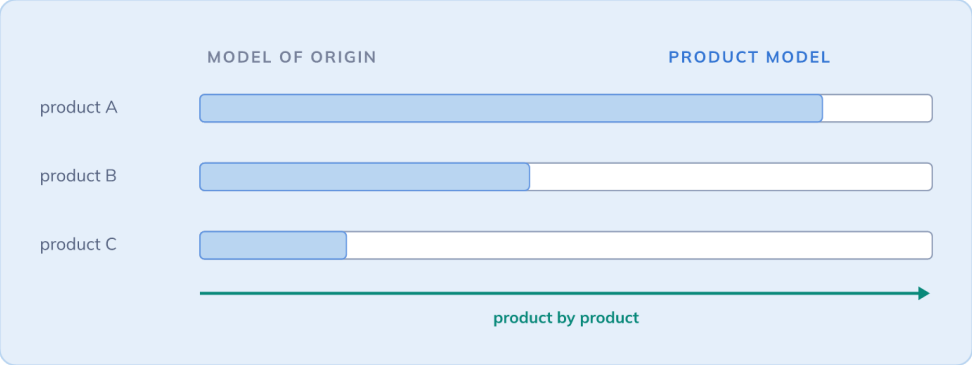


Figure 11. *An incremental transition: the organization moves to the model product by product, and each group advances at its own pace.*

23.1 Why relabeling without changing fails

The best-documented failure of transitions consists of relabeling what exists as "product" without touching the structure, the funding, or the incentives. The teams are called product teams, but they are funded by project, decisions remain in committees, and delivering the plan is what gets rewarded. The label changes and the behavior stays the same.

Hence the holistic nature of the transition. The model only works if the five dimensions change coherently, because each one reinforces or neutralizes the others. An empowered team that is still evaluated on meeting dates, for example, soon goes back to behaving like a project team.

23.2 The incremental journey

The recognized path is to sequence: start with some products, learn, and extend what has been learned. Each product group can advance at a different speed depending on its maturity and its context, and the transformation is understood as a journey that goes on. Managing it as a project, with its scope, its plan, and its end date, would repeat precisely the model the organization wants to leave behind.

Holistic and incremental at the same time. Holistic means that, wherever there is progress, the five dimensions progress together, without trying to change the whole company at once. Progress is made area by area, and completely within each area.

23.3 AI makes the transition more urgent

Artificial intelligence has added urgency to this diagnosis. Mik Kersten, author of *Project to Product* and *Output to Outcome*, summed it up in April 2026 in words quoted by IT Revolution: “With the Product Operating Model, AI makes it faster to do the right things. With the Project Operating Model, AI makes it faster to do the wrong things.”

Other authors reach the same conclusion. In July 2026 Marty Cagan observed that most companies use AI to accelerate their project-based way of working, that is, to produce the artifacts of the old model faster. In August 2026 Roman Pichler argued that, without the right product operating model, AI delivers limited benefits, and that taking advantage of it requires redesigning workflows.

For the transition, the consequence is twofold. Adopting AI tools without changing the model multiplies the production of more of the same and aggravates the problems described in this chapter. And redesigning workflows with AI is one more reason to move all five dimensions together in each area being transformed.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why the transition is holistic and what happens when only one dimension changes.
- Describe why relabeling projects as products without changing funding, governance, and incentives fails.
- Argue for the incremental and continuous nature of the transformation.
- Distinguish between advancing through complete areas and advancing through isolated dimensions across the whole company.
- Recognize when the transition is being managed as a project with an end date.

- Explain why AI accelerates the results of a product organization and aggravates the problems of an organization that works by project.

Common mistakes and antipatterns

- **Relabeling and declaring victory.** Changing the names to “product” and considering the transition done without touching funding, governance, or incentives.
- **Transformation as a project.** Managing the change with scope, a plan, and an end date, repeating the model the organization wants to move beyond.
- **Changing everything at once.** Trying to transform the whole organization at the same time instead of advancing through complete areas and learning from each one.
- **AI without a change of model.** Investing in AI tools to produce faster within the project model and expecting the results of the product model from it.

Further reading

· Scrum.org — APOM State of Play, Part 1: Fundamentals

· Mik Kersten — Output to Outcome

· IT Revolution — What Is a Product Operating Model, and Why Your AI Investment Depends on It

· Roman Pichler — Product Operating Model: The Crucial Enabler for AI Success

· Marty Cagan (SVPG) — The AI Productivity Paradox

· SVPG — The Product Operating Model

BLOCK G · THE TRANSITION TO THE PRODUCT OPERATING MODEL

Chapter 24. Leading the change, objections, and power dynamics · ESTABLISHED

The transition requires sponsorship and personal transformation from senior leadership, prepared answers to the foreseeable objections, and recognition of the power dynamics.

