Written by

Ilia Pavlichenko

Rethinking Product Groups

Integration Around Product Families

January **2026**

Executive Summary

The article reframes the product group as an **integration container around a product family**, whose shape and autonomy depends on the architecture of the business portfolio and the degree of relatedness across the company's businesses.

To choose an appropriate product group design, the article uses portfolio relatedness patterns from Kates & Kesler / Galbraith: **single integrated business**, a **closely related portfolio**, and a **loosely related portfolio**, showing how in each case the container's form shifts from a logical layer to a formally semi-autonomous unit. In a single integrated business (e.g., Toyota), product groups often exist as a logical layer: integration is achieved via Chief Engineer / product leaders, shared processes, goals, and culture, while a single functional line of reporting is preserved.

In a closely related portfolio (Big Tech platforms, large banks), product groups receive some dedicated functions (sales, parts of marketing, P&L focus), while critical Product & Tech remain centralized to maintain a common platform, engineering culture, and standards. This creates a balance where domain leaders have enough responsibility to grow their domains but are constrained by corporate architectural decisions and standards.

In a loosely related portfolio, product groups become semi-independent business units with their own product, engineering, sales, marketing, operations, and risk functions, as illustrated by the MTS Kassa case. This structure minimizes dependence on corporate platforms and is especially convenient for transformation, because the head of the product group controls key resources and can change the tech landscape and processes.

The article outlines three types of product groups by degree of formalization and autonomy: a logical (virtual) group, a group with partially dedicated functions, and a semi-independent group. These types form a continuum between efficiency and synergy from shared functions on one side and speed and differentiation at the product-group level on the other; choosing among them is not an ideological debate about centralization but a pragmatic response to portfolio relatedness and company strategy.

Rethinking Product Groups

In the book *Creating Agile Organizations*, a product group was defined as a structural unit built around a product family (or product line), led by a senior manager who is fully accountable for its success and owns all necessary resources and decision rights. That idea was rooted in Toyota's experience and Jeffrey Liker's work *The Toyota Way*.
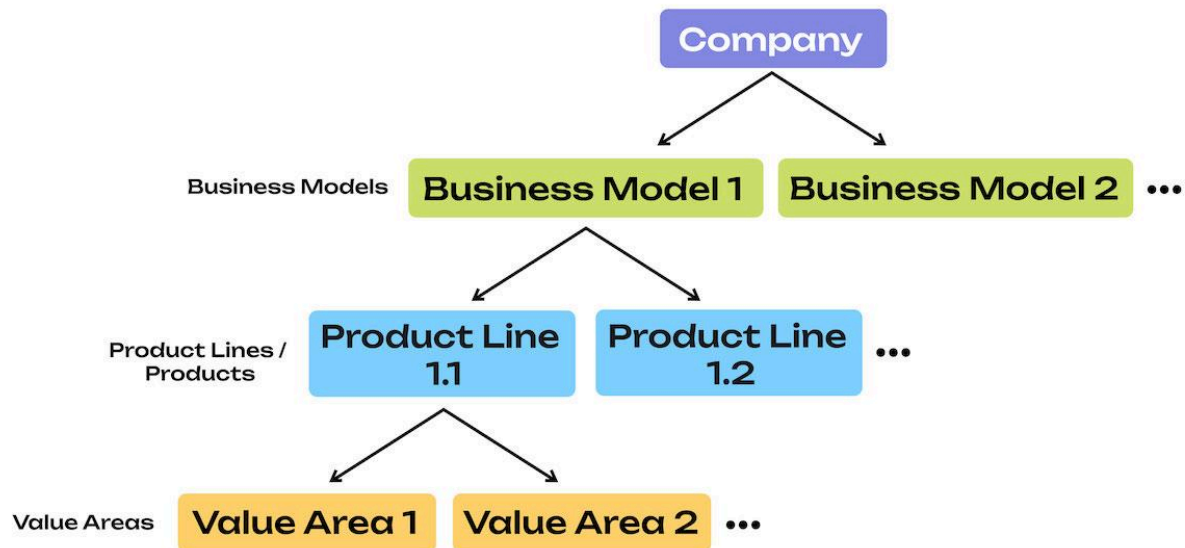
Over the years, this understanding has been refined rather than replaced. Experience across different companies has shown that a product group can be implemented in multiple ways. Today, a product group is viewed as a **container that fosters collaboration by reducing the cost of coordination and integration**, but it is clearer that, depending on the context, this container can take different formal shapes in the organizational structure – from a program-like logical overlay to a formal unit with varying degrees of autonomy, from highly dependent to highly self-contained.

