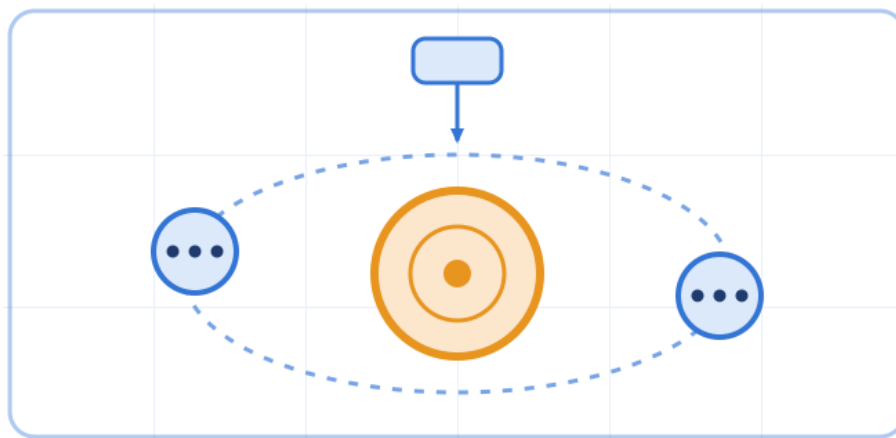


Guide

Agile product operating model

In-depth development of the State of the art topics



Updated June 2026

About this document

This document is an in-depth development of the topics in the reference map (State of the art) for operating an organisation 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 —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 right now: the book Transformed (Cagan and SVPG) is from 2024 and Scrum.org's Agile Product Operating Model, from 2024-2025, is still evolving. The cheaper cost of development brought by AI does not change these principles; 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 1 · 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 2 · 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 3 · Structure: empowered product teams, roles and organisational design

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

Block 4 · Funding and governance by value streams

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

Block 5 · 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 6 · 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 7 · 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 1 · 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 about structure, funding, governance, metrics and culture with which an organisation builds and operates products on a continuous basis. It is not a methodology or a process, but a set of principles.

Introduction

An operating model describes how an organisation organises 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. Talking about an operating model "by product" means that the organisation is structured around durable products —and the teams that look after them— rather than around temporary projects or isolated functions.

It is worth establishing from the outset what kind of object this is. It is not a framework to be rolled out step by step, nor a methodology with ceremonies and artefacts. It is a set of principles about how the organisation should operate; each company instantiates them in its own way.

Understanding it this way avoids the most common mistake: looking for "the recipe" for the product operating model, which does not exist.

1.1 The five dimensions it orders

The operating model touches 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) and culture (what behaviours and values are expected). The rest of the guide runs through these five dimensions. Their coherence is what is decisive: changing only one —for example, asking for empowered teams but keeping project funding— produces friction and, often, the failure of the transition.

Core idea. A product operating model is not something a team "does"; it is how the whole company is organised so that product work is possible. That is why it is a responsibility of management, not only of the teams.

1.2 What it is not: neither methodology nor prescribed process

The two reference formulations agree on this point. Cagan insists that the model is a set of principles and warns against "process addiction": confusing following ceremonies with operating well.

Scrum.org stresses that the model is owned by those who operate it and that it admits many valid instances. The practical consequence is that two companies can both operate "by product" with very different structures, cadences and tools.

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

This topic deals with the operating model —how the company is organised—, not the craft of product (discovery, prioritisation, writing OKRs, research techniques), which is a separate subject and appears here only in its place within the model. Nor is it a course on organisational change

management as a general discipline: of the transition, only what is specific to this model is addressed. And it is distinguished from the scaling frameworks (SAFe, LeSS, Nexus), which are a different layer of instantiation, dealt with 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 name the five dimensions it orders.
- Distinguish an operating model from a methodology or a prescribed process.
- Argue why the five dimensions must change coherently and not separately.
- Locate the boundary of the model against the craft of product, change management and scaling frameworks.
- Recognise when an organisation 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 roll out, instead of as principles that each organisation instantiates to its context.

Changing a single dimension. Asking for empowered teams while keeping project funding, activity metrics or phase-gate governance; the incoherence cancels out the change.

Confusing craft and model. Mixing discovery or roadmapping techniques with the design of the operating model, which operates at a different level.

Further reading

- [SVPG · The Product Operating Model](#)
- [Scrum.org · Agile Product Operating Model](#)

BLOCK 1 · FOUNDATIONS AND FRAMING OF THE PRODUCT OPERATING MODEL

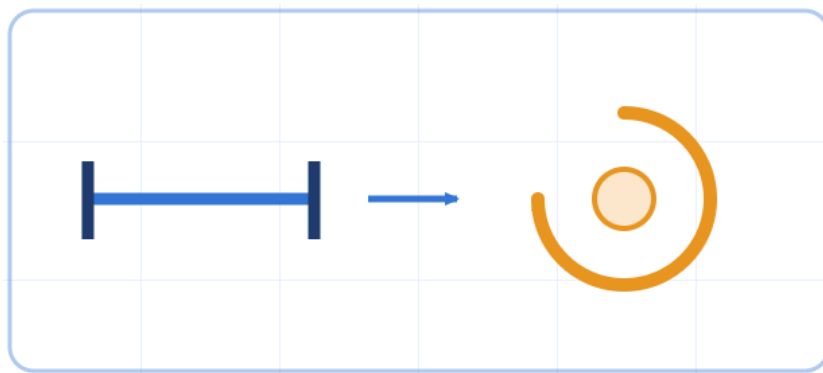
Chapter 2. From project to product: the paradigm shift · ESTABLISHED

The move from the originating models —"IT", "project", "feature team"— to the product model, understood as a continuous and enduring effort rather than a temporary initiative with a beginning and an end.

Introduction

Most organisations that adopt a product model come from one of three originating models. The "IT model" treats technology as an internal supplier that fulfils requests from the business. The "project model" organises work into initiatives with a scope, a budget and an end date. The "feature-team model" keeps stable teams but hands them lists of features to build. All three share an assumption: the work ends.

The product model breaks that assumption. A product is a continuous and enduring effort: it is not "delivered and closed", it is looked after and evolves for as long as it provides value. That change of frame drags everything else along with it: who decides, how it is funded and what teams are held accountable for.



From the project —a finite effort with a beginning and an end— to the product as a continuous and enduring effort.

2.1 The originating models and their limits

Each originating model has a reasonable internal logic, and that is why it persists. The problem appears when it is applied to digital products that live for years and compete in fast-changing markets. The project model, for example, optimises the delivery of a fixed scope on time and on budget, but it penalises learning: any deviation from the plan is experienced as a failure, even when that deviation is precisely what the market was asking for.

2.2 What changes when moving to product

Three things move at the same time. First, the unit of organisation: from the temporary project to the persistent product. Second, accountability: from meeting a plan to achieving outcomes. Third, the relationship with time: 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 first and most common failure of the transition, developed in chapter 23.

It is not just terminology. Moving to product is not about changing the name of the teams. If funding, accountability and cadence remain those of the project, the organisation is still operating by project even if it says otherwise.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Describe the three originating models (IT, project, feature team) and their internal logic.
- Explain what assumption they share and why it clashes with durable digital products.
- Identify the three things that change when moving to product: unit, accountability and relationship with time.
- Detect when an organisation has "renamed" projects as products without changing the substance.
- Explain why the product is understood as a continuous effort and not as an initiative with an end.

Common mistakes and antipatterns

Product with an end date. Keeping the project mindset under the new name: planning the product as a single delivery with a finish.

Business that requests, technology that serves. Retaining the customer-supplier relationship of the IT model, which prevents shared accountability for outcomes.

Success = plan met. Continuing to measure success by adherence to the plan instead of by the outcome achieved, penalising learning.

Further reading

- [SVPG · The Product Operating Model](#)
- [Scrum.org · Project to Product and APOM](#)

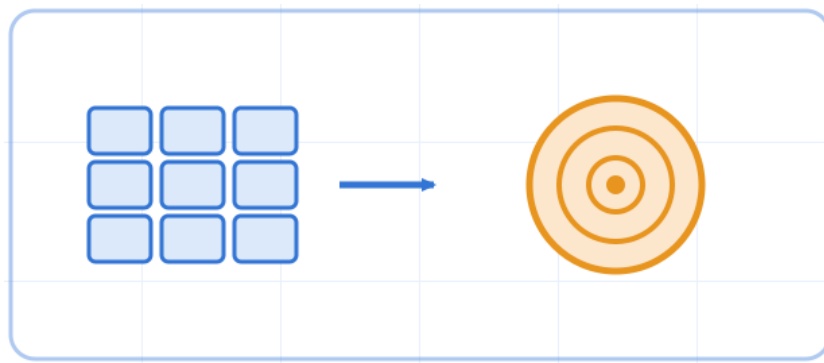
BLOCK 1 · FOUNDATIONS AND FRAMING OF THE PRODUCT OPERATING MODEL

Chapter 3. Outcomes over outputs · ESTABLISHED

The overarching, cross-cutting principle: assign teams problems to solve and outcomes to achieve, not lists of features to build.

