iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

API-FIRST CAPABILITY PLATFORMING AND INNOVATION PORTFOLIO GOVERNANCE

Basil Obute
ORCID ID - 0009-0001-1348-4572
PhD Student, Information Technology Department, University of the Cumberlands, Kentucky, USA

Nzeribe A. Okeh
ORCID ID - 0009-0008-6358-1718
PhD Student, Information Technology Department, University of the Cumberlands, Kentucky, USA.

Kingsley C. Ugwu
ORCID ID - 0009-0002-4061-5563
PhD Student, Artificial Intelligence, University of the Cumberlands, Kentucky, USA.

ABSTRACT
Application programming interface (API) portfolio governance has become a significant yet untheorized organizational capability. Although application programming interfaces (APIs) have been increasingly used to support an "API-first" strategy, innovation research continues to view APIs primarily as technical tools for integrating systems rather than as strategic instruments for governing innovation portfolios. This study provides a conceptual framework to position API portfolio governance, including governance of lifecycle decisions, enforcement mechanisms, economic portfolio analysis, and standards for the developer experience, as a high-level dynamic capability that links capability platform architecture to the outcomes of innovation portfolios. This study uses a systematic literature review of 127 articles to identify 7 enterprise cases (including one negative example in which governance was intentionally reduced) and provides evidence consistent with the operation of governance mechanisms across those organizations. In this study, we identified 4 governance mechanisms: enforcing modularity within the architecture of capabilities, creating options through combinations of capabilities, regulating boundaries of ecosystems contingent upon the level of externalization strategy used by an organization, and calibrating innovation velocity. In addition to identifying these governance mechanisms, we found that developer experience governance served as a cross-cutting moderator, shaping the mechanisms' effectiveness. We found a relationship across all of our cases such that higher levels of API governance maturity were positively correlated with better proximal indicators of portfolio performance (i.e., reuse rates, onboarding time, trend in defects) and reported by informants as having improved their ability to get products to market faster and to achieve an optimal balance between exploration and exploitation; however, due to the cross-sectional nature of our study, we cannot definitively state the causal relationships among these variables. In addition to providing a mechanism-based nomological network with testable propositions, we offer a transparent diagnostic rubric and systematic analysis of alternative explanations for each of the 7 enterprise cases.

Keywords:
API-First Strategy, Capability Platforming, Innovation Portfolio Governance, Dynamic Capabilities, Developer Experience, API Product Management, Reuse Economics.

1. INTRODUCTION

The proliferation of APIs has dramatically changed the way organizations build, deploy, and manage their digital innovations. As such, the use of an API-first strategy is widespread in large enterprise companies [1]. However, the study of innovation management as it relates to portfolio-level governance mechanisms for determining whether API portfolios create strategic value or simply represent technical sprawl has been largely ignored in the literature on innovation management.

There exist two distinct bodies of research. On one hand, software engineering researchers examine the technical aspects of API lifecycle management, API design patterns, and the security issues associated with APIs [2]. On the other hand, innovation management researchers study portfolio governance using a variety of

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

frameworks, including project selection and resource allocation; however, these researchers typically do not consider the architectural structure underlying digital innovation [3]. This results in a significant theoretical gap. There remains a need to develop conceptualizations of governance practices that link API-level architectural decision-making processes to portfolio-level innovation outcomes.

There are several frameworks that provide some level of governance for enterprises. COBIT provides enterprise-level governance principles, while TOGAF provides architectural governance processes [34, 38]. However, neither of these frameworks provides any guidance on how the governance of an organization's various digital building blocks (such as APIs), treated as products with road maps, customers, and economic characteristics, translates into innovation outcomes at the enterprise level. The literature on internal developer platforms has begun to address aspects of the infrastructure governance of digital building blocks, but it has not examined the strategic implications of those building blocks at the portfolio level [35, 36, 39].

This paper attempts to fill a gap in the literature on the relationship between the governance of digital building blocks and strategic outcomes for innovation portfolios by developing and empirically validating the construct of API portfolio governance as a higher-order dynamic capability operating at the capability platform layer between infrastructure and project governance. We define API portfolio governance as the decision rights, policies, the enforcement mechanisms, and portfolio economics through which an organization governs the creation, evolution, reuse, and retirement of API products from multiple producer teams. The central research question is: What API portfolio governance mechanisms are associated with enterprise innovation portfolio outcomes? A secondary research question that emerged during the analysis is: How should these mechanisms be sequenced and aligned to maximize governance effectiveness? This emergent question is addressed in Section 6.2 (Figure 2) and Section 6.3.

