Gartner®

How to Govern and Fund Platforms in a Product World

Published 8 September 2021 - ID G00754839 - 7 min read

By Analyst(s): Bill Swanton, Akis Sklavounakis

Initiatives: Software Engineering Strategies; Software Engineering Technologies

Applications and software engineering leaders are adopting product governance and funding for applications directly used by customers, but struggle applying these to underlying platforms. The solution: Have platform teams treat product teams as "customers," and select among three governance models.

Overview

Key Findings

- Product lines often use the services of various platforms, but the term platform is overused for various technical, cloud and application services, which causes confusion over how to govern and fund different platforms.

- Like any agile team, platform teams need to understand their customers, but the real internal or external customer (i.e., user of the product) is one (or more) steps removed from the team.

- Because the platforms are not visible to the end user, software engineering leaders and platform product managers struggle to justify their team's budgets and priorities to a variety of business stakeholders.

Recommendations

Application and software engineering leaders implementing product and platform governance, funding, and budget models should:

- Identify platforms based on the type of services they provide and what teams use their services.

- Direct platform teams to treat the platform as a product and consider the product or feature teams as their customers. Use the number of teams consuming it to choose among a direct, customer advisory board or mass market approach to governance and product management.
- Budget for each platform, based on the type of services, the scope of use, the degree its use can be attributed to specific activities and your company's overall approach to cost allocation.

Introduction

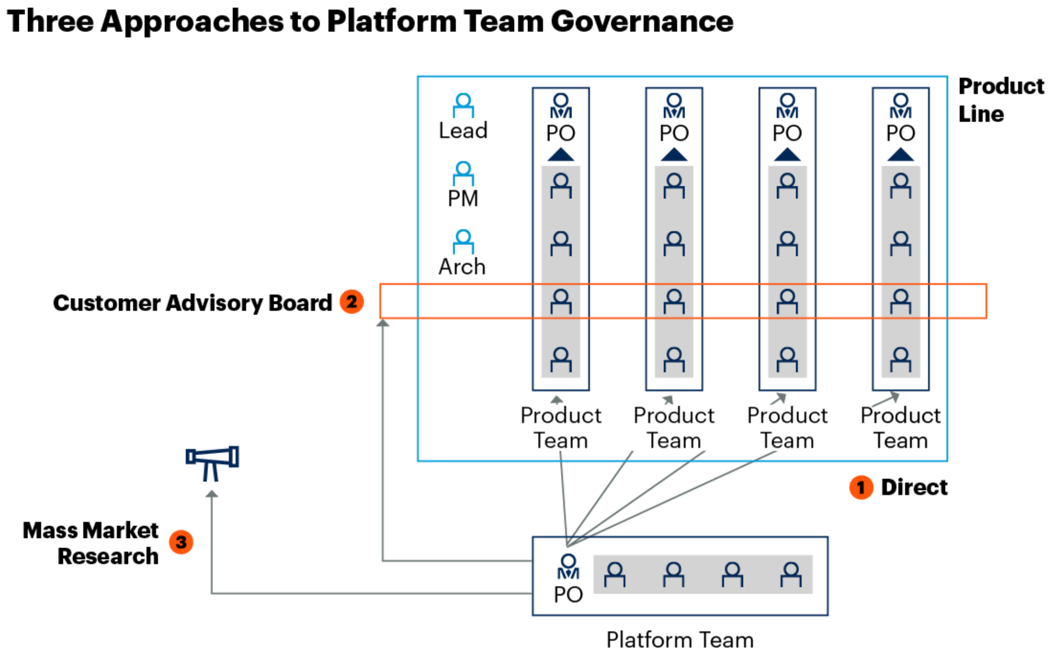
Platform is an overused term, with more than 60 different architectures identified as platforms in Gartner research. However, most modern application architectures involve multiple types of platforms used by product (or feature) teams to build their software.

Platforms are often a shared service providing capabilities needed by many product teams, thus simplifying product development. They may also be a full development environment for building and running software.

Governing platforms is tricky because the many product teams sharing them may have different priorities. And the last thing you want is to have the platform team building features the product teams don't need! Figure 1 shows three approaches to governance, which will be described in more detail below.

Because platforms are shared by product teams, but are out of sight of business stakeholders, software engineering leaders may struggle to justify the staff, priorities and budget for platform teams. The key to making the staff and funding decisions is to evaluate the nature and use of your platforms, then to apply the appropriate governance and funding model to each based on its situation.

Figure 1: Three Approaches to Platform Team Governance



Source: Gartner (August 2021)

Lead = Product Line Leader; PO = Product Owner; PM = Product Manager; Arch = Architect or Lead Engineer

754839_C

Analysis

Identify Platforms Based on the Type of Services They Provide

When deciding on a governance framework, it helps to understand the nature of the platform and how it is consumed. Platforms may be fully built within the company, or they may incorporate software from vendors or open-source projects. In any case, there is a team in your company responsible for delivering that platform.