Introduction

Output is what is built (features, releases, lines of code); outcome is the change that this produces in the customer and in the business. The "outcomes over outputs" principle says that teams should be held accountable for outcomes, not for output. It is the principle that best distinguishes the product model from a "feature factory", where success is measured by the quantity delivered.



Governing towards an outcome to be achieved, not towards the volume of outputs produced.

3.1 Assign problems, not solutions

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. This contrasts with the feature model, where the solution is already decided and all that is left for the team is to build it. Assigning outcomes does not mean abandoning the team: it means giving it context, constraints and a measure of success, and leaving it the "how".

3.2 Why AI reinforces this principle

As producing output becomes cheaper and faster —and AI tools are today the most visible example—, the risk is not building too little, but building fast what provides no value. Cagan sums it up as AI making it possible to "do the wrong things faster". The cheaper the output, the more it matters to govern towards the outcome: the bottleneck shifts from the ability to build to the ability to decide what is worth building.

The acid test. If a team can declare success having delivered everything planned but without moving any value metric, the organisation still measures output, not outcome.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Distinguish output from outcome with concrete examples from their own context.

- Reframe a brief expressed as a list of features in terms of problem and outcome.
- Explain why measuring output leads to the "feature factory".
- Argue why the cheapening of output by AI increases, not reduces, the importance of the outcome.
- Define an outcome-oriented measure of success for a given initiative.

Common mistakes and antipatterns

Feature roadmap. Governing work with a list of deliverables committed by date instead of with outcomes to achieve.

Speed as value. Taking the quantity or speed of delivery as proof of success, ignoring whether the output moved any value metric.

Decorative outcome. Stating an outcome in the presentation but continuing to evaluate the team by the delivery of the planned scope.

Further reading

- [SVPG · The Product Operating Model: An Introduction](#)
- [Scrum.org · APOM, evidence-based approach](#)

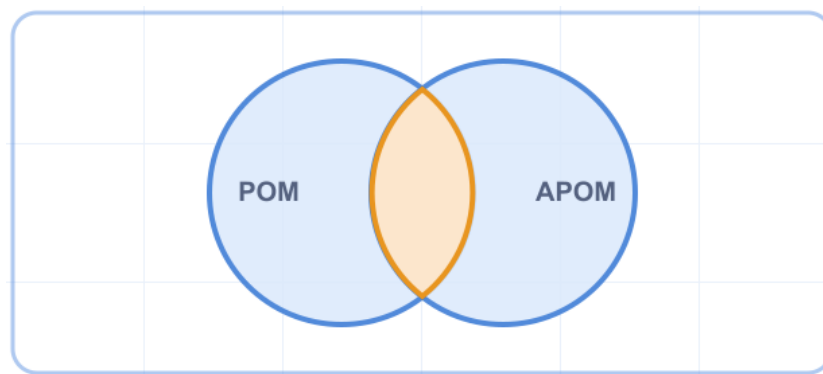
BLOCK 1 · FOUNDATIONS AND FRAMING OF THE PRODUCT OPERATING MODEL

Chapter 4. The two formulations and their divergence · CONSOLIDATING

Cagan/SVPG's Product Operating Model and Scrum.org's Agile Product Operating Model: a common core and points of divergence, including that of the "agile" label itself.

Introduction

The topic is not unified under a single label. Two reference formulations coexist that share a broad common core and diverge in emphasis, in anchoring and even in name. Knowing both avoids two symmetrical mistakes: believing there is a single canonical model and believing they are different things. They are two articulations of the same underlying shift towards product.



Two formulations with a broad common core and their own zones of divergence.

4.1 The Product Operating Model (Cagan / SVPG)

Formulated by Marty Cagan and SVPG, above all in the book *Transformed* (2024), it is a model of principles articulated in three dimensions: how you build (continuous and reliable delivery), how you solve problems (discovery, empowered teams) and how you decide which problems to solve (product strategy). Cagan deliberately avoids the adjective "agile" and maintains a critical stance towards "process addiction".

4.2 The Agile Product Operating Model (Scrum.org)

Scrum.org formulates APOM on Professional Scrum, Nexus and Evidence-Based Management. It describes it through seven characteristics: Unique to each organisation, Holistic, Evidence-based, made up of Empowered Teams, Empirical, Complete, and with Change Management built in. It is a younger formulation and still evolving.

4.3 Common core and divergences

Both share the essentials: product over project, outcomes over outputs, empowered teams, evidence-based governance and continuous delivery. They diverge in anchoring (APOM rests on Scrum/Nexus/EBM; the POM does not), in the weight of process (Cagan is wary of it) and in the label: the adjective "agile" belongs to one of the formulations, not to the consensus. This guide captures the common core and points out the divergences where they appear.

Why it is consolidating. The explicit pairing of both formulations is recent and APOM, in particular, is still evolving: new instalments, surveys and possible adjustments to its seven characteristics.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Describe the three dimensions of Cagan/SVPG's POM.
- List the seven characteristics of Scrum.org's APOM.
- Identify the common core that both formulations share.
- Explain the points of divergence, including that of the adjective "agile".
- Locate a concrete practice or claim within the formulation it comes from.

Common mistakes and antipatterns

A single canonical model. Presenting one of the two formulations as "the" product operating model, ignoring the other.

Confusing the label with the consensus. Treating the "agile" of APOM as part of the common core when it is specific to that formulation.

Mixing without attribution. Combining elements of both without pointing out where each one comes from, losing the traceability of the divergences.

Further reading

- [SVPG · The Product Operating Model](#)
- [Scrum.org · APOM State of Play \(I\)](#)
- [Scrum.org · The context of APOM and survey](#)

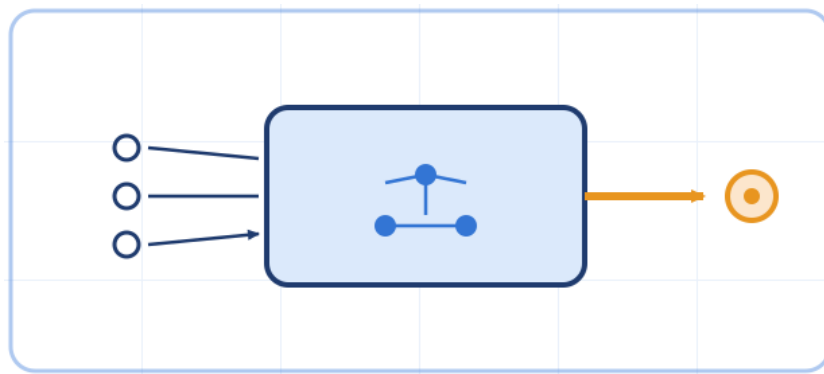
BLOCK 2 · THE PRODUCT AS AN OPERATING UNIT AND THE PORTFOLIO

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

The characteristics of a product —clear boundaries, recognised users and stakeholders, measurable value— and its distinction from project, feature, service or platform.

Introduction

If the model is organised "by product", the first question is what counts as a product. The answer is not trivial: on it depend the boundaries of the teams, the allocation of funding and the distribution of authority. A product, in this model, has clear boundaries, recognisable users and stakeholders and a value that can be measured.



A product: clear boundaries, recognised users and measurable value as its output.

5.1 The characteristics of a product

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

5.2 Product versus project, feature, service or platform

It is worth not confusing the product with neighbouring concepts. A project is temporary; a product is continuous. A feature is a part of the product, not a product in itself. A service can be a product if it has its own users and value. A platform is a particular type of product —internal— whose "customers" are other teams. When something does not fully fit the notion of a product, the model is applied with nuances rather than forced.

The boundary matters. Drawing a product's boundaries badly creates dependencies and overlaps between teams. The definition of a product is, in practice, a decision of organisational design (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.
- Distinguish a product from a project, a feature, a service or a platform.
- Assess whether a concrete effort can be treated as a product and with what nuances.
- Explain why the definition of a product conditions the boundaries of the teams and the funding.