Contributions

In addition to identifying a set of 16 best practice indicators to guide developers in their selection of APIs for their applications, this study also contributes to our understanding of the role that APIs can play in supporting innovation by making three contributions to both the field of innovation management and the field of Information Systems:

Conceptual: This study is the first to define API Portfolio Governance as a distinct governance concept and to articulate its distinctions from other forms of governance (such as IT governance using COBIT, Enterprise Architecture Governance using TOGAF, and Platform Engineering/IDP).

Theoretical: This study introduces a mechanism-based framework (M1-M4) and a cross-cutting developer experience moderator, and articulates a nomological network of testable and measurable propositions related to API Portfolio Governance.

Diagnostic/Empirical: This study uses triangulation to gather evidence from a systematic literature review and seven enterprise cases. It also provides a transparent diagnostic rubric with operational indicators to enable researchers to replicate the study, and includes a systematic analysis of alternative explanations (Appendix B) to document rival explanations for each case, thereby increasing the interpretive credibility of the findings.

Boundary Regulation (M3) is explicitly modeled as contingent on the firm's exposure to the outside environment or on high autonomy levels among its complementors. Therefore, the remainder of this paper will proceed as follows: Section 2 describes the theoretical foundation for this study; Section 3 describes the development of the conceptual framework; Section 4 describes the methodologies used in this study; Section 5 describes the empirical findings of the study; Section 6 discusses the implications of these findings; and Section 7 summarizes the findings of the study and draws conclusions based upon them.

2. THEORETICAL FOUNDATIONS

2.1 Dynamic Capability Theory and Digital Platform

Teece, Pisano, and Shuen [4] developed the "dynamic capability" theoretical approach, which Teece later refined [5]. The dynamic capability theory is based on the ability of organizations to recognize opportunities and respond to changes to maintain a competitive edge in turbulent environments. The dynamic capability theory has been applied to digital platforms; it now includes the idea that the ability to organize and manage complementary assets across ecosystem boundaries is a key competency [6]. In the digital transformation research stream, researchers have differentiated between "ordinary" capabilities, which allow companies to maintain their current business model, and "dynamic" capabilities, which support long-term strategic change [7]. As mentioned above, API portfolio governance is positioned differently as it acts as a "meta-capability", or an intermediary mechanism, which determines how individual technical capabilities can be used together to produce new and higher-level innovation outcomes. This aligns with Teece's [5] assertion that the development

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

of sensing, seizing, and reconfiguring abilities requires formalized governance mechanisms rather than informal, ad hoc ones. Therefore, we will consider governance as "dynamic" only if the governance mechanisms institutionalize repeated patterns of routine activities in the areas of (i) sensing portfolio signals (trends in consumption, defects, and onboarding); (ii) seizing through explicit investment and decision-making rights (funding reuse, deprecation, and productization); and (iii) reconfiguring the capability platform (refactoring boundaries, revising contractual obligations, recalibrating release velocity) in reaction to changed opportunity and risk profiles. The governance mechanisms described in Section 3 provide micro-foundational elements that implement the sensing, seizing, and reconfiguring activities at the capability platform level.

Research on platform governance has primarily focused on external ecosystems and multi-sided markets [8,43]. However, Gawer and Cusumano [9] identified "platform leadership" as a competency that involves creating and managing architectures, standards, and ecosystem orchestration, but they analyzed these issues within the context of inter-firm dynamics rather than intra-organizational governance of internal API platforms. Research by Tuominen and Martinsuo [10] has emphasized the importance of governance actor constellations in portfolio governance; however, they did not extend to architectural governance of digital building blocks.