Although there are many different types of platforms in use, they can be roughly divided into two basic types — development and transactional.

Development platforms provide development capabilities to product teams, with software defined services to manage, build and run the code. Examples include:

- Low code/no code application platforms (LCAP)
- Platform as a service (PaaS), including the continuous integration/continuous delivery (CI/CD) toolchain

- DevOps value stream delivery platform (VSDP), providing a fully integrated set of capabilities to enable continuous delivery of value
- SaaS platforms — such as Salesforce and ServiceNow — where people both configure applications in SaaS and write extensions in the accompanying PaaS
- Design systems for building user experiences
- API management platforms

Transaction platforms provide services via APIs provided by application components in the IT landscape. These may include:

- Applications custom-built with microservices, mini-services and so on
- Complicated subsystems within products
- Traditional applications accessed through APIs instead of their native user interface, such as an existing ERP system or wrapped mainframe application
- Commercial off-the-shelf transaction platforms (e.g., commerce or content management) accessed either via their native APIs or via APIs added by the platform team to provide an isolation layer.

The type of platform will often indicate how widely it is used, which will affect the governance approach used.

Product Teams Are the Platform Team's Customers

Platform teams should use the same agile product delivery methods as product teams. They have a product owner or manager, developers and a scrum master, and they work from a product roadmap and backlog. The difference is that instead of the customer being an end user or external customer, the platform team's customers are the product teams that use the platform.

The number of product teams using the platform drives how it is governed. Figure 1 shows the three approaches described below:

- **For fewer than 10 product teams using the platform,** the platform product owner should be able to work with each team individually to set the roadmap. Priorities can be resolved in the context of a release planning meeting with the product and platform teams involved (case "A").
- **For platforms with a large number of product team customers,** implement a community analogous to software advisory boards used by some vendors. Representatives from product teams can meet and discuss new features and relative priorities with the platform product owner (case "B").
- **For a widely used platform in a very large development organization,** the platform product owner will have to adopt mass market research techniques to figure out what the customers want. Surveys and dialogues with key customers will be combined with the product manager's vision to set and prioritize features (case "C").

The platform roadmap will be like a product roadmap, but the business outcomes may not be directly measurable with business metrics (see How to Use Product Roadmaps for Funding and Governance of Agile Product Delivery Teams). Metrics might be binary (adding the ability to do something teams need), performance-related (e.g., SLAs, response times, scalability) or adoption-related (e.g., how many teams use the feature or how much they use it).

> Ultimately the platform team's performance should be judged in how well it helps the product teams meet their objectives.

Two alternative approaches to getting platform work done are:

- **Virtual platform team:** Rather than a dedicated team working on the platform, members of the teams using the platform work on it part time as a virtual team. This is common for development platforms and shared capabilities.
- **Innersource:** The platforms or shared capabilities are managed as a company-specific open-source project, with teams contributing to and using code from the repository (see A CTO's Guide to Top Practices for Open-Source Software).

The platform's success relies on reducing product development and delivery friction and providing a compelling value proposition compared to alternatives. If the platform is perceived as adding delays, limiting solution options and restricting innovation, it will not be used and will fail. Common pitfalls include developing something in a vacuum and expecting teams to adopt it or delivering so slowly that teams can make faster progress on their own. The key to any of these governance approaches is for the product teams to feel they are being listened to and allowed to help shape the roadmap.

Budget for Platform Based on Scope and Usage

Find a logical home for the platform team in the budget based on where the product teams using it are funded. This may be in a narrow product line, within a functional segment of the product budget (such as sales/marketing, production or field support) or in some broad IT budget.