Common mistakes and antipatterns

Feature promoted to product. Treating as an independent product what is a part of another, multiplying teams and dependencies.

Product with no 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 limits between products, a source of overlaps and authority conflicts.

Further reading

- [SVPG · The Product Operating Model](#)
- [Scrum.org · APOM State of Play \(I\)](#)

BLOCK 2 · THE PRODUCT AS AN OPERATING UNIT AND THE PORTFOLIO

Chapter 6. Taxonomy and topology of products · ESTABLISHED

Customer-facing products, internal products, platforms and enablers, and the principle that each product can have its own operating model.

Introduction

Not all products are the same, and forcing a single mould onto all of them is a mistake. A simple taxonomy helps to reason: customer-facing products, internal products, platforms and enablers. Each type has different users, risks and rhythms, and can therefore operate differently.

6.1 Types of product

Customer-facing products serve external users and compete in a market. Internal products serve users within the organisation itself (for example, operations tools). Platforms are products whose customers are other teams: they reduce their cognitive load by offering capabilities as a service. Enablers solve cross-cutting needs (security, data, identity). The distinction is not bureaucratic: it guides how the value of each one is measured.

6.2 Each product, its own operating model

A principle that both formulations capture is that each product can have its own operating model within a common framework. A mature, regulated product and a product in an exploration phase do not need the same cadence, governance or metrics. The organisation sets the common constraints (compliance, security, funding); within them, the model of each product is decided by those who operate it.

Platform as a product. Treating the platform as a product —with its internal users, its value proposition and its measurement— is what stops it from becoming a standardisation project disconnected from those who use it.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Classify products into customer-facing, internal, platforms and enablers.
- Explain why each type can measure its value differently.
- Argue the principle that each product can have its own operating model.
- Identify the common constraints that the organisation sets above each product.

Common mistakes and antipatterns

Single mould. Imposing the same cadence, governance and metrics on all products regardless of their nature.

Platform as a project. Managing the platform as a standardisation initiative without treating the consuming teams as customers.

Invisible enabler. Not recognising cross-cutting enablers as products, leaving them with no owner and no measurement.

Further reading

- [Scrum.org · Agile Product Operating Model](#)
- [Scrum.org · APOM in practice \(Accenture\)](#)

BLOCK 2 · THE PRODUCT AS AN OPERATING UNIT AND THE PORTFOLIO

Chapter 7. Product portfolio management · CONSOLIDATING

How products are aligned, funded and prioritised at the portfolio level: the "bets" between solutions and the work across products. It is not the roadmap of a product, but the governance of the set.

Introduction

Portfolio management operates one level above the individual product: it decides which products to invest in, how resources are distributed between them and how efforts that cut across several products are coordinated. It is not the roadmap of a specific product —that is craft—, but the governance of the set: where the organisation places its bets.

7.1 From portfolio planning to bets

In a product model, the portfolio is governed with outcomes, not with lists of projects. Portfolio planning starts by setting objectives —frequently through OKRs at the portfolio level (chapter 16)— and allocating persistent resources to the products that can best contribute to those objectives. Large initiatives are treated as bets: investments with risk, reviewable in light of the evidence, not closed commitments.

7.2 Work across products

Much of the value arises at the intersection of several products. Portfolio management coordinates that work across products without reintroducing project management: it does so by aligning common objectives and minimising dependencies, not by creating coordination committees that take authority away from the teams. It is one of the practices being formalised the most and where organisations vary the most, hence its consolidating state.

Portfolio by outcomes. If the portfolio is still governed as a list of projects to be funded and controlled by phases, the organisation operates by project at scale, even if its individual 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 instead of by a list of projects.
- Describe the concept of a "bet" and its reviewable nature in light of evidence.
- Identify how to coordinate work across products without reintroducing project management.
- Recognise the role of portfolio OKRs in the alignment between products.

Common mistakes and antipatterns

Portfolio = list of projects. Continuing to govern the set as a portfolio of initiatives with scope and date, phase-gate control included.

Coordination committee. Coordinating work across products by creating layers of governance that take authority away from the teams.

Irreversible bet. Treating large initiatives as closed commitments instead of investments reviewable in light of the evidence.

Further reading

- [Scrum.org · APOM, evidence-based approach](#)
- [Scrum.org · Agile Product Operating Model](#)

BLOCK 2 · THE PRODUCT AS AN OPERATING UNIT AND THE PORTFOLIO

Chapter 8. From strategic context to product strategy · ESTABLISHED

How business strategy comes down to product strategy and to the teams: vision, focus, insights and transparency as context that guides without dictating the solution.

Introduction

Empowered teams need direction, not instructions. That direction arrives as strategic context: a vision, a focus, insights and a transparency that allow each team to decide well without anyone dictating the solution to it. This chapter deals with the place of strategy in the model, not the techniques of formulating it, which belong to the craft.

8.1 Strategy as context, not as a mandate

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 is as important as vision: a strategy that wants everything guides nothing. Transparency —that teams see the logic of the decisions, not only the decisions— is what makes context replace control.

8.2 Insights and focus

Insights —about the market, the data, the technology and the customers— are the material with which strategy is built and what is shared with the teams so that they decide with sound judgement. Coming down with strategy is not handing out tasks, but handing out understanding: when a team understands the why, it can adapt the how in the face of what it discovers, which is precisely the behaviour the model seeks.

Context over control. The clearer the strategic context, the less direct control is needed. Where context is missing, the organisation tends to reintroduce micromanagement to make up for 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.
- Argue why focus is an essential part of strategy.
- Describe the role of insights and transparency in empowerment.

Common mistakes and antipatterns

Strategy that wants everything. Stating a strategy with no focus, which does not allow teams to prioritise or to say no.

Handing down tasks, not context. Handing out a list of work instead of the understanding that lets the team decide the how.

Decisions with no why. Communicating decisions without their logic, which forces teams to execute without being able to adapt.

Further reading

- [SVPG · The Product Operating Model](#)
- [SVPG · The Product Operating Model: An Introduction](#)

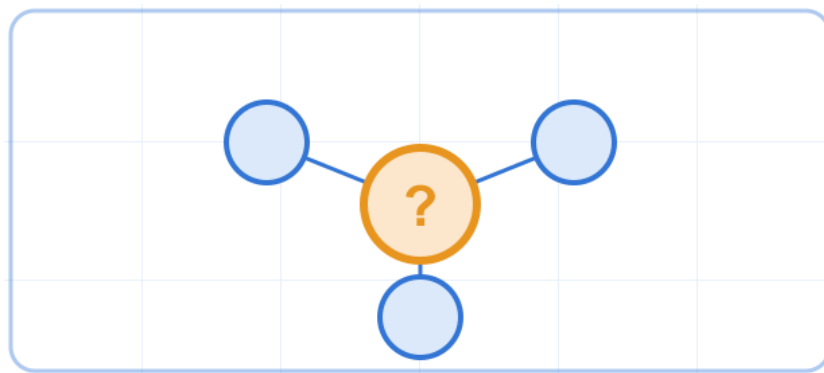
BLOCK 3 · STRUCTURE: EMPOWERED PRODUCT TEAMS, ROLES AND ORGANISATIONAL DESIGN

Chapter 9. Empowered product teams · ESTABLISHED

The cross-functional, durable team to which problems are assigned and which is held accountable for outcomes, with empowerment, a sense of ownership and collaboration. It is, according to Cagan, the most fundamental concept of the model.

Introduction

The empowered product team is the central piece of the model. It is a cross-functional team —it brings together the competencies needed to discover and deliver—, durable —it is not assembled and dismantled by project— and accountable for an outcome, not for a list of tasks. Cagan describes it as the most fundamental concept of the whole product model.



The cross-functional team is assigned a problem to solve, not a list of features.

9.1 Cross-functional and durable

Cross-functional means that the team has within it —or accesses without friction— the product, design and engineering competencies it needs to take an idea from the problem to the delivered value, without depending on handovers to other teams. Durable means that it persists: it accumulates knowledge of the product, the domain and the users, capital that temporary teams destroy at each dissolution.

9.2 Empowerment and ownership

To empower a team is to assign it a problem and accountability for the outcome, and to entrust it with the decision about the solution. Empowerment is real only if the team has both autonomy to decide and accountability for the outcome; autonomy without accountability is abandonment, and accountability without autonomy is the trap of the feature model. The sense of ownership —feeling the product as one's own— is the cultural consequence of that empowerment.