This article presents this updated perspective on what a product group is and how these different types can be used in practice.

Business architecture: business, product families and value areas

At the top level, every company manages a **portfolio of businesses** or **business models**. In a small company this "portfolio" may collapse into a single business model and a single product, but the logic of the architecture remains the same.

Toyota is a useful example. Beyond its well-known automotive brands (Toyota, Lexus, Daihatsu), the company also owns financial services, real estate, trading and logistics, as well as textile machinery and robotics. All these diverse businesses sit under the common umbrella of Toyota.

Typical business architecture

Within each business there can be **product families** and/or **individual products**. Product families are groups of products that share common functionality and a unified technology base. The rationale is efficiency and economies of scale: it is more efficient to develop a family on a common platform than to develop each product from scratch. Nevertheless, companies may deliberately step away from this concept, accepting higher costs and duplication as the price for exploring a broader solution space and increasing the odds of innovative breakthroughs.

Note: in very large companies such as Toyota or P&G, additional layers – categories and brands – may sit between the business and the product lines / products. For simplicity, these layers are omitted here.

The bottom level is Value Areas. A Value Area is a valuable product part that addresses the needs of a customer segment but has no useful value or identity apart from its inclusion in the product. This pattern helps scale a product by letting teams specialize on coherent slices of customer value while still contributing to a single integrated product.

Product Group as an integration container

Any organizational design balances two forces: Separation (functional specialization) and Integration (coordination and synergy). Classic management theorists such as Lawrence and Lorsch, as well as Chandler, identified this dilemma as central.

*A Product Group is an **integration mechanism** – a container that brings together org-functions, roles and units around a product family / product and value stream, encouraging collaboration and reducing coordination costs.*

How this product group is configured depends to a large extent on how the elements of the business portfolio are connected to each other.

Portfolio connectedness: operating models

Amy Kates and Greg Kesler, in *Networked, Scaled, and Agile*, building on Jay Galbraith's work, describe four archetypes of business portfolio connectedness:
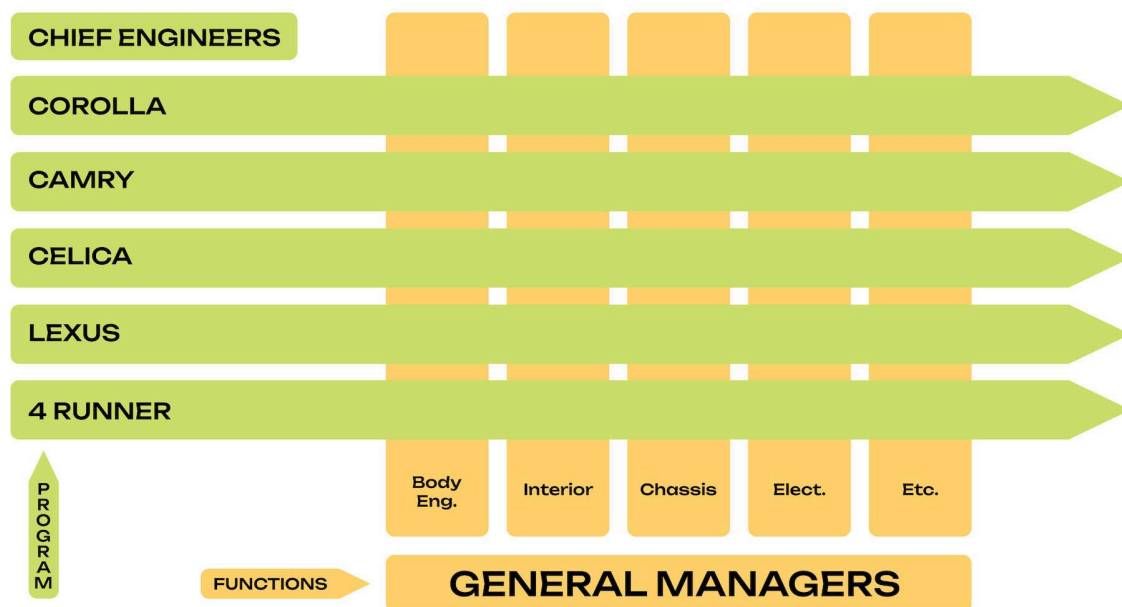- single integrated business,
- closely related portfolio,
- loosely related portfolio,
- and holding / conglomerate.

In this article, the focus is on the first three archetypes, because the fourth one – holding / conglomerate – effectively works as a recursive case where each independent business can itself be analyzed using one of the first three models and corresponding product-group designs.

