International Journal of Management, IT & Engineering
Vol. 16 Issue 02, February 2026,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Bridging the Gap Between Developer Needs and Product Management Principles: Challenges in Platform Product Management

Vishal Jain*

Abstract (12pt)

Developer-centric platform products, such as APIs, SDKs, and internal developer platforms, present unique challenges for product management. Unlike traditional consumer-facing products, where the end user is the primary customer, platform products position developers as primary customers while end users remain indirect beneficiaries. This conceptual paper investigates the tensions and trade-offs inherent in managing such platforms and proposes a practical framework to guide product managers of platform products.

The paper advances implications for both practice and research. For practitioners, it emphasizes multi-stakeholder centricity, ecosystem-oriented metrics, technical literacy, and transparent trade-off management. For researchers, it extends product management theory to platform contexts, integrates platform ecosystem and product management research streams, and identifies opportunities for empirical validation. By adopting a practice-informed yet scholarly perspective, the study bridges theory and real-world application, providing a foundation for future research and practical guidance in developer-centric platform management.

Keywords:

Platform Product Management; Developer Experience; Product Management Framework;

Author correspondence:

Vishal Jain,
Senior Product Manager, Technical
Amazon.com, Seattla, WA, USA
Email: vishaljain25@gmail.com

1. Introduction (12pt)

Product management emphasizes customer-centricity, focusing on understanding end-user needs and translating them into product features that drive adoption and business value. However, in the context of developer-centric platform products—such as public APIs, software development kits (SDKs), internal developer platforms, and cloud infrastructure, this assumption requires significant adaptation. In these environments, developers act as primary customers, who help deliver value to downstream end users. This shift introduces unique challenges for product managers as they need to manage indirect value creation, navigate multi-stakeholder trade-offs while maintaining platform governance and stability.

This paper provides a framework for developer-centric platform product management. The framework identifies four key dimensions: developer experience enablement, platform governance and stability, value translation to end users, and business and strategic alignment. The objectives are to extend product management theory to the domain of developer-centric platforms and provide a tool for managers navigating platform-specific challenges.

2. Product Management Landscape

2.1. Product Management Foundations

Explaining Product management as a discipline has traditionally been grounded in the principle of customer-centricity, emphasizing the identification of customer needs, market opportunities, and the delivery of value through well-defined products and services. Classic product management literature positions the product manager as the "voice of the customer," responsible for balancing customer needs, business goals, and technical feasibility. Frameworks such as the product lifecycle, market segmentation, and value proposition design have been widely adopted to guide decision-making in product development.

In conventional product contexts, the customer is typically the end user who directly interacts with and derives value from the product. Success metrics are often closely tied to user adoption, satisfaction, retention, and revenue generation. However, these foundational assumptions become less straightforward in platform-based products, where value creation and consumption are distributed across multiple stakeholders.

2.2. Platform Products and Multi-Sided Markets

Platform products differ fundamentally from traditional standalone products. In such systems, value is co-created by the platform owner and external participants who build, extend, or integrate with the platform. Platform owners must manage not only the core product but also the rules, interfaces, and incentives that govern participation. This introduces governance challenges, including decisions around openness, control, standardization, and pricing. From a product management perspective, platforms require long-term ecosystem thinking rather than short-term feature optimization.

2.3. Developers as Primary Customers

A defining characteristic of many modern technology platforms is that software developers function as the primary customers of the platform, even when the ultimate value is delivered to end users. Examples include application programming interfaces (APIs), software development kits (SDKs), cloud infrastructure platforms, and internal enterprise platforms. In these cases, developers consume the platform's capabilities to build applications, services, or integrations that serve downstream users.

Unlike traditional consumers, developers are both users and producers, making their relationship with the platform more complex. Their adoption decisions are influenced not only by functional requirements but also by learning costs, switching costs, and long-term maintainability.

This developer-centric dynamic challenges the traditional notion of customer-centricity in product management. The platform product manager must deeply understand developer workflows, technical constraints, and productivity concerns, while simultaneously ensuring that the platform enables positive outcomes for end users and aligns with broader business objectives.