The piece that holds up the rest. 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 it is the most fundamental concept.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Define an empowered product team by its three traits: cross-functional, durable and accountable for outcomes.
- Explain why durability preserves knowledge that temporary teams destroy.
- Distinguish real empowerment from autonomy without accountability and from accountability without autonomy.
- Argue why this is the most fundamental concept of the model.
- Recognise when a "product team" is really a feature team in disguise.

Common mistakes and antipatterns

Feature team in disguise. Calling "empowered" a team that is still handed already-decided solutions to build.

Autonomy with no measure. Giving the team freedom without a clear measure of success, which turns empowerment into a lack of direction.

Ephemeral teams. Assembling and dissolving teams by initiative, destroying the accumulated knowledge of the product and the domain.

Further reading

- [SVPG · The Product Operating Model](#)
- [Scrum.org · Project to Product and APOM](#)

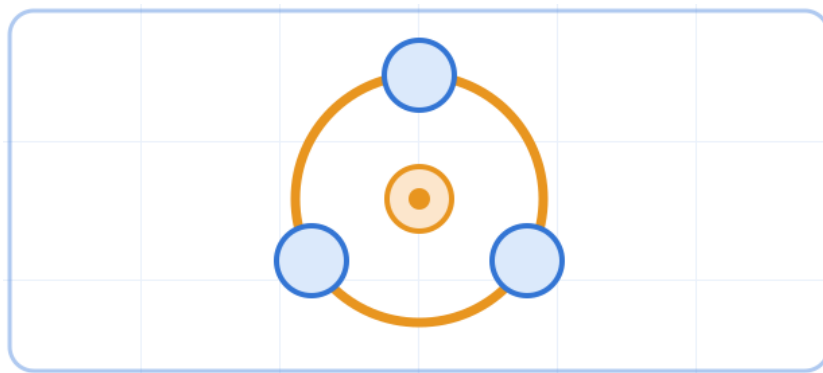
BLOCK 3 · STRUCTURE: EMPOWERED PRODUCT TEAMS, ROLES AND ORGANISATIONAL DESIGN

Chapter 10. Roles and competencies of the model · ESTABLISHED

Product manager, product designer, tech lead/engineering and product leader; competencies over titles, including that of using the tools of product work with sound judgement.

Introduction

A product team brings together competencies rather than job titles. The model names typical roles —product manager, product designer, engineering/tech lead and, above them, product leaders—, but the guiding principle is that what matters is the competencies the team must have, not the titles on the org chart. This chapter deals with the place of roles in the model, not the detailed craft of each one.



Competencies —not titles— are what the team must bring together; leadership articulates them.

10.1 The typical roles and competence over title

The product manager is accountable for value and viability; the designer, for the experience; engineering, for feasibility and delivery. In small teams, one person may cover more than one role, and in large ones, a role may be split. What cannot be missing are the competencies: if the team lacks any of them, it will decide badly no matter how well the titles are covered.

10.2 Using the tools with sound judgement (AI among them)

Among the competencies of product work is that of using the available tools with sound judgement, and AI is today the one that most reorders the weight of each competency: part of the mechanical work becomes cheaper, and judgement gains value —deciding which problem deserves attention, interpreting evidence, validating what the tool produces. AI enters here as context that changes the emphasis of the competencies, not as a new role.

Competence, not the org chart. The right question is not "do we have a PM, a designer and a tech lead?", but "does this team bring together the competencies to discover and deliver value autonomously?".

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 one.
- Explain why the model prioritises competencies over titles.
- Assess whether a team brings together the necessary competencies, beyond the job titles.
- Describe how AI reorders the weight of the competencies without creating a separate role.

Common mistakes and antipatterns

Covering titles, not competencies. Considering a team complete because it has the expected job titles, even though it lacks some key competency.

AI as a role. Treating the use of AI as a separate function instead of as a cross-cutting competency of the team.

Siloed roles. Turning roles into silos that hand work off to each other, breaking the team's cross-functionality.

Further reading

- [SVPG · The Product Operating Model](#)
- [Scrum.org · Agile Product Operating Model](#)

BLOCK 3 · STRUCTURE: EMPOWERED PRODUCT TEAMS, ROLES AND ORGANISATIONAL DESIGN

Chapter 11. Product leadership · ESTABLISHED

Leading product people through coaching and leadership by context, and the role of leaders and of senior management—including the CEO—in sustaining the model.

Introduction

If teams decide the "how", what do leaders do? They provide context and develop people. Product leadership is exercised through coaching and through leadership by context: the leader sets the direction and the constraints, grows their people and removes obstacles, instead of handing out solutions. And senior management, up to the CEO, has an irreplaceable role in sustaining the model.

11.1 Coaching and leadership by context

Leading by context is giving each team the information, objectives and constraints it needs to decide well, and trusting it to decide. Coaching is the other half: developing people's competencies so that that trust is justified. A leader who gives solutions instead of context reintroduces the feature model through the back door, however good their intention.

11.2 The role of senior management

The operating model is a company-wide decision, not a departmental one. That is why the sponsorship of senior management—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 a team does not control. Without that sponsorship, teams are left empowered on paper but trapped by a structure that does not change.

Context, not solutions. The best indicator of the health of product leadership is what it hands to the teams: if it hands problems, outcomes and context, the model breathes; if it hands solutions and plans, it suffocates it.

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 replaces direct control.
- Describe the role of coaching in the development of teams.
- Argue why the sponsorship of senior management is a condition of the model.
- Recognise when a leader reintroduces the feature model by giving solutions.

Common mistakes and antipatterns

Leader who gives solutions. Replacing context with detailed instructions, emptying the team's empowerment of content.

Empowerment without sponsorship. Asking for empowered teams without management changing funding, governance and incentives.

Delegate and vanish. Confusing leading by context with absence, leaving teams without direction or development.

Further reading

- [SVPG · The Product Operating Model](#)
- [Scrum.org · APOM in practice \(Accenture\)](#)

BLOCK 3 · STRUCTURE: EMPOWERED PRODUCT TEAMS, ROLES AND ORGANISATIONAL DESIGN

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

How teams are grouped around products and value streams: team types and interactions, cognitive load, and the balance between autonomy and alignment with the minimisation of dependencies.

Introduction

With products and teams defined, what remains is to decide how they are grouped and how they interact. The organisational design of the model rests on consolidated frameworks —Team Topologies is the common reference— to group teams around value streams, limit their cognitive load and minimise the dependencies that slow the flow. It deals with the structural role of these frameworks, not their detailed teaching.

12.1 Team types and interactions

Team Topologies proposes four team types —stream-aligned, platform, enabling and complicated-subsystem— and three interaction modes —collaboration, X-as-a-service and facilitation. The stream-aligned team is the primary one: it delivers value end to end over a stream. The others exist to put it in the best conditions, reducing its cognitive load. The value of the framework lies in restricting the variety: four types and three interactions are enough.

12.2 Autonomy, alignment and dependencies

The design seeks the balance between autonomy (each team decides and delivers on its own) and alignment (everyone rows towards the same strategy). The main lever is to minimise dependencies: the fewer waits between teams, the more value flows. When the architecture does not allow full decoupling, teams of teams appear, which link to the scaling boundary of chapter 25.

Conway in action. The structure of teams tends to be reflected in the architecture of the product, and vice versa. Designing teams is, in fact, designing the system; ignoring this condemns you to fighting against your own organisation.

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.
- Argue how cognitive load and dependencies condition the flow of value.
- Relate team design to product architecture (Conway's law).
- Identify when teams of teams appear and their link with scaling.

Common mistakes and antipatterns

Hybrids and bespoke types. Inventing your own team types instead of using the four in the framework, which reintroduces the confusion the framework avoids.

Platform that standardises, not serves. Turning the platform into a control body instead of a service that reduces the cognitive load of the stream teams.

Unmanaged dependencies. Tolerating permanent dependencies between teams as if they were inevitable, instead of redesigning to reduce them.