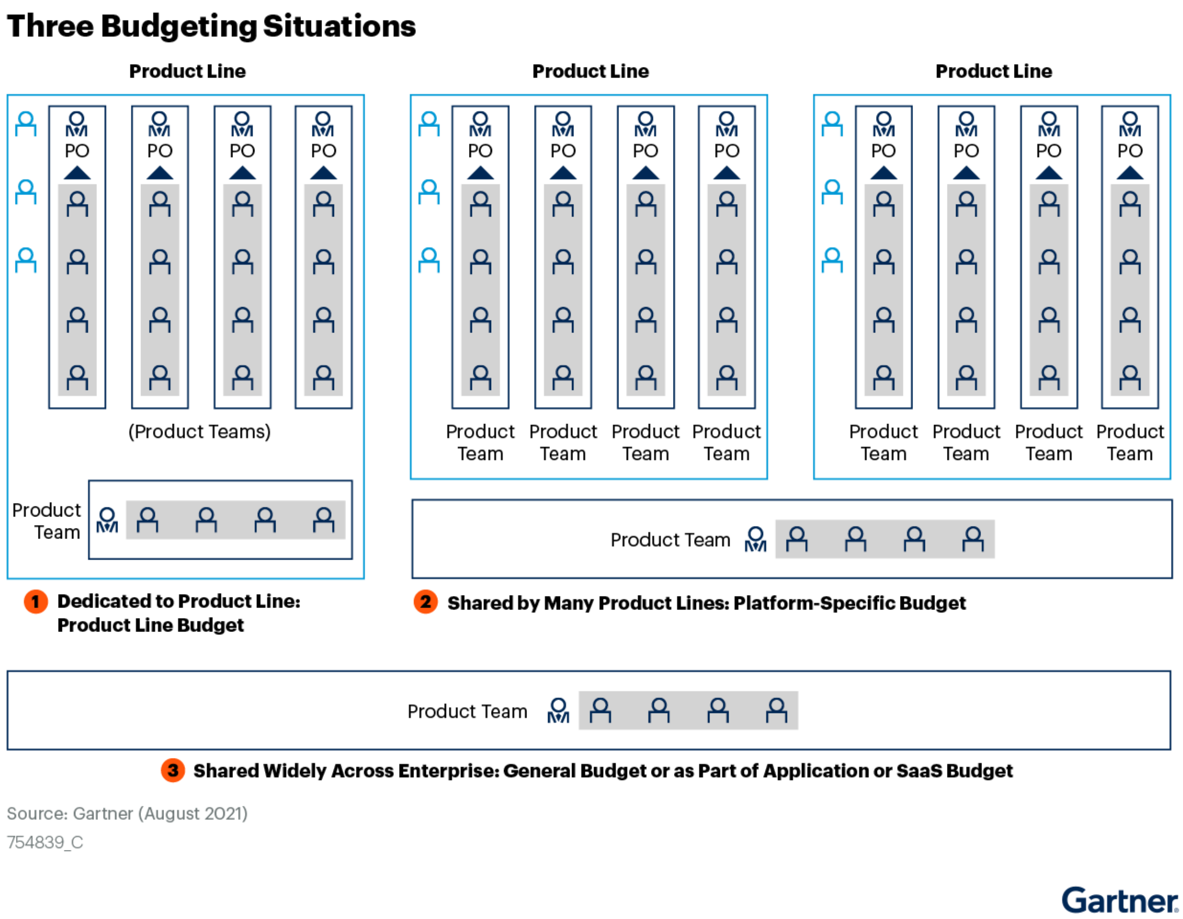
Three common situations are shown in Figure 2:

- **If the platform is only used by a single product line,** make the team part of the product line for planning and budgeting. This is common where product line teams decide to avoid the duplication of work by each team and rely on complicated subsystem teams or product-line-specific transactional platform teams (case "A").

- **If the platform is used across multiple product lines,** it will need a platform-specific budget set up within the overall IT budget or within a specific functional part of the overall product budget. This is common for development platforms, API management platforms, cloud-native application platforms or private PaaS (case "B").

- **Enterprisewide platforms are typically built on traditional systems of record or SaaS platforms.** Embed the budget for work needed to support them as platforms in the budget for that application (case "C").

Figure 2: Three Budgeting Situations


Three Budgeting Situations
Product Line
PO
(Product Teams)
Product Team
Product Line
PO
Product Team
Product Line
PO
Product Team
Product Team
1 Dedicated to Product Line: Product Line Budget
2 Shared by Many Product Lines: Platform-Specific Budget
Product Team
3 Shared Widely Across Enterprise: General Budget or as Part of Application or SaaS Budget
Source: Gartner (August 2021)
754839_C
Gartner

Source: Gartner (August 2021)

Evidence

This research is based on the personal experiences of Gartner analysts who have been part of agile development organizations and more than 1,000 interactions on platforms, PlatformOps and value stream development platforms with clients over the past two years.

Recommended by the Authors

Some documents may not be available as part of your current Gartner subscription.

Using Platform Ops to Scale and Accelerate DevOps Adoption

How to Manage and Market Platforms as Products for DevOps Teams

4 Principles for Designing Your Product Portfolio for Product-Centric Software Delivery

Building a Platform for Product Team Productivity (adidas)

Design and Build the New Digital Foundations

How to Build Agile Infrastructure Platforms That Enable Rapid Product Innovation

How to Use Product Roadmaps for Funding and Governance of Agile Product Delivery Teams

A CTO's Guide to Top Practices for Open-Source Software

Build a Focused ITFM Cost Model and Use a Simple Allocation Approach to Deliver Rapid Value