Introduction

No transition to product succeeds without leadership of the change. The dimensions that a team does not control, such as funding, governance, or incentives, can only be moved by senior leadership, and the distribution of authority that the model requires touches power dynamics that generate resistance.

In this chapter we will see the sponsorship and personal transformation of leaders, the usual objections with their answers, and the power dynamics, always in what is specific to this model and without going into change management as a general discipline.

24.1 Sponsorship and the personal transformation of leaders

Sponsorship from the CEO and senior leadership is a necessary condition, because they are the ones who authorize change in the structural dimensions. Authorizing is not enough, and the leaders themselves often have to transform the way they lead, moving from control through approvals to leadership by context, as we saw in chapter 11.

Transformation teams, cross-functional and with a real mandate, help to sustain the journey. Their role is to enable the product teams to carry out the transformation themselves, and their risk is becoming a silo that “does the transformation” to everyone else.

24.2 Objections and answers

In *Transformed*, Marty Cagan devotes an entire section to stakeholder objections and their answers, because anticipating them is part of the work. Objections are predictable, they repeat from one organization to another, and they mix legitimate concerns with resistance to the shift in power. Three of the most common concern predictability, spending control, and commitments to customers.

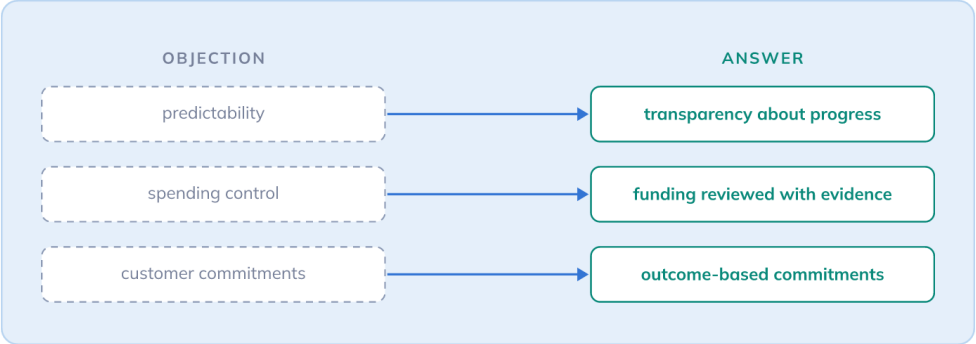


Figure 12. *Objections and answers: the usual objections to the transition, such as predictability, spending control, or commitments to customers, are answered with transparency, evidence, and a different way of committing.*

- Predictability. “You can't plan like this” expresses the legitimate need to know what to expect and when, the same demand we saw in chapter 22 as the root of many objections. It is answered with transparency about progress toward outcomes and with frequent deliveries that show real progress.
- Spending control. “We will lose control of the money” arises from the shift from approving projects to funding persistent teams. It is answered with evidence, because funding is reviewed regularly according to the outcomes obtained, as we saw in chapter 13, and leadership retains the ability to reallocate it.
- Commitments to customers. “Our customers demand dates” reflects real obligations, especially in contracts and custom development. It is answered with a different way of committing, which reserves scope and date commitments for what the customer really needs and offers continuous visibility and frequent deliveries for everything else.

Anticipating the objection. Having an honest answer ready for each objection, one that separates the real concern from resistance, is part of the work of leadership.

24.3 Power dynamics

A delicate case is divesting from a product whose owner is influential, even when the data advise doing so. There, what blocks the decision is the power dynamic, however clear the analysis may be, and recognizing that political dimension is part of leading the change.

Cagan himself has acknowledged it. In September 2026 he admitted that for years he “tried to ignore the importance of politics, hoping that strong product work would be enough to carry the day,” and called that attitude naive. Today he considers that “politics is a reality, especially at larger companies,” and part of the fabric of nearly every company.

The Scrum.org podcast on Accenture's shift to a product-based operating model, linked at the end of the chapter, offers the perspective of a large organization that has gone through this transition.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why senior leadership sponsorship is a condition of the transition.
- Describe the personal transformation that the model demands of the leaders themselves.
- Place the role of transformation teams and their risk of becoming a silo.
- Answer the objections about predictability, spending control, and commitments to customers with transparency, evidence, and a different way of committing.
- Distinguish, within an objection, the legitimate concern from resistance to the shift in power.
- Recognize the power dynamics that block decisions even when the analysis is clear.

Common mistakes and antipatterns

- **Transformation without leadership.** Driving the change from the bottom up without real sponsorship from those who control funding, governance, and incentives.
- **Transformation team as a silo.** Creating a group that "does the transformation" to everyone else instead of enabling the teams.
- **Ignoring politics.** Treating objections only as problems of analysis, without recognizing the power dynamics that sustain them.
- **Dismissing the objection.** Treating every objection as resistance and ignoring the legitimate concern it often contains.