2.4. Tension Between Developer Needs and End-User Value

One recurring theme is the inherent tension between optimizing for developers and optimizing for end users. Enhancements that improve developer flexibility or speed may introduce complexity, performance overhead, or inconsistent user experiences at the application level. Conversely, constraints imposed to protect end-user experience may limit developer creativity or increase development effort.

Misalignment between platform capabilities and developer needs can lead to reduced adoption, fragmentation, or the emergence of competing platforms. Product managers operating in this environment must make trade-offs that are not always visible through traditional customer research methods. The indirect relationship between platform decisions and end-user outcomes complicates prioritization and impact assessment.

2.5. Product Management Challenges in Platform Contexts

The challenges faced by product managers working on platform products include the difficulty in defining the target customer, challenges in collecting actionable feedback, and ambiguity in measuring success. Because developers act as intermediaries, feedback from end users is often filtered or delayed, reducing its usefulness for platform-level decision-making.

Additionally, platform product managers must coordinate closely with engineering teams to ensure architectural decisions align with product strategy. Technical debt, backward compatibility, and interface stability are more critical in platforms than in consumer-facing products, as changes can have cascading effects across the ecosystem. This places additional responsibility on product managers to understand technical architecture and long-term system evolution.

3. Framework

3.1. Purpose and Scope of the Framework

The purpose of this framework is to articulate how product management principles can be adapted for developer-centric platform products. It provides a structured lens for understanding decision-making in platform product management and offers a foundation for future empirical validation.

3.2 Core Assumptions of Developer-Centric Platforms

The framework is built on three core assumptions derived from the literature. First, value creation in platform products is indirect and mediated by developers, meaning that platform success depends on the quality, productivity, and engagement of the developer ecosystem. Second, platform product decisions have systemic effects, as changes to interfaces, tooling, or governance can propagate across multiple applications and user experiences. Third, product managers operate within a multi-stakeholder environment, requiring continuous negotiation between developer needs, business objectives, technical constraints, and end-user value.

These assumptions distinguish developer-centric platforms from traditional product environments and justify the need for a specialized conceptual model.

3.3 Key Dimensions

The proposed framework consists of four interrelated dimensions that collectively shape product management effectiveness in developer-centric platforms.

3.3.1 Developer Experience Enablement

Developer experience enablement refers to the platform's ability to support developers in efficiently building, deploying, and maintaining solutions. This dimension encompasses usability of APIs and SDKs, quality of documentation, availability of tooling, reliability, and support mechanisms. From a product management perspective, this dimension reframes customer-centricity around developer workflows and productivity rather than traditional user satisfaction metrics.

Good developer experience reduces cognitive and operational friction for developers, lowering adoption barriers and increasing long-term ecosystem participation. However, excessive focus on developer experience may introduce architectural complexity or compromise platform consistency, highlighting the need for balanced decision-making.

3.3.2 Platform Governance and Stability

Platform governance and stability capture the rules, standards, and constraints that regulate ecosystem participation. This includes interface contracts, backward compatibility guarantees, versioning strategies, security policies, and deprecation practices. Product managers must ensure platform stability to maintain developer trust, as unexpected changes can disrupt dependent applications and erode ecosystem confidence.

This dimension emphasizes the temporal responsibility of platform product management, where decisions must account for long-term consequences rather than short-term optimization. Governance mechanisms serve as a stabilizing force but may limit rapid experimentation or customization.

3.3.3 Value Translation to End Users

Value translation refers to the mechanisms through which platform capabilities ultimately deliver value to end users through developer-built applications. Because platform teams rarely interact directly with end users, this dimension addresses the challenge of ensuring that developer-focused decisions align with downstream user outcomes such as usability, performance, and reliability.

Product managers must rely on indirect signals, such as developer adoption patterns, integration quality, and partner feedback, to assess end-user impact. This dimension highlights the need for proxy metrics and inferential reasoning in platform product management.