Further reading

- [Team Topologies · Key concepts](#)
- [Scrum.org · APOM State of Play \(I\)](#)

BLOCK 4 · FUNDING AND GOVERNANCE BY VALUE STREAMS

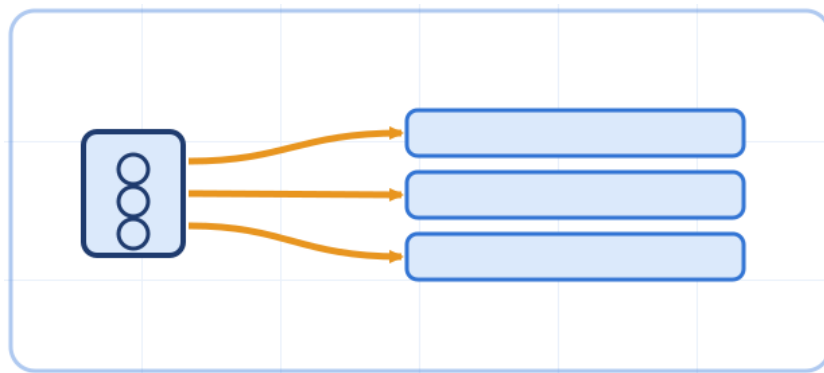
Chapter 13. From project funding to product/value-stream funding

CONSOLIDATING

Why project funding clashes with the model, and the persistent funding of products and value streams as opposed to allocation by initiative.

Introduction

Funding is the dimension where the most transitions run aground. An organisation 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 providing persistent resources to the teams that look after a product, and reviewing that provision by outcomes, not by milestones of a plan.



Funding that flows persistently to the products, not allocated by initiative or project.

13.1 Why project funding clashes with the model

Project funding presupposes that the work has a known scope and an end. It forces up-front estimation of what can only be discovered by doing, penalises learning (any change of course is a budget deviation) and disbands teams when the project "ends", destroying knowledge. All of this contradicts the basic assumption of the product as a continuous effort.

13.2 Funding products and value streams

The alternative is to fund persistent teams and review the investment periodically in light of the evidence of value (chapter 17), treating large initiatives as bets. The cheapening of the marginal cost of building —with AI among the causes— reinforces this logic: if building is ever cheaper, what is scarce and valuable is deciding well what to keep investing in, something that persistent funding with evidence-based review facilitates and project funding hinders.

Why it is consolidating. The principle is widely advocated, but its adoption is still partial: Scrum.org's survey points out that silos and structural impediments persist and limit the real autonomy of teams.

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 continuous product.
- Describe what funding products or value streams persistently consists of.
- Relate the review of investment to the evidence of value instead of to milestones of the plan.
- Argue why the cheapening of output reinforces persistent funding.
- Recognise when the financial structure drags the organisation back to the project model.

Common mistakes and antipatterns

Product funded as a project. Providing "products" with scope-and-date budgets, reintroducing the project logic by way of the money.

Reviewing by milestones, not by evidence. Releasing or cutting funding according to progress against the plan instead of according to demonstrated value.

Reorganising without refunding. Changing the team structure while leaving the budget-allocation model intact.

Further reading

- [Scrum.org · Barriers to value delivery \(APOM\)](#)
- [SVPG · The Product Operating Model](#)

BLOCK 4 · FUNDING AND GOVERNANCE BY VALUE STREAMS

Chapter 14. Lean budgeting and Beyond Budgeting · CONSOLIDATING

Incremental allocation, short horizons and funding of persistent teams; the principles of Beyond Budgeting and of lean-agile budgeting at the level of concept, not the mechanics of a framework.

Introduction

If persistent products are funded, a way of budgeting that is coherent with uncertainty and with the continuous cadence is needed. The principles of lean budgeting and of Beyond Budgeting offer that conceptual framework: allocating incrementally, with short, reviewable horizons, instead of a closed, negotiated annual budget. Here the principles are dealt with, not the mechanics of any specific framework.

14.1 The principles of Beyond Budgeting

Beyond Budgeting proposes separating what the annual budget mixes together —objectives, forecasts and resource allocation— and managing it adaptively: relative and ambitious objectives instead of fixed figures, continuous planning instead of an annual event, and resources available as needed instead of handed out in advance. Its principles are grouped into leadership (values, transparency, autonomy, accountable teams) and management processes (targets, rewards, planning, coordination, resources, controls).

14.2 Lean-agile budgeting applied to product

Brought to the product model, this means providing resources to persistent teams, allocating funds over short horizons that are reviewed against the evidence, and preventing the budget from becoming a disguised scope commitment. It does not require abolishing all budgeting overnight: it requires that allocation stop being based on closed annual plans and start being guided by outcomes and learning.

Adoption, not novelty. These principles are mature —Beyond Budgeting is decades old—, but their adoption in product organisations remains uneven. That is why this entry is consolidating: for reasons of adoption, not of conceptual novelty.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain what three things the annual budget mixes together and why it is worth separating them.
- Describe the principles of Beyond Budgeting in its two groups: leadership and management processes.
- Translate those principles into the budgeting of persistent product teams.
- Distinguish allocating incrementally and reviewably from budgeting by annual scope.
- Argue why its state is one of uneven adoption and not of conceptual immaturity.

Common mistakes and antipatterns

Disguised annual budget. Calling "product funding" a closed annual budget under another name.

Fixed target as goal and as reward. Tying rewards to fixed negotiated figures, which incentivises caution and bargaining instead of value.

All or nothing. Postponing any change because "you cannot abolish the budget", instead of adopting the principles incrementally.

Further reading

- [Beyond Budgeting · The 12 principles](#)
- [Scrum.org · APOM, evidence-based approach](#)

BLOCK 4 · FUNDING AND GOVERNANCE BY VALUE STREAMS

Chapter 15. Governance, product authority and constraints · CONSOLIDATING

Who decides what once the strategy is set: the distribution of product authority, the role of management and of the CFO, light governance, and distinguishing fundamental constraints from those inherited without justification.

Introduction

Empowering teams immediately raises the question of governance: who decides what, and where are the limits? The model distributes product authority so that, once the strategy is set, teams decide within clear constraints, with light governance that verifies outcomes instead of approving every step. It is the dimension where the transition is hardest, because it touches power.

15.1 Distribution of authority and light governance

Authority over product decisions comes down to the teams once the strategy and the constraints are clear. The role of management and of the CFO changes: from approving plans and controlling by phases to setting the context, ensuring the legitimate constraints and reviewing outcomes. Light governance does not mean the absence of governance, but verification by evidence instead of control by approvals.

15.2 Fundamental versus inherited constraints

Not all constraints are equal. Fundamental ones —compliance, security, regulation, real risks— are legitimate and must be respected. Inherited ones are those that only exist "because it has always been done this way": approval processes, templates and controls that no longer protect against anything. APOM insists on distinguishing one from the other: confusing an inherited constraint with a fundamental one is one of the main sources of friction and of false impossibility.

Power is the hardest part. Distributing authority changes who decides, and that is why this transition runs into power dynamics. Distinguishing the fundamental constraint from the inherited one is the tool for unlocking autonomy without taking on real risks.

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 is set.
- Describe the change of role of management and of the CFO towards light governance.
- Distinguish light governance from the absence of governance.
- Differentiate a fundamental constraint from one inherited without justification.
- Identify where power dynamics slow the distribution of authority.

Common mistakes and antipatterns

Governance by approvals. Keeping committees and approval gates that control every step instead of verifying outcomes.

Inherited constraint as law. Treating inherited processes and controls as fundamental constraints, blocking autonomy without reducing any real risk.

Empowering without ceding authority. Announcing empowerment while decision authority remains concentrated at the top.

Further reading

- [Scrum.org · APOM State of Play \(I\)](#)
- [SVPG · The Product Operating Model](#)

BLOCK 5 · OUTCOME ORIENTATION: METRICS AND EVIDENCE-BASED GOVERNANCE

Chapter 16. Objectives by outcomes: OKRs · ESTABLISHED

The OKR as a mechanism of alignment and outcome governance at the portfolio and team levels: focus and context, not a task list.

Introduction

OKRs (objectives and key results) are the most widespread mechanism for aligning the organisation by outcomes. An objective expresses a qualitative direction; the key results make it measurable. In the product model, OKRs connect the portfolio with the teams: they translate strategy into outcomes to achieve and give each team a measure of success without dictating its solution. This chapter deals with their function in the model, not the technique of writing them, which is craft.

16.1 The OKR as focus and context

Well used, an OKR is a tool of focus: it forces a choice of what really matters over a short horizon and a renouncing of the rest. It is also context: it communicates to the teams the outcome sought and leaves them the how. Badly used, it degenerates into a task list with the appearance of objectives, or into an instrument of top-down control that reintroduces micromanagement.