2.2 API Economy & API Product Management

The API economy has evolved from a specialized technology issue into a strategic priority. Today, APIs are viewed much the same way other products are: they have roadmaps, customer (developer) feedback loops, and Service Level Agreements (SLAs) [1] in place, much as other companies manage their products. As such, this productization creates governance challenges related to managing a product portfolio similar to those experienced by companies using a product portfolio approach to managing products; e.g., balancing investment between exploratory and exploitative API products, determining when to release new versions and when to discontinue older versions, and establishing prices to encourage developers to consume APIs within an organization [11]. API product management is the application of product management principles to design and govern interfaces (APIs); API product managers differentiate between designing APIs as products, where APIs are designed to provide specific value propositions to developer customers (API-as-product), and designing APIs as infrastructure (API-as-Infrastructure). These distinctions carry important governance implications as product-oriented management allows organizations to create feedback mechanisms and collect usage analytics to develop intelligence about their API portfolios at the portfolio level. Research in software engineering and information systems governance consistently identifies API portfolio visibility, the degree to which an organization can observe, catalog, and act on the state of its full API inventory, as an unsolved operational and strategic challenge [14, 15, 16]. Industry maturity models typically evaluate governance based on three distinct areas: i.e., design consistency, lifecycle management, and security, which are complementary to but do not encompass all of the innovation portfolio governance mechanisms that we have theoretically identified here. Therefore, our theoretical framework offers additional guidance by linking governance practices to innovation portfolio outcomes through mechanisms grounded in theory.

2.3 Developer Experience as Cross-Cutting Moderating Mechanism

(DX) is increasingly recognized as a key component of the effectiveness of internal platforms [35]. As such, DX represents the user-friendliness, discoverability, documentation quality, amount of friction in onboarding, and the speed at which developers receive feedback while using APIs as products [36]. Although DX is often viewed as a design concern secondary to the architectural aspects of APIs, recent studies have demonstrated that DX directly impacts adoption rates, reuse behaviors, and overall developer satisfaction, all of which also affect the portfolio outcomes discussed in this study [37]. In addition, the role of automated tooling in shaping developer workflow practices highlights how DX can extend far beyond documentation into the day-to-day practice of developing [40].

As a cross-cutting moderating mechanism, DX serves as a cross-cutting factor, rather than a stand-alone mechanism. For example, poor DX will diminish the effectiveness of the other four mechanisms. Poorly-documented APIs will become mere "compliance theater" as developers simply find ways to bypass them; poorly-designed marketplace interfaces will fail to facilitate discovery of new options created through combinatorial option-creation processes; complex onboarding processes for partners will stifle boundary-regulation efforts; and poor transition-migration tooling will create friction, rather than stability, for velocity-calibration processes. On the other hand, high levels of DX maturity will amplify the effectiveness of governance by diminishing the costs associated with adoption. Therefore, we include DX as a moderating variable within our conceptual framework.

2.4 Innovation Portfolio Management and Governance.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Innovation Portfolio Management (IPM) encompasses the processes, governance structures, and decision-making frameworks an organization uses to allocate resources across innovation initiatives to create the greatest strategic value [14]. Most traditional IPM research has focused on project financial valuations, the use of stage-gate development processes, and the strategic allocation of funds [15]. However, the emerging body of literature on digital innovation identifies several unique challenges it presents to IPM, such as layered architectures, the generative potential of platforms, and faster experimentation cycles [16].

More importantly, while there are many extant IPM models for selecting projects and allocating resources, they assume that the technological infrastructure supporting those innovations is either provided by the organization itself or is independent of the organization's portfolio management [3]. As such, these assumptions are no longer valid when innovations rely on shared API platforms for building on one another, and when the governance of those platforms directly impacts the nature of the innovations that can be developed, how quickly they can be developed, and at what cost. Salonen et al. [17] contend that because of these characteristics, the design principles for managing portfolios of digital innovations require significantly different designs, particularly governance mechanisms that accommodate modular and recombinant digital artifacts. Their research shows that platform-based governance mechanisms may also help create greater strategic alignment among investments within a portfolio; however, their model did not provide a solution for using the API layer as a governance mechanism. Therefore, this is the gap that we seek to fill with our research.

2.5 Differentiation from Existing Frameworks

It will be important to specifically differentiate API portfolio governance from established frameworks; therefore, this can be done with a summary of the differences, as shown in Table 1.