1. Single integrated business

A single strategy steers all businesses. The corporate center controls strategy and execution; processes are unified and there is one corporate culture. Functional costs are managed centrally. Examples include Apple, Coca-Cola, Heineken. In this model, product groups often exist as a **logical layer** rather than as separate structural units: all product lines are aligned to a single strategy, so coordination happens through shared goals, processes and leadership instead of creating additional standalone business entities.

CreatingAgileOrganizations.com

Product centers at Toyota. Toyota works in a similar fashion: the Chief Engineer plays a coordinating role. At Toyota, the Chief Engineer is the senior integrator for a vehicle program: a program-level leader with end-to-end responsibility for the success of a product line, from concept to production and market performance, who leads a small dedicated team while most engineers remain in functional departments.

Simplified example of Product Centers at Toyota

Development is organized in a matrix-like model, where the product line is managed in product centers. Each center then uses program-like structures (Corolla, Camry, Celica) that act as logical containers, orchestrated through processes, goal alignment and incentives rather than standalone divisional units. The development center managers run the organization, while the Chief Engineer "wins with the product" by owning the vision, priorities and overall success of the vehicle line.
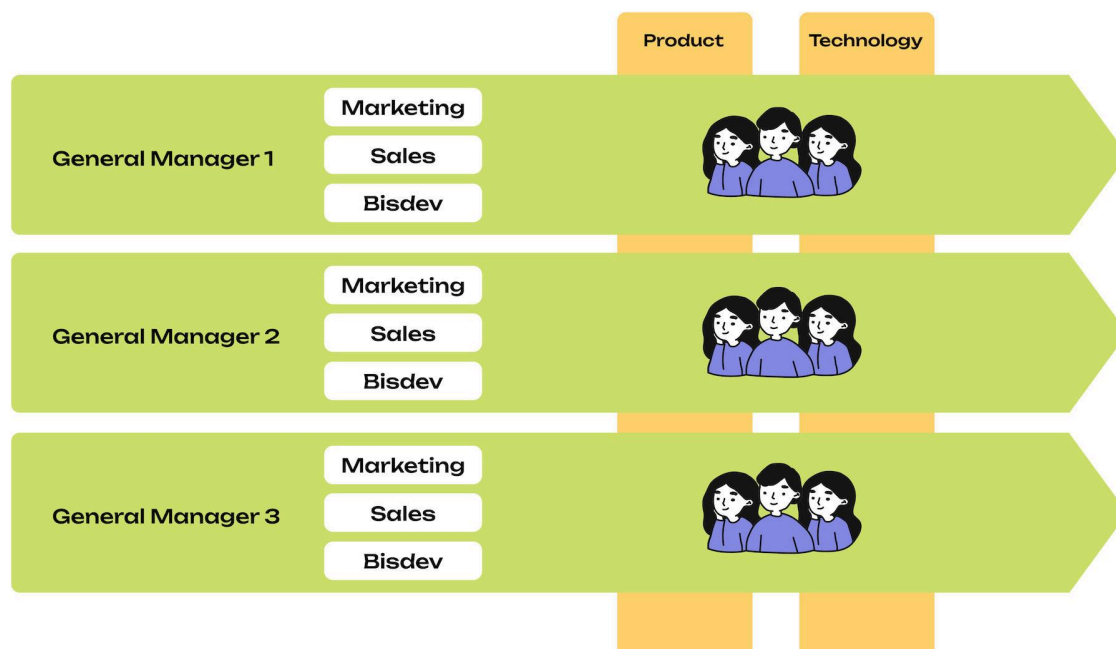
Although it may look like a classic matrix with all the usual problems, Toyota largely avoids those failure modes. The reason is cultural and structural: "Customer First" is treated as a non-negotiable principle, and both the Chief Engineer and the general managers internalize it from day one. In addition, the Chief Engineer does not have

formal line authority over engineers, so reporting remains single-line. What appears as a matrix is, in practice, a clear accountability model with strong alignment rather than dual command.

2. Closely related portfolio

This archetype is characterized by related businesses with meaningful synergies. Business units pursue similar strategies and therefore have a lot of shared resources. The center sets the shared strategic agenda and key processes, supports execution, and coordinates talent and shared services. Examples include *Siemens*, *Microsoft*, *P&G*.