16.2 Portfolio and team OKRs

At the portfolio level, OKRs guide the bets and the work across products (chapter 7). At the team level, they give the measure of success of the assigned problem. The key is vertical coherence without imposition: team OKRs should contribute to the portfolio ones, but be formulated with the team's participation, not handed down to it already solved. OKRs are not the same as KPIs: they change over time and express the change being sought, not the operation being monitored.

Outcomes, not tasks. If a team's key results are really deliverables ("launch feature X"), the OKR has gone back to measuring output. A good key result describes a change in the customer or the business.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the function of the OKR as a mechanism of focus and of context within the model.
- Distinguish an outcome-oriented key result from a disguised deliverable.
- Describe how portfolio and team OKRs relate without imposition.
- Differentiate OKRs from KPIs.
- Recognise when OKRs degenerate into a task list or into top-down control.

Common mistakes and antipatterns

OKR as a task list. Turning key results into committed deliverables, losing the outcome orientation.

Top-down OKRs. Imposing already-solved OKRs on teams, using them as an instrument of control instead of context.

Too many objectives. Diluting focus with a long list of OKRs, which amounts to having no focus.

Further reading

- [Scrum.org · APOM, evidence-based approach](#)
- [Scrum.org · Agile Product Operating Model](#)

BLOCK 5 · OUTCOME ORIENTATION: METRICS AND EVIDENCE-BASED GOVERNANCE

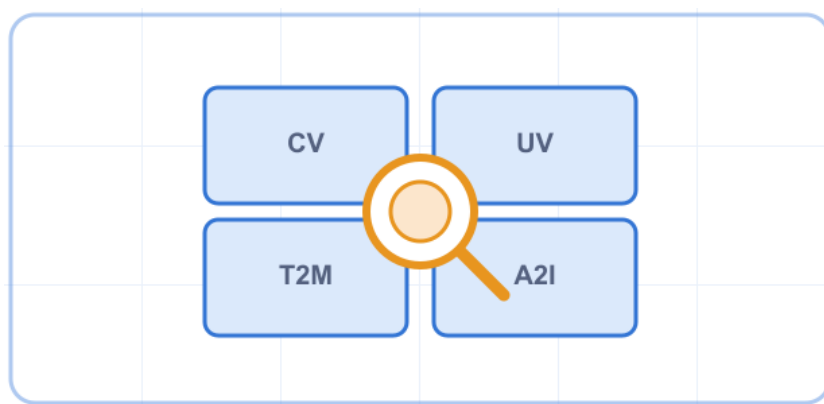
Chapter 17. Evidence-Based Management (EBM) and the four Key Value Areas

· ESTABLISHED

The four Key Value Areas —Current Value, Unrealized Value, Time-to-Market and Ability to Innovate— and empiricism at the management level: hypotheses and experiments instead of activity.

Introduction

Evidence-Based Management (EBM) is Scrum.org's framework for governing by value instead of by activity. It proposes measuring what matters in four Key Value Areas and treating management decisions as hypotheses to be tested against evidence. It is the evidence anchor of the APOM formulation and a settled standard.



The four Key Value Areas of EBM, inspected empirically to govern by value.

17.1 The four Key Value Areas

EBM organises measurement into four areas. Current Value measures the value the product delivers today to customers and to the organisation. Unrealized Value estimates the potential value not yet captured in the market. Time-to-Market measures the ability to deliver quickly and frequently. And Ability to Innovate measures how much of the organisation's capacity is available to create new value and is not consumed by friction. The first two describe market value; the last two, organisational capability.

17.2 Empiricism at the management level

Activity and output —how many features, how many lines, how many points— do not, on their own, tell you whether value is being delivered or whether the organisation is able to keep delivering it. EBM shifts the focus towards those four areas and treats decisions as experiments: formulate a value hypothesis, act, measure the effect on the areas and adapt. It is the same empiricism of product work, brought to the management and portfolio level.

Four lenses, not a single number. None of the four areas is sufficient on its own: a good Current Value with a poor Time-to-Market or scant Ability to Innovate describes an organisation living off its rents. Evidence-based governance looks at the four together.

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 organisational capability.
- Explain why activity metrics do not tell you about the delivery of value.
- Describe the empirical management cycle: hypothesis, action, measurement, adaptation.
- Argue why the four areas must be looked at together.

Common mistakes and antipatterns

Measuring activity. Governing with output metrics (velocity, number of releases) as if they told you about the value delivered.

A single area. Optimising Current Value while ignoring Time-to-Market or Ability to Innovate, and living off the rents.

Metrics with no hypothesis. Collecting indicators without a value hypothesis that gives meaning to their reading and to the decision.

Further reading

- [Scrum.org · Evidence-Based Management Guide](#)
- [Scrum.org · Evidence-Based Management](#)

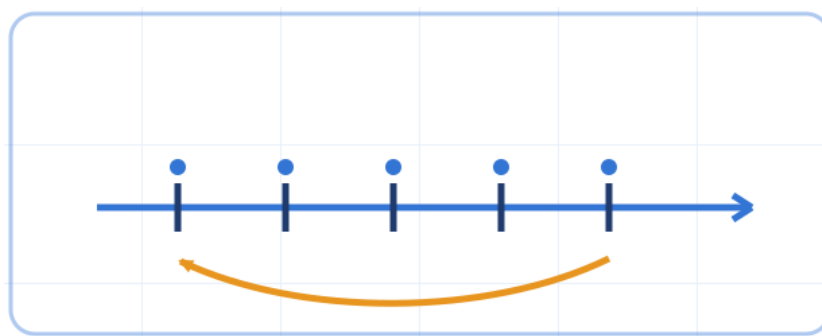
BLOCK 5 · OUTCOME ORIENTATION: METRICS AND EVIDENCE-BASED GOVERNANCE

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

Small, frequent and reliable releases —at least every two weeks, ideally continuous— with instrumentation, observability and phased deployment. It is captured as a condition of the model, not as an engineering technique.

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: if delivering takes months, the loop of hypothesis and evidence breaks. That is why delivery cadence is captured here as a condition of the model, not as just another engineering technique.



Small, frequent releases that close a learning loop about real value.

18.1 Small, frequent and reliable

The reference the sources give is to deliver at least every two weeks, and ideally continuously. What matters is not the exact number, but the principle: small batches that reduce the risk of each change, frequency that shortens the learning loop, and reliability that makes delivering not a traumatic event. An organisation that can only deliver twice a year cannot operate by product, no matter how it reorganises its teams.

18.2 Instrumentation, observability and phased deployment

Reliable cadence rests on technical conditions: instrumenting the product to know what is happening, observability to detect problems early, and phased deployment to contain the impact of each change. The guide captures these conditions as part of the model —without them the cadence is not reliable—, but their detailed technique (CI/CD, testing, observability) belongs to engineering and falls outside the scope.

A condition, not a frill. Delivery cadence is not a luxury of mature teams: it is the requirement that makes empiricism, evidence-based governance and reviewable funding possible. Where it fails, the rest of the model stays on paper.

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 not an accessory technique.
- Justify the value of small batches, frequency and reliability.
- Relate cadence to the empirical learning loop and evidence-based governance.
- Name the technical conditions that sustain a reliable cadence, placing their technique outside the scope.

Common mistakes and antipatterns

Reorganising without speeding up delivery. Creating product teams that keep delivering at a low frequency, breaking the learning loop.

Delivery as an event. Treating each deployment as a high-risk occurrence, which discourages delivering often.

Cadence without observability. Delivering fast without instrumentation or observability, turning speed into unmeasured risk.

Further reading

- [SVPG · The Product Operating Model: An Introduction](#)
- [SVPG · The Product Operating Model](#)

BLOCK 5 · OUTCOME ORIENTATION: METRICS AND EVIDENCE-BASED GOVERNANCE

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

Metrics at the portfolio level and of the transition itself, and the signals that the organisation genuinely operates by product and not only in name.

Introduction

Beyond measuring the value of each product, the model needs to know whether the organisation itself is genuinely 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 standardised practice than EBM, and that is why it is consolidating: there is agreement that it is needed, but no settled standard for how to do it.

19.1 Portfolio metrics

At the portfolio level, what is of interest is how investment is distributed between products, what outcomes each bet is producing and whether the set is advancing towards the strategic objectives. The four EBM areas (chapter 17) offer an aggregable starting point, but portfolio measurement adds its own questions: balance between exploiting the existing and exploring the new, dependencies between products, and the real speed of reallocating resources in light of evidence.