Table 1. Differentiation of API portfolio governance from established frameworks

Dimension	COBIT [34]	TOGAF [38]	Platform Eng. [35,36]	API Portfolio Gov. (this paper)
Governance scope	Enterprise IT governance	Enterprise architecture method	Internal developer infrastructure	API products as innovation building blocks
Unit of analysis	IT processes and controls	Architecture building blocks	Developer toolchains and platforms	API product portfolios
Abstraction layer	Enterprise / strategic	Architecture/design	Infrastructure/operations	Capability platform (between infrastructure and projects)
Innovation linkage	Indirect (risk/value alignment)	Indirect (architecture principles)	Indirect (developer productivity)	Direct (portfolio outcomes theorized)
Economic mechanisms	Cost optimization, risk mgmt	Not addressed	Build vs. buy decisions	Reuse economics, option value, internal pricing
DX integration	Not addressed	Not addressed	Central concern	Cross-cutting moderator

Table 1 demonstrates that the structure of API portfolio governance has an analytical distinction of its own. While COBIT [34] offers a general framework for enterprise-level governance principles and does not go down to the level of the API product nor develop theoretical models of innovation portfolio results, TOGAF [38] focuses on governing architectural design processes and does not focus on ongoing governance of production API portfolios or their economic characteristics. While the internal developer platform literature [31, 35, 36] and work on developer portals as an innovation supporter [39] discuss developer-centric infrastructure governance, they do not relate to strategic portfolio management theory. API portfolio governance integrates the theoretical

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

constructs of all three traditions, the institutionalization of governance from COBIT, the modularity of architecture from TOGAF, and developer-centricity from platform engineering, while introducing a new theoretical model that describes the linkages between the governance of architecture and innovation portfolio performance. It is not the individual practices that are novel, but the combination of these practices, specifically located in the API domain, and their joint operation at the capability platform layer that produce emergent portfolio effects.

Terminology Note: We will refer to our Capability Platform as the internal platform layer that packages reusable capabilities (frequently through APIs) for multiple initiative teams to consume. The internal developer platform, internal developer portal, and other similar elements can be considered enablers of DX and should therefore not be used interchangeably with the term API Portfolio Governance. An API Product is an interface treated as a product with a lifecycle, consumers, and quality-of-service commitments. An API Product Portfolio is a collection of products managed under a specific organizational structure.

3. CONCEPTUAL FRAMEWORK

The conceptual framework describes how an organization's decision-making regarding the architecture of its API platforms is influenced by the organization's innovation portfolio performance through the organizational capabilities (i.e., governance) that govern the API portfolio. The framework provides four unique governance mechanisms (each with unique practices) that ultimately transform the governance at the level of individual APIs into governance for the entire portfolio. Additionally, the governance for developer experience across all four governance mechanisms will moderate this relationship. As shown in Figure 1, the model is illustrated.

Figure 1. Conceptual framework linking API portfolio governance to innovation portfolio outcomes.

3.1 Construct Scope and Boundaries

The portfolio governance of an organization's API portfolio entails a collection of decision-making authority and lifecycle policies such as versioning and deprecation that govern how decisions on ownership and accountability for a portfolio of APIs are made in addition to mechanisms used to enforce adherence to those decisions such as reviews and policy as code gates and also how funding, chargeback, and prioritization decisions are made as it relates to the use of APIs across many development teams. The characteristics of a governance system that qualify this type of system include: (1) APIs are formally recognized as a portfolio of products with specific lifecycle states (e.g., create/evolve/reuse/retire); (2) Decision rights and incentives are extended beyond a single product team's backlog to all product teams both creating/consuming the APIs; and (3) Conformance is enforced by operationally viable control systems (i.e., automation, review gates, exception management) rather than aspirational guidelines or practices. Portfolio governance innovation is treated as an external outcome of the governance properties listed above rather than as part of their definition.

Governance locus and decision rights: In addition to determining where decision-making authority for an API portfolio resides, the actual process of governing an API portfolio is structured in layers: (i) The first layer consists of enterprise or business unit portfolio authorities that determine how much money will be invested into an API portfolio, and what specific goals should be achieved by this portfolio (e.g., reuse, risk posture, externalization strategy). (ii) The second layer includes individual API product owners, who develop their own API roadmap and manage the lifecycle of their respective APIs. (iii) A third layer enforces compliance with standards established by the organization's API CoE, Architecture Review Board, Platform Engineering, and Policy-As-Code Gates. When the authors refer to "the" governance construct at the portfolio level, they mean it exists only when decision rights span beyond a single platform team and can be enforced across all the producer teams that produce APIs in that portfolio.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Discriminant Boundary Tests for Governance of APIs: (a) The API Management Operations (gateways, authentication, monitoring) provide the basic infrastructure for API Management to operate [12]. Without Portfolio Decision Rights and Lifecycle Governance, these are not API Portfolio Governance. (b) Enterprise Architecture Governance can establish Principles and Target States for the Architecture, but without Interface-Contract Conformance and Portfolio-Level Ownership and Incentives, this is still Enterprise Architecture Governance. (c) Platform Engineering/IDP Product Management can function without API Portfolio Decision Rights across the Enterprise; it will be considered API Portfolio Governance once it governs API Products outside the Platform Team's Own Backlog and enforces portfolio-wide Lifecycle and Reuse Rules.