BigTech example. A large multi-category platform is a classic example of a tightly linked portfolio. The platform is structured around several businesses ("Mobility", "Housing", "Jobs", "Goods"), each accountable for its market domain and for driving related product and commercial initiatives.
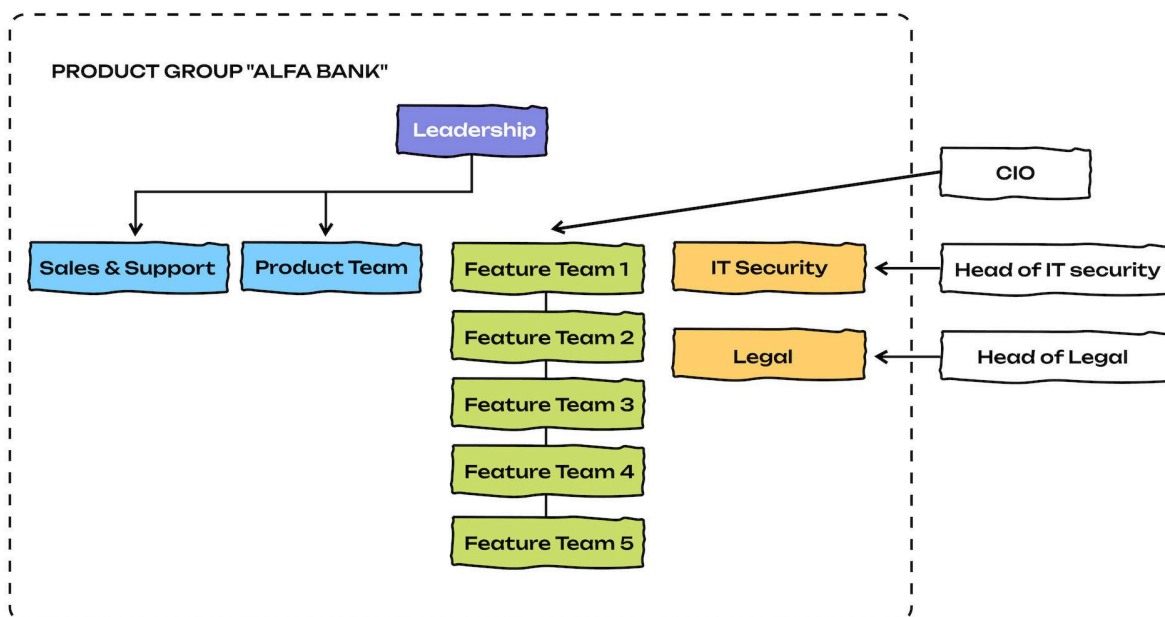

Closely related portfolio in a BigTech company

Leaders of these businesses own the strategy, development and performance of their domains. Their remit typically includes allocated business resources: commercial functions (sales and account management), part of marketing, and financial expertise

focused on the specific product group to help it achieve its targets. At the same time, core Product & Tech functions remain centralized. There is a common technology landscape and product culture spanning all domains. A single platform and centralized Product & Tech create a strong engineering culture and standardization – shared approaches to architecture, development and operations, as well as unified principles for working with data and quality.

This model also gives the company flexibility: product and engineering teams can support multiple domains and shift between them depending on the company-wide priorities, not just the needs of a single product group. Centralization of Product & Tech also sets constraints for domain leaders. Key decisions on architecture, platforms and engineering standards are made at the central level, so domain leaders cannot fully redesign the technology landscape or product practices solely in the interest of their own area and must account for company-wide priorities and standards.

Bank example. In Creating Agile Organizations, the SME business at Eastern European bank was used as an example of a product group in a closely related portfolio. A significant part of its capabilities, including the core development block, was supplied by centralized functions. Only sales, support and a small group of product managers reported directly to the Product Owner of this product group.
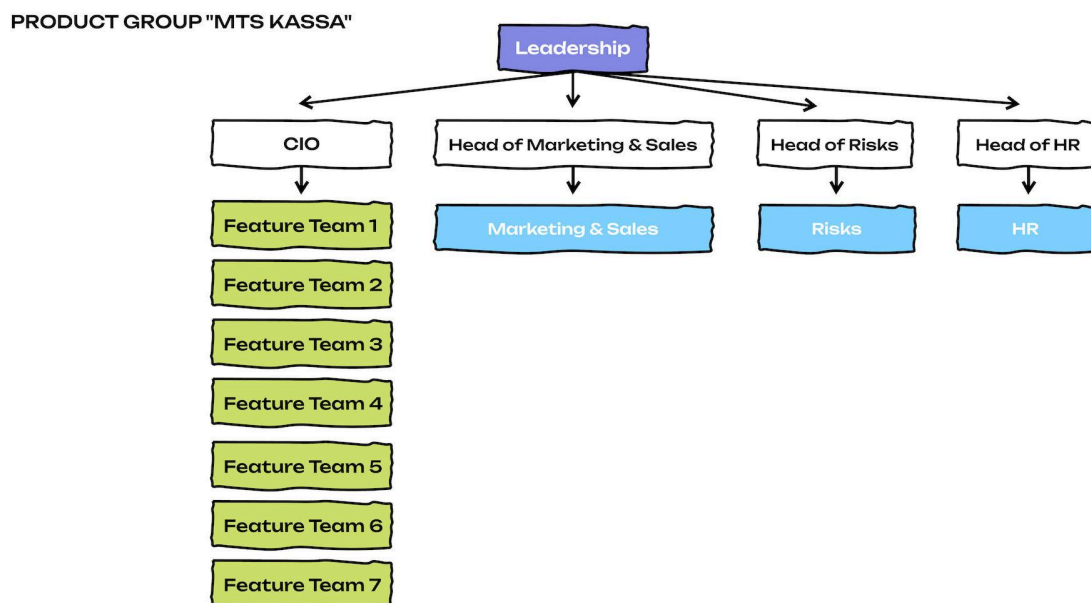
Product Group SME in a Bank