19.2 Signals of "really" operating by product

The most useful signal is not a number, but a set of indications: are teams assigned problems or solutions?, are they funded persistently or by project?, is investment reviewed by evidence or by milestones?, do teams decide or do committees approve? An organisation can have all the vocabulary of product and operate by project in practice; measuring the health of the model is detecting that gap between the rhetoric and the practice.

Rhetoric versus practice. The characteristic risk of transitions is cosmetic change: new names, old behaviours. Measuring the health of the model is, above all, making that gap visible so that it can be closed.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why the health of the model must be measured in addition to the value of each product.
- Describe what own questions portfolio measurement adds to the EBM areas.
- Identify signals that an organisation operates by product as opposed to cosmetic change.
- Argue why this practice is less standardised and consolidating.

Common mistakes and antipatterns

Cosmetic change. Taking the transition for granted because the vocabulary has been adopted, without measuring whether behaviours changed.

Only product metrics. Measuring the value of each product but not the health of the model, without seeing whether the organisation operates by product.

Health metric as a ranking. Turning the health signals into a punitive scorecard, which incentivises window-dressing instead of improvement.

Further reading

- [Scrum.org · Barriers to value delivery \(APOM\)](#)
- [Scrum.org · Evidence-Based Management](#)

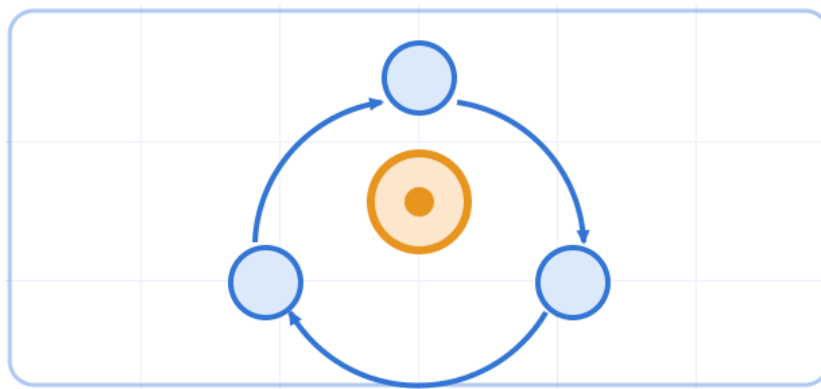
BLOCK 6 · CULTURE OF PRODUCT WORK

Chapter 20. Product mindset, trust and psychological safety · ESTABLISHED

The culture oriented towards product and outcomes as opposed to the one oriented towards projects and dates, and the conditions that sustain it: starting from the "why", transparency and trust as the basis of autonomy.

Introduction

Culture is the dimension that makes the rest of the model take root or stay on paper. A product mindset values outcomes over meeting dates, learning over certainty and ownership over carrying out orders. And that mindset needs a substrate: trust and psychological safety, without which empowerment is nominal.



An empirical culture sustained by trust: continuous hypothesis, experiment and learning.

20.1 Starting from the why and transparency

A product culture starts by explaining the why: teams that understand the reason for what they do decide better and take ownership of the outcome. Transparency —sharing the information, the data and the logic of the decisions— is what enables that why and what turns context into a substitute for control. An opaque organisation cannot empower, because empowering requires sharing what is needed to decide.

20.2 Trust and psychological safety

Autonomy rests on trust: trusting that the team will decide well with the right context, and the team trusting that it can say what it thinks, raise doubts and admit mistakes without reprisals. That psychological safety is what enables learning (chapter 21): where admitting a mistake is punished, mistakes are hidden and the organisation stops learning from them.

Culture as the foundation. You can copy the structure, funding and metrics of a product organisation and still fail, if the culture that sustains them is missing. Culture is not the model's ornament: it is its foundation.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Characterise a product mindset as opposed to a project mindset.
- Explain why starting from the why improves decision-making and ownership.
- Relate transparency to the possibility of empowering.
- Argue why psychological safety is a condition of learning.
- Recognise when an opaque or punitive culture empties empowerment of content.

Common mistakes and antipatterns

Empowering without transparency. Asking teams to decide while withholding the information and the logic they need.

Punishing mistakes. Penalising honest failures, which teaches people to hide them and switches off learning.

Culture of dates. Rewarding adherence to the calendar over the outcome, reinstalling the project mindset.

Further reading

- [SVPG · The Product Operating Model](#)
- [Scrum.org · APOM State of Play \(I\)](#)

BLOCK 6 · CULTURE OF PRODUCT WORK

Chapter 21. Learning, experimentation and error management · ESTABLISHED

An empirical culture: discovery as the norm, tolerance of controlled failure and continuous improvement. Cheap, early error is information, not a defect to hide.

Introduction

The product model assumes that it is not known in advance what will work: that is why it discovers before committing. The culture that sustains it treats learning as the norm, not the exception, and early, cheap error as valuable information. It is the cultural counterpart of delivery cadence (chapter 18) and of evidence-based governance (chapter 17).

21.1 Discovery as the norm

To discover is to validate ideas with controlled risk before investing in building them at scale: talking to users, testing prototypes, running small experiments. In a product culture, this is not a preliminary phase that is skipped when in a hurry, but a habitual way of working. The alternative —building blindly what someone assumed— is precisely what the cheapening of output makes more dangerous.

21.2 Error management and continuous improvement

Tolerating controlled failure is not celebrating failure, but designing so that errors are small, early and reversible, and so that learning is extracted from them. Continuous improvement closes the cycle: each experiment, whether it succeeds or fails, leaves the organisation knowing something it did not know before. This only works on the psychological safety of the previous chapter: without it, error is hidden and the learning is lost.

Cheap error is information. The question is not whether there will be errors —there will be—, but whether they will be small, early and reversible, and whether the organisation 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 working norm and not a dispensable phase.
- Distinguish tolerating controlled failure from celebrating failure.
- Design so that errors 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.

Heroic failure. Confusing tolerance of error with taking large, irreversible risks without learning.

Post-mortem that looks for culprits. Turning the analysis of an error into an apportioning of blame, which guarantees that the next error is hidden.

Further reading

- [SVPG · The Product Operating Model: An Introduction](#)
- [Scrum.org · Evidence-Based Management](#)

BLOCK 6 · CULTURE OF PRODUCT WORK

Chapter 22. The relationship with stakeholders and with "the business"

· ESTABLISHED

Moving beyond the "business that requests / technology that serves" model: the collaboration between product and the rest of the organisation and the handling of cultural objections.

Introduction

The product model breaks the customer-supplier relationship between "the business" and "technology". In the IT model, the business requests and technology serves; in the product model, technology and business solve problems together, jointly accountable for the outcome. Changing that relationship is one of the deepest cultural changes and one that arouses the most resistance.

22.1 Beyond "business that requests, technology that serves"

The critique of the "IT model" is central to Cagan's formulation: when technology only executes requests, no one owns the outcome and what was asked for gets built even if it does not solve the problem. Real collaboration means that product teams take part in understanding the problem and in deciding the solution, and that stakeholders provide context and constraints instead of closed specifications.

22.2 Handling cultural objections

This change clashes with entrenched expectations: stakeholders used to commissioning features, areas that measure their power by the size of their backlog, executives who expect scope-and-date commitments. Handling these objections —listening to them, distinguishing the legitimate concern from resistance to the shift of power, and showing results— is part of the cultural work; it links to the catalogue of transition objections (chapter 24).

Shared accountability. The indicator that the IT model has been overcome is simple: when something does not work, is the question "who requested it" or "what did we learn"? The former is customer-supplier; the latter, product's shared accountability.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain what the "IT model" consists of and why it prevents ownership of the outcome.
- Describe the collaborative relationship between product and business in the product model.
- Distinguish providing context and constraints from delivering closed specifications.
- Identify typical cultural objections and separate the legitimate concern from resistance.

Common mistakes and antipatterns

Business that dictates solutions. Keeping stakeholders who commission specific features instead of posing problems and outcomes.

Product that only executes. Accepting the role of internal supplier and building what was requested without shared accountability for the outcome.

Backlog as power. Measuring an area's influence by the size of its request queue, incentivising asking for more instead of achieving more.