Exclusions and Interfaces: Data governance and cybersecurity governance are not addressed as stand-alone domains here unless the data governance and/or cybersecurity governance manifests as an API Portfolio Decision Rule(s) at some point during the API Product Lifecycle (i.e., tiered based on the level of external exposure, minimum contractual security control thresholds, privacy by design requirements in API Standards etc.).

The Unit of Analysis: This model uses the enterprise API portfolio (or a clearly defined business-unit portfolio) and its capability platform layer as the unit of analysis. The model is best suited for organizations that have many-to-many producer-consumer relationships, which are likely to be able to achieve some level of reuse across their various initiatives, and large enough to create governance externalities; in practice, this generally equates to portfolios of >200 active APIs or multiple independent product teams using a common capability.

3.2 Architectural Modularity Enforcement (M1)

Modularity enforcement is how a company enforces modularity by creating rules for an application programming interface (API)-encapsulated service, enabling many combinations using those services. Examples of this include setting up a mandatory design board for APIs, a validation gate within a continuous integration/deployment pipeline, and a backward-compatibility test. The most important feature of this enforcement is that it focuses on structural integrity (preserving the boundaries of an API, ensuring that contracts do not break, keeping coupling low). Modularity enforcement is a prerequisite for reuse because it provides architectural preconditions (interfaces will be clean, contracts will be stable, coupling will be low) for reuse at large scales, but it does not provide reuse behavior or an economic incentive for reuse. One hypothesized outcome of modularity enforcement is increased initiative independence (the ability to run more initiatives concurrently).

Operational definition: Present when interfaces have standardized contract definitions for which there is active verification of compliance with those definitions to allow for localized change and predictable consumer upgrade efforts.

Proximal metrics: Consumer Breakage Rate per Release; Mean Time to Remediate Integration Defects; Upgrade Latency (Time from Deprecation Notice to Consumer Migration); Change Failure Rate for API Releases

Failure modes: Standards Exist but Lack Enforcement (Limited Automation, Broad Exceptions); Shadow APIs and Undocumented Endpoints Grow in Number; Deprecations Are Announced but Not Implemented, Creating Indefinite Version Sprawl.

3.3 Combinatorial Option Creation (M2)

The practice of combinatorial option creation reflects how organizations structure their APIs as building blocks of value that can be used to generate real options for future growth and innovation. While M1 governs how good an organization's API boundaries are structured, M2 governs the economic and behavioral factors that encourage and support the discovery, consumption, and reuse of modules across organizational boundaries. Internal curation of an internal marketplace; developer portal-based discoverability standards; and economic incentive structures based on credits awarded to creating teams for each time they create an API that is subsequently consumed by other teams are examples of such key practices [20, 21]. Thus, an organization may have modularized its APIs so that they are internally well-structured (i.e., high M1), but still have little reuse, because there has been no effort at either curation or providing incentives to use them (i.e., low M2), a situation exemplified by Case Study Zeta (Section 5.3). Every modularized API product represents a very-low-cost opportunity to compose novel digital products using previously developed capabilities. The hypothesized result is an increase in portfolio throughput.

Operational Definition: The portfolio's use of a registry to increase the number of safe reusable capability-building blocks will lower the marginal cost of combining new solutions across initiatives.

Proximal Metrics: Reuse Rate (unique consumers using an API); Time to First Successful Call; Share of new initiatives using pre-existing APIs versus creating net-new APIs.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Failure Modes: Registry has been created, but is out of date and therefore untrusted; Incentives for local delivery only, therefore producers do not invest enough in reusable interfaces; Overly general design will create friction, thereby decreasing actual reuse.