Product experts and some sales and support functions reported directly to the head of the product group, while development (five feature teams) and functions such as security, legal and operations were provided via central functions.

3. Loosely related portfolio

In a loosely linked portfolio, the company operates diverse, autonomous businesses with limited synergies. Business units fully own their strategy and execution, and rely on only a minimal set of shared functions (regulatory affairs, basic shared services) and some cultural harmonization. Examples include *Unilever*, *Aditya Birla Group*, *United Technologies*.

Example: MTS Kassa. That was a semi-autonomous product group inside the MTS corporation. The business had its own product team and engineers, as well as its own sales, marketing, legal and risk functions. MTS Kassa did not rely on powerful shared product platforms and services of MTS: its technology landscape and development processes were built independently, which created space for transformation.
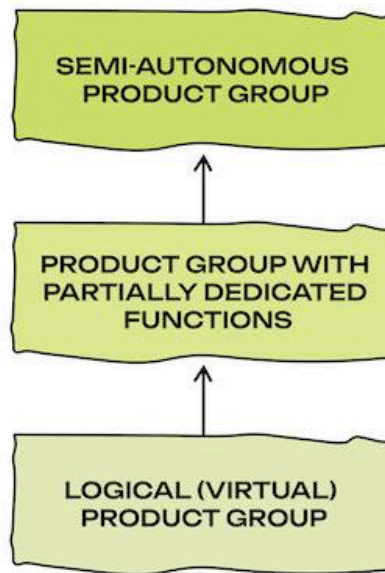

Product Group of MTS Kassa

Within a loosely related portfolio, this effectively means a semi-autonomous product group, and this type is particularly convenient when planning a transformation. In most cases, the support of the product group head, who controls the product, key resources and priorities, is sufficient. In more centralized models (single integrated business and closely related portfolio), this is no longer enough: due to heavy dependence on shared functions, platforms and standards, explicit support from the board and functional leaders is required.

Designing product groups: types and trade-offs

Using the portfolio-connectedness perspective, three types of product groups can be distinguished, differing in the degree of formalization and autonomy of the container.

- **A Logical (virtual) product group** relies mainly on coordinating roles, goals, programs and reward systems; people remain in their functional homes, and the product group acts as a logical layer above them.
- **A product group with partially dedicated functions** has some key capabilities (for example, product, marketing, UI/UX, core engineers, analytics) dedicated to the group, while other functions are still shared.
- **A semi-autonomous product group** brings most core business functions (product, tech, operations, marketing, sales, support, etc.) into one container, while selectively using shared corporate platforms and standards.

Three types of product groups

In practice, companies rarely choose between "full centralization" and "full autonomy"; most operate somewhere along this continuum. The three types of product groups can be viewed as points on a design spectrum that balances **differentiation** and **speed** at the product-group level against **efficiency** and **synergy** from shared functions. In more closely related portfolios, it is often effective to keep product groups more logical or partially dedicated and rely on strong centralized capabilities, whereas in more loosely related portfolios semi-autonomous groups become both feasible and advantageous. The key insight is that the choice of product-group type is not an ideological stance about centralization, but a pragmatic response to portfolio connectedness and the company's strategy.

References

1. Galbraith's Star Model Explained in Depth
2. Organization Design Challenges / Amy Kates
3. Treacy-Wiersema Value Disciplines Framework
4. Jay Galbraith. Designing Your Organization
5. Jay Galbraith. The Star Model™
6. Spencer Stuart. Product Organization Structure
7. The Remarkable Chief Engineer
8. How Apple Is Organized for Innovation (Harvard Business Review)