Further reading

- [SVPG · The Product Operating Model](#)
- [SVPG · The Product Operating Model: An Introduction](#)

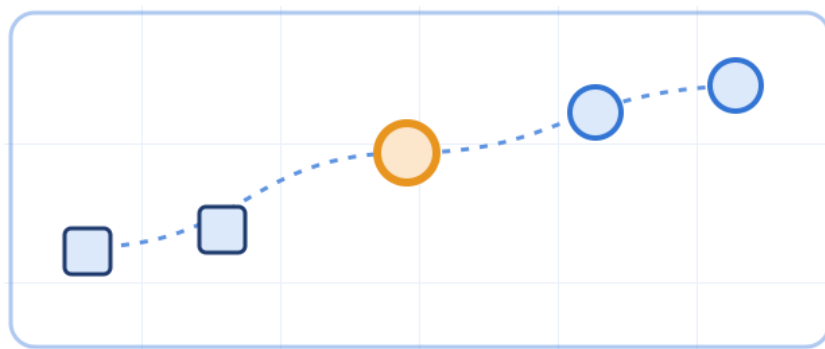
BLOCK 7 · THE TRANSITION TO THE PRODUCT OPERATING MODEL

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

Why "renaming to product" without changing structure, funding and incentives fails, and why the change is a continuous journey, product by product, where each group advances at its own pace.

Introduction

The transition to a product model is holistic and incremental. Holistic because the five dimensions — structure, funding, governance, metrics, culture— must move together; changing only one produces friction. Incremental because it is not done all at once: it advances product by product, at different paces, as a continuous journey and not as a final state that is reached and considered finished.



An incremental transition, product by product, where each group advances at its own pace.

23.1 Why renaming without changing fails

The most documented failure of transitions consists of renaming the existing as "product" without touching the structure, the funding or the incentives. Teams are called product teams but are funded by project, committees decide and plan delivery is rewarded. The label changes; the behaviour does not. Hence the holistic nature: the model only works if the five dimensions change coherently.

23.2 The incremental journey

The alternative is to sequence: start with some products, learn, and extend. Each product group can advance at a different pace according to its maturity and its context, and the transformation is understood as a journey that continues, not as a project with an end date. Treating the transition as a project —with its scope, its plan and its end— would, ironically, be repeating the model you want to abandon.

Holistic and incremental at once. Holistic does not mean "everything at once across the whole company": it means that, wherever you advance, the five dimensions advance together. You progress by areas, but in each area completely.

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 if only one dimension changes.
- Describe why renaming projects as products without changing the substance fails.
- Argue the incremental and continuous nature of the transformation.
- Distinguish advancing by complete areas from advancing by isolated dimensions across the whole company.
- Recognise when the transition is being managed, wrongly, as a project with an end.

Common mistakes and antipatterns

Rename and declare 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, plan and end date, repeating the model you want to overcome.

Big bang. Trying to change the whole organisation at once instead of advancing by complete areas and learning.

Further reading

- [Scrum.org · APOM State of Play \(I\)](#)
- [SVPG · The Product Operating Model](#)

BLOCK 7 · THE TRANSITION TO THE PRODUCT OPERATING MODEL

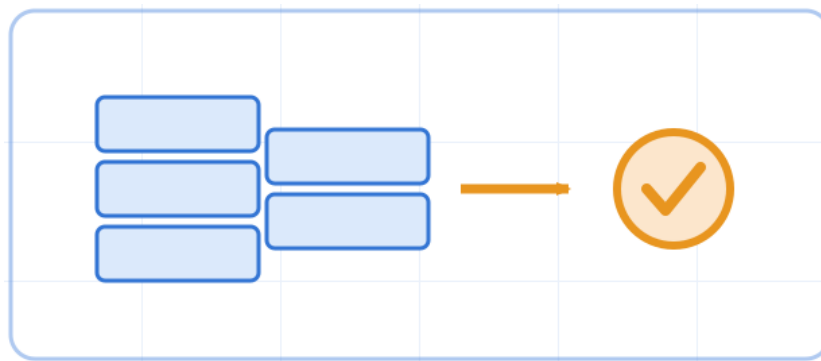
Chapter 24. Leading the change, objections and power dynamics ·

ESTABLISHED

The role of the CEO and of senior management, transformation teams, the personal transformation of leaders, and the catalogue of stakeholder objections and their responses.

Introduction

No transition to product prospers without leadership of the change. The dimensions a team does not control —funding, governance, incentives— are moved only by senior management, and the distribution of authority the model requires touches power dynamics that generate resistance. This chapter deals with leadership of the change in what is specific to this model, not with change management as a general discipline.



Each typical stakeholder objection has a prepared response to match it.

24.1 Sponsorship and personal transformation of leaders

The sponsorship of the CEO and of senior management is a necessary condition: they are the ones who authorise the change in the structural dimensions. But authorising is not enough: often the leaders themselves must transform their way of leading, moving from control by approvals to leadership by context (chapter 11). Transformation teams —cross-functional and with a real mandate— help sustain the journey, provided they do not become a new silo that "does the transformation" to everyone else.

24.2 Objections and power dynamics

Transformed devotes an entire section to stakeholder objections and their responses, because anticipating them is part of the work. The typical objections —"we will lose control", "you cannot plan like this", "this does not apply to our sector"— mix legitimate concerns with resistance to the shift of power. A delicate case is divesting from a product that is valuable today but whose owner is influential: there it is the power dynamic, not the analysis, that blocks. Recognising that political dimension is part of leading the change.

Anticipate the objection. Objections are not unforeseen obstacles: they are predictable and recur across organisations. Having an honest response ready for each one —one that distinguishes the real concern from the resistance— is part of the work of leadership.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why the sponsorship of senior management is a condition of the transition.
- Describe the personal transformation the model demands of the leaders themselves.
- Locate the role of transformation teams and their risk of becoming a silo.
- Anticipate typical stakeholder objections and distinguish the legitimate concern from resistance.
- Recognise the power dynamics that block decisions beyond the analysis.

Common mistakes and antipatterns

Transformation without management. Driving the change from below 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 recognising the power dynamics that sustain them.

Further reading

- [SVPG · The Product Operating Model](#)
- [Scrum.org · APOM in practice \(Accenture\)](#)

BLOCK 7 · THE TRANSITION TO THE PRODUCT OPERATING MODEL

Chapter 25. The boundary with scaling frameworks · CONSOLIDATING

How SAFe, LeSS and Nexus relate to the operating model —as a layer of instantiation or implementation— without teaching them, and what distinguishes them from the model itself.

Introduction

A recurring question is how the product operating model fits with the agile scaling frameworks (SAFe, LeSS, Nexus). The answer that is settling is that they operate at different levels: the operating model states the "what" and the "why" —how the company should be organised to operate by product—, while a scaling framework is one possible form of the "how", a layer of instantiation. This chapter situates that boundary; it does not teach the frameworks.

25.1 Scaling as a layer of instantiation

A scaling framework can be a way of implementing a product operating model in a large organisation: it helps organise teams of teams, coordinate dependencies and provide cadence when the architecture does not allow full decoupling. Scrum.org has raised this explicitly for SAFe, presenting it as a route to instantiate APOM, with the caveat that a badly used framework can reproduce the "feature factory" at scale instead of overcoming it.

25.2 What distinguishes the model from the framework

The key distinction: a scaling framework is not, in itself, a product operating model. Adopting SAFe does not turn an organisation into a product organisation if it still funds by project, measures output and decides by committee. The operating model is the criterion; the framework, a tool that can serve it or betray it. That is why this boundary is consolidating: the "scaling as instantiation" framing is recent and is being actively discussed.

The framework serves the model, not the other way round. The right question is not "which scaling framework do we adopt?", but "which operating model do we want, and which instantiation — framework included— best serves it?". Inverting that order is how a framework ends up reproducing what it was meant to overcome.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the relationship between the operating model and the scaling frameworks in terms of different levels.
- Describe scaling as a possible layer of instantiation of the model.
- Argue why adopting a scaling framework is not the same as operating by product.
- Recognise when a scaling framework reproduces the "feature factory" at scale.
- Situate the consolidating state of this framing by its recency.

Common mistakes and antipatterns

Framework equals model. Believing that adopting SAFe, LeSS or Nexus turns the organisation into a product organisation.

Framework that scales the factory. Using scaling to coordinate the delivery of features at large scale without changing funding, metrics or governance.

Choosing the framework before the model. Deciding the scaling framework before defining which operating model you want, inverting the correct order.

Further reading

- [Scrum.org · APOM and SAFe, do they relate?](#)
- [Scrum.org · 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.