Further reading

· Marty Cagan (SVPG) — Strong Opinions, Loosely Held

· SVPG — The Product Operating Model

· Scrum Manager BoK — Executive involvement (in Spanish)

· Scrum.org — Shifting to a Product-Based Operating Model with APOM: The Accenture Way (podcast)

BLOCK G · THE TRANSITION TO THE PRODUCT OPERATING MODEL

Chapter 25. The boundary with scaling frameworks ·

CONSOLIDATING

SAFe, LeSS, and Nexus can be an instantiation layer of the operating model; the model states the what and the why, and the scaling framework, one possible form of the how.

Introduction

A recurring question is how the product operating model fits with agile scaling frameworks such as SAFe, LeSS, or Nexus. The answer that is taking hold is that they operate at different levels. The operating model states the what and the why, that is, how the company must be organized to operate by product, and a scaling framework is one possible form of the how, an instantiation layer.

In this chapter we will look at that boundary without teaching the frameworks, which the Scrum Manager BoK describes in its pages on agile scaling models, SAFe, LeSS, and Nexus, and at the test that tells a framework that serves the model apart from one that only borrows its name.

25.1 Scaling as an instantiation layer

A scaling framework can be one way of implementing a product operating model in a large organization. It helps to organize teams of teams, to coordinate dependencies, and to provide a common cadence when the architecture does not allow the teams' work to be fully decoupled.

The relationship between APOM and SAFe has been raised by Yuval Yeret, a Professional Scrum Trainer at Scrum.org and a recognized SAFe expert, in a Scrum.org podcast from April 2025. Yeret presents SAFe as a way of implementing or instantiating APOM and warns that it has gaps, above all its focus on features, which pushes outcomes into the background. It is an expert's reading in a podcast, and Scrum.org has not formulated it as an institutional position.

The risk that this warning points to is well known. A misused framework can reproduce the feature factory at scale, coordinating the delivery of more features across more teams without bringing the organization any closer to outcomes.

25.2 What distinguishes the model from the framework

Adopting SAFe does not turn an organization into a product organization if it keeps funding by project, measuring outputs, and deciding in committees. The operating model is the criterion, and the framework is a tool that can serve it or betray it.

The boundary has become blurrier because some frameworks already present themselves as an operating model. In June 2026 Scaled Agile published AI-Native SAFe under the title of “the operating model for the AI-native organization,” and describes it

with a portfolio that steers funding toward outcomes and with outcomes expressed as OKRs from the portfolio down to the teams.

The test of this chapter also applies to a framework that calls itself an operating model. What decides whether an organization operates by product is whether funding, metrics, and governance really change, regardless of the name of the framework it adopts. For LeSS and Nexus, no recent changes in their relationship with the operating model are on record.

The fact that framing scaling as an instantiation layer is recent, and that the frameworks themselves are redefining their role, is what keeps this boundary in consolidation.

The framework serves the model, not the other way around. The question worth asking is what operating model is wanted and which instantiation, framework included, serves it best. Choosing the framework first reverses that order and is the route by which a framework ends up reproducing what it was meant to move beyond.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the relationship between the operating model and scaling frameworks as different levels.
- Describe scaling as a possible instantiation layer of the model.
- Attribute the relationship between APOM and SAFe precisely to Yuval Yeret and to its context, a Scrum.org podcast.
- Apply to any framework, including those that present themselves as an operating model, the test of whether funding, metrics, and governance really change.
- Recognize when a scaling framework reproduces the feature factory at scale.
- Explain why this boundary is consolidating.

Common mistakes and antipatterns

- **Framework equals model.** Believing that adopting SAFe, LeSS, or Nexus turns the organization into a product organization.
- **Trusting the label.** Accepting a framework because it presents itself as an operating model without checking what changes in funding, metrics, and governance.
- **A framework that scales the factory.** Using scaling to coordinate the delivery of features on a large scale without changing funding, metrics, or governance.
- **Choosing the framework before the model.** Deciding on the scaling framework before defining which operating model is wanted, reversing the right order.
- **Mistaking an opinion for an institutional position.** Presenting what an expert suggests in a podcast as the official stance of an organization.

Further reading

· Scrum.org — Agile Product Operating Model (APOM) and SAFe: Do They Relate? (podcast)

· Scaled Agile — AI-Native SAFe Is the Operating Model for the AI-Native Organization

· Scrum Manager BoK — Agile scaling models (in Spanish)

· Scrum Manager BoK — SAFe (in Spanish)

· Scrum Manager BoK — LeSS (in Spanish)

· Scrum Manager BoK — Nexus (in Spanish)

· Scrum.org — The Agile Product Operating Model

© 2026 Scrum Manager®. This work is published under a Creative Commons Attribution-NonCommercial 4.0 International licence (CC BY-NC 4.0). Official Scrum Manager trainers and centres are licensed under the terms of CC BY 4.0 for their training activity.