3.4 Ecosystem Boundary Regulation (M3, Contingent)

Boundary regulation describes how organizations determine which APIs to expose to the outside world, in what form, and how those exposures will affect internal priorities. Unlike M1, M2, and M4 — which are validated as the three core mechanisms across the full case set — M3 is presented as a theorized contingent extension supported empirically by a single case (Gamma). It is included in the framework because it is theoretically coherent and practically important for organizations with external API programs; however, it requires targeted future validation before being elevated to co-equal status with the core mechanisms. Some examples of boundary regulation practices include productization tiers (public/internal/partner), rate limiting and monetization strategies, and governance processes related to onboarding partners [22, 23]. Through effective boundary regulation, organizations can selectively open their capability platforms to external innovators to generate additional innovation capacity while retaining ownership of their core capabilities. The outcome is that the company can access additional innovation opportunities through complementors' contributions, depending on how the organization chooses to expose its capabilities to them. Organizations typically develop their boundary regulations as a later-stage capability that builds upon existing internal modularity and marketplace governance, as the exposure to external consumers increases the cost of inconsistency.

Scope conditions. Conditions that make M3 salient are: (S1) The organization is actively pursuing an intentional level of external exposure; (S2) Complementors have a high degree of autonomy; (S3) Regulatory, safety, or privacy requirements necessitate enforced limits on usage; or (S4) Monetization produces significant negative consequences if not controlled. If none of the above conditions exist, then M3 is likely to be in a nascent state but will not negatively impact internal performance.

Proximal Metrics: Onboarding Cycle Time for New Complementors by Tier; Policy Violation Rate and Remediation Time; Risk-Adjusted Growth of Complementors.

Failure Modes: Overly restrictive gates to integration will slow the rate of adoption and drive teams to work around the gates through unofficial integrations; Loose tiers with inconsistent contracts will lead to unpredictable levels of cost and effort for supporting the relationships between the company and its complementors; Compliance reviews that occur after the decision to open the capabilities to public use, rather than before, will drive additional levels of unnecessary rework.

3.5 Innovation Velocity Calibration (M4)

Velocity Calibration establishes the rate of change in an API Portfolio to achieve a balance between speed and stability. Examples of velocity calibration practices include version cadence policy, deprecation timelines, breaking-change governance committees, and DX Quality Gates [24, 25]. The absence of velocity calibration can allow fast-moving teams to disrupt dependent projects, and an overcautious approach to governance can stifle innovation. The theoretically expected outcome of velocity calibration is optimal exploration-exploitation trade-offs.

Operational Definition: This occurs when an organization's governance intentionally sets release cadence, compatibility standards, and investments at levels that enable its portfolio to maintain throughput while avoiding accumulation of debilitating technical and coordination debt.

Proximity Metrics: Release Frequency vs Change Failure Rate; Lead Time Variance (Predictability); Deprecation Backlog Age; Version Sprawl; Upgrade Effort per Release.

Failure Modes: Excessive speed without sufficient compatibility standards results in a high degree of churn and loss of reuse; Overly restrictive freeze windows result in project bottlenecks or "shadow" delivery; No capacity reserved for the eventual retirement of APIs results in chronic version sprawl.

3.6 Developer Experience as Cross-Cutting Moderator

High levels of DX maturity result in more effective use of governance mechanisms; therefore, DX maturity does not directly affect portfolio outcomes but instead influences how governance mechanisms are utilized, determining whether they can be effectively implemented to achieve desired portfolio outcomes. In other words, when DX maturity is low, poorly designed governance mechanisms will have little to no portfolio effect due to "adoption resistance" (i.e., developers will resist using them). Conversely, when DX maturity is high, even well-designed governance mechanisms will have a greater portfolio effect because developers will likely accept and use them appropriately.

Separation of the moderating effect of option creation from the moderating effect of the developer experience. The moderator effect of framing as described by the moderator effect of the configuration is shown through

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

configurations that are based upon observation: (a) the ability to create high levels of developer experience while having limited opportunity for the creation of options can be seen in cases where developers are investing in documentation and software development kit (SDK) quality but there is no marketplace and therefore no economic benefits for reuse of options - developers have a positive experience with options but the signals for reuse of those options are still under-developed in terms of institutionalization; (b) strong levels of structure in creating options and poor levels of developer experience can be found in cases where there is a formal catalog and incentives for creating new options but due to onboarding friction, adoption rates of these options remain suppressed which results in duplication of effort among teams although they may have formal governance processes in place to prevent this duplication. These two configurations separate the moderation of developer experience (cost of adoption) from M2 (recombinant portfolio).