3.3.4 Business and Strategic Alignment

Business and strategic alignment ensure that platform investments support organizational goals such as growth, monetization, risk management, and competitive differentiation. Platform product managers must balance ecosystem openness with control mechanisms that protect business value. Monetization strategies, pricing models, and partner incentives are central considerations within this dimension.

Tensions often arise when business objectives conflict with developer preferences, such as when monetization introduces friction or restricts access. The framework positions product management as an integrative function that mediates these conflicts through transparent trade-offs and long-term ecosystem thinking.

3.4 Dynamic Trade-Offs and Interdependencies

A central contribution of the framework is its emphasis on dynamic trade-offs among the four dimensions. Improvements in developer experience may strain governance mechanisms, while stricter governance can slow developer innovation. Similarly, business-driven decisions may negatively affect developer trust or end-user outcomes if not carefully managed.

The framework conceptualizes platform product management as an ongoing balancing act rather than a static optimization problem. Product managers must continuously reassess priorities as the ecosystem evolves, technologies change, and stakeholder expectations shift.

3.5 Role of the Product Manager

Within this framework, the product manager functions as a system integrator rather than a traditional feature owner. Key responsibilities include evaluating stakeholder inputs, translating abstract business goals into platform capabilities, and anticipating second-order effects of product decisions. This role requires a combination of strategic thinking, technical literacy, and stakeholder negotiation skills.

The framework thus reframes product management competence in platform contexts as the ability to manage complexity and interdependence rather than direct customer optimization.

4. Implications for Product Management Practice

The conceptual framework and archetypes developed in this paper have several important implications for product management practice, particularly for practitioners operating in developer-centric platform environments. These implications translate the theoretical insights into actionable guidance while remaining grounded in scholarly reasoning rather than prescriptive tactics.

Table 1: Mapping Framework Dimensions to Key Product Management Activities

Dimension	Product Management Activities	Examples of Decisions
Developer Experience Enablement	Developer onboarding, documentation, tooling	API design, SDK updates, developer support channels
Platform Governance and Stability	Version control, compliance, backward compatibility	Interface standards, deprecation policies, security
Value Translation to End Users	Monitoring end-user impact, feedback loops	Proxy metrics, integration testing, UX alignment
Business and Strategic Alignment	Roadmap prioritization, monetization, ecosystem incentives	Pricing models, partner incentives, strategic trade-offs

4.1 Reframing Customer-Centricity as Multi-Stakeholder Centricity

In developer-centric platforms, treating developers as the sole “customer” risks neglecting downstream end-user outcomes, while focusing exclusively on end users ignores the realities of platform adoption and ecosystem health. Product managers should therefore adopt a multi-stakeholder centricity mindset, explicitly recognizing developers, end users, business leaders, and internal engineering teams as interdependent stakeholders.

In practice, this implies designing discovery processes that incorporate developer workflow analysis, platform usage telemetry, and indirect signals of end-user impact. Product managers must become adept at synthesizing incomplete and asymmetric feedback rather than relying on direct user research alone.

4.2 Shifting Success Metrics Toward Ecosystem Health

Traditional product success metrics such as feature adoption or short-term revenue are often insufficient indicators of platform effectiveness. The framework suggests that product managers should incorporate ecosystem-oriented metrics, including developer retention, integration depth, platform reliability, and long-term adoption trends.

From a practical standpoint, this requires close collaboration with data and engineering teams to define proxy measures that reflect developer productivity and downstream value creation. Product managers should also communicate the limitations of short-term metrics to stakeholders, emphasizing the long-term nature of platform investments.

4.3 Strengthening Technical Literacy

Because platform decisions are deeply intertwined with system architecture, product managers must understand concepts such as modularity, backward compatibility, and interface stability to make informed trade-offs.

This does not imply that product managers must assume engineering roles, but rather that effective platform product management requires sufficient technical fluency to anticipate second-order effects and engage credibly in architectural discussions. Organizations may need to adjust role expectations and training to support this expanded competency.