3.7 Inter-Mechanism Dynamics

The four mechanisms operate interdependently, with each influencing the operation of the others. Three specific relationships are important to understand how the mechanisms interact. First, there is a positive relationship between M1 (API boundary definition) and M2 (marketplace consumption): when developers have a clear understanding of what an API does, this improves the ability of applications to work together (composability). As developers build new applications using existing modular components, the usage data from those applications provides evidence that the modularity has value, encouraging continued investment in its governance. Second, there is a negative relationship between M2 (marketplace consumption) and M4 (versioning and deprecation): if developers aggressively introduce new versions of their components or remove old ones, it can create uncertainty about whether future applications will be able to continue to use prior versions of the components. This creates a trade-off between the need to rapidly move forward with innovation (speed) and the need to provide long-term stability to support reuse across multiple applications. Third, M3 (external exposure) acts as a governance multiplier because it increases the risk associated with all other mechanisms by creating a dependency outside of the organization's control. When an organization exposes its APIs to users outside of the organization, there are higher expectations placed on all of the other mechanisms. Specifically, modularity (M1) must be implemented more strictly to protect against unintended consequences. Option architectures (M2) must be designed to accommodate potential changes made by third-party consumers, rather than simply being developed based on internal assumptions. And velocity policies (M4) must be developed more conservatively to limit the rate at which new options are introduced. Therefore, the effectiveness of governance does not solely depend on the maturity of individual mechanisms, but also on the degree to which the mechanisms align with one another—what we refer to as governance alignment. The cross-case analysis explores these dynamics empirically. Table 2a summarizes the mechanisms, their theoretical foundations, and theorized portfolio outcomes.

Mechanism	Core Practices	Theoretical Basis	Portfolio Outcome
M1 Modularity enforcement	Design review boards, schema validation, backward-compatibility testing	Modularity theory [18], architectural innovation [19]	Higher initiative independence
M2 Option creation	Marketplace curation, discoverability standards, reuse incentives	Real options [20], recombinant innovation [21]	Accelerated portfolio throughput
M3 Boundary regulation (contingent)	Productization tiers, rate limiting, partner onboarding	Platform ecosystem theory [22], boundary resources [23]	Expanded innovation scope (conditional)
M4 Velocity calibration	Versioning cadence, deprecation timelines, DX quality gates	Ambidexterity theory [24], governance modes [25]	Exploration–exploitation balance
DX governance (moderator)	Documentation standards, onboarding governance, SDK policies, feedback loops	Internal developer platforms [35, 36, 37, 40]	Amplifies/attenuates all mechanism outcomes

Table 2a. Governance mechanisms, practices, theoretical bases, and theorized portfolio outcomes.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

3.8 Propositions and Nomological Network

To establish theoretical causal claims using our cross-sectional, qualitative data, we will establish theoretically grounded propositions that describe expected associations of our findings. The propositions we develop will represent a theoretical network of relationships that link mechanisms to both intermediate (proximal) measures of portfolio performance and to final (distal) measures of portfolio performance related to innovation.

P1 (Enforcement of Modularity → Independence of Initiatives): There is an association between the enforcement of modularity in a system and the degree of inter-team coupling in the system; and there is also an association between the enforcement of modularity and the number of cross-initiative coordination shocks that occur within the system. Therefore, a system that enforces modularity enables greater initiative and independence within the system. Indicators of this relationship may be represented by the following: a) few cross-team change approvals per release, b) low consumer breakage rates, c) high deprecation compliance, and d) low mean time to remediate integration defects.

P2 (Option Creation → Innovation Throughput): The creation of options through combinations of individual components can lead to a decrease in marginal build costs when creating new projects/initiatives, and an increase in innovation throughput. Indicators which are expected to be positive include: a greater share of newly created projects that reuse pre-existing APIs; a decreased time-to-first-successful-call for new consumers accessing the system; and an increase in the rate at which the scope of new projects expands as a result of reusing existing components.

P3 (Boundary regulation → scope expansion under control, preliminary): Boundary regulation can be strategic to increase exposure to an outside environment or an autonomous internal ecosystem. In such cases, the use of boundary regulation is correlated with a higher number of complementary participants, at a lower risk-adjusted cost for incidents and support. The expected indicator metrics are as follows: faster tier-based onboarding of new partners, lower rate of reported abuses/incidents per active key, and a consistent Service Level Agreement (SLA) and predictable service/support demand. There is little evidence base for this proposed relationship. Therefore, further research and validation of this relationship in companies that have established mature External Application Programming Interface (API) programs is necessary.