4.4 Managing Trade-Offs Transparently and Deliberately

Trade-offs among developer experience, governance, end-user value, and business objectives are unavoidable. Product managers should make these trade-offs explicit rather than implicit. Articulating the rationale behind decisions—such as prioritizing stability over speed or control over openness—can build trust within the developer community and across internal stakeholders.

This transparency also supports organizational learning, enabling teams to revisit and revise assumptions as platform conditions evolve.

4.5 Redefining the Product Manager Role for Platforms

Finally, there needs to be a shift in how the product manager role is defined and evaluated in platform contexts. Rather than being assessed primarily on feature delivery or roadmap execution, platform product managers should be recognized for their ability to manage complexity, align stakeholders, and steward ecosystem health over time.

For practitioners, this reframing can provide language and conceptual grounding to explain the unique demands of platform product management to organizational leaders, supporting more realistic expectations and sustainable practices.

4. Conclusion

Four key dimensions; (1) developer experience enablement, (2) platform governance and stability, (2) value translation to end users, and (4) business and strategic alignment; collectively shape the effectiveness of platform product management.

In conclusion, the paper advances understanding of developer-centric platform product management by providing a coherent conceptual model that bridges theory and practice. It underscores the importance of treating platform product management as a distinct domain requiring nuanced decision-making, multi-stakeholder awareness, and strategic ecosystem stewardship, thereby offering a foundation for managerial guidance in an increasingly digital and platform-driven economy.

References

[1] Baldwin, C. Y., & Woodard, C. J. (2009). The architecture of platforms: A unified view. Platforms, Markets and Innovation, 19(3), 19-44.

[2] Cusumano, M. A., Gawer, A., & Yoffie, D. B. (2019). *The business of platforms: Strategy in the age of digital competition, innovation, and power*. Harper Business.

[3] Eisenmann, T., Parker, G., & Van Alstyne, M. (2006). Strategies for two-sided markets. *Harvard Business Review*, 84(10), 92-101.

[4] Fichman, R. G., Dos Santos, B. L., & Zheng, Z. E. (2014). Digital innovation as a fundamental and powerful concept in the information systems curriculum. *MIS Quarterly*, 38(2), 329-353.

[5] Gawer, A. (2014). Bridging differing perspectives on technological platforms: Toward an integrative framework. *Research Policy*, 43(7), 1239-1249.

[6] Gawer, A., & Cusumano, M. A. (2014). Industry platforms and ecosystem innovation. *Journal of Product Innovation Management*, 31(3), 417-433.

[7] He, Q., & Chen, W. (2020). Developer experience in software platforms: A literature review. *Journal of Systems and Software*, 164, 110546.

[8] Iansiti, M., & Levien, R. (2004). Strategy as ecology. *Harvard Business Review*, 82(3), 68-78.

[9] Meyer, A. D., Gaba, V., & Colwell, K. A. (2005). Organizing far from equilibrium: Nonlinear change in organizational fields. *Organization Science*, 16(5), 456-473.

[10] Moore, G. A. (1991). *Crossing the chasm: Marketing and selling high-tech products to mainstream customers*. HarperBusiness.

[11] Parker, G. G., Van Alstyne, M. W., & Choudary, S. P. (2016). *Platform revolution: How networked markets are transforming the economy and how to make them work for you*. W. W. Norton & Company.

[12] Schilling, M. A. (2017). *Strategic management of technological innovation* (5th ed.). McGraw-Hill Education.

[13] Tiwana, A. (2014). *Platform ecosystems: Aligning architecture, governance, and strategy*. Morgan Kaufmann.

[14] Von Hippel, E. (2005). *Democratizing innovation*. MIT Press.

[15] Yoo, Y., Henfridsson, O., & Lyytinen, K. (2010). Research commentary—The new organizing logic of digital innovation: An agenda for information systems research. *Information Systems Research*, 21(4), 724-735.