P4 (Velocity calibration → time-to-market with stability): More effective Velocity calibration is associated with faster and more predictable time-to-market while maintaining compatibility and reuse of code. Expected indicators include a higher release frequency with stable change failure rate, lower upgrade latency, and controlled version sprawl.

P5 (Dx as Moderator): The positive moderation of developer experience governance on P1-P4 indicates that good developer experience governance reduces the cost of adoption and compliance. When developer experience is poor, the effect of the mechanisms may be reduced. This will be evidenced through the use of low-quality documentation or SDKs and weak feedback loops, which will reduce the take-up rate regardless of the existence of governance standards.

P6 (Inter-Mechanism Tension): The development of options and the calibration of velocity for option creation can negatively impact modularity enforcement and boundary regulation where there are weak standards and/or a lack of lifecycle discipline. Good governance (Governance Alignment) will help to reduce inter-mechanism tension. Specifically, inter-mechanism tension is expected to be evidenced through: (a) a negative correlation between the frequency at which APIs are released (M4) and the stability of reused APIs (M2); (b) an increase in breakage experienced by consumers (M1) as a result of accelerated deprecation (M4); and (c) greater variance in time to first call when the velocity of API versioning exceeds the ability of the migration framework to support it. Gamma's experience will provide empirical evidence to support this proposition.

Measurements Agenda: The Diagnostic Rubric in Appendix A could be used as an initial formative measure of Governance Maturity (an integrated bundle of practices that together define the construct) instead of as a reflective measure. Future development of the index may include validating and refining it through longitudinal, panel, or quasi-experimental designs such as staggered rollout of new policies, "policy-as-code" gate adoption, or natural experiments surrounding platform reorganization. Additionally, future validation of the index will likely involve linking Mechanism Scores to Proximal Measures and Innovation Portfolio Outcomes.

4. METHODOLOGY

4.1 Research Design

The research design for this study is a multi-phase qualitative design with embedded quantitative indicators [26] with two sequential phases. The first phase consists of a systematic literature review to provide theoretical

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

underpinnings and to identify dimensions of governance practices. The second phase comprises a multiple-case study to empirically support the conceptual framework and to explore the governance mechanisms employed in practice. Although both phases use qualitative methods, the sequential nature of the phases allows for a similar function to mixed-methods research by employing the literature-based conceptual framework to guide the case analyses, while allowing for emergent themes to develop during the case studies [27]. The researchers choose the term "multi-phase qualitative design" instead of "mixed-methods" to represent the fact that no quantitative data collection instruments were used in the study, and to acknowledge that all quantitative values in the results section (e.g., number of employees) were organizational metrics provided by the informant and documented as part of the study's documentation; however, they were not quantitative values obtained from any researcher-administered instrument.

4.2 Phase 1: Systematic Literature Review

The authors followed a PRISMA protocol in conducting this review and identified studies from five databases (Web of Science, Scopus, IEEE Xplore, ACM Digital Library, AIS Electronic Library) by using: ("API governance" OR "API management" OR "API-first") AND ("innovation" OR "portfolio" OR "platform") over the period January 2015 through December 2025. In total, they obtained 843 results at first. They removed duplicates from these results and screened titles and abstracts of the remaining 627 for eligibility according to the four inclusion criteria: (a) substantive treatment of API governance (i.e., more than just an incidental reference), (b) that provided evidence of organizational or strategic outcomes, (c) peer-reviewed journal articles, conference proceedings, or rigorously edited practitioner reports, and (d) were published in English. Articles that dealt only with API security, only with individual API design patterns, or were vendor whitepapers that did not clearly detail the methods used to develop their research were all eliminated from this study. Using the title alone as an elimination process resulted in the removal of 433 articles. The next step was to review the remaining 194 full-text articles to produce a final corpus of 127 articles: Software Engineering (n = 52), Information Systems (n = 41), and Management (n = 34).

Only rigorously edited practitioner reports with transparent methodology were retained and treated as contextual evidence to support the operationalization of constructs; these reports do not serve as a substitute for peer-reviewed theory. The thematic coding occurred in three iterations: first, an open coding iteration to identify governance practices; second, an axial coding iteration to group identified governance practices by categories (i.e., mechanism); third, a selective coding iteration to relate each identified mechanism to the respective portfolio outcome. A level of intercoder agreement was obtained on the thematic coding process using Cohen's Kappa (k=0.83) [28].

4.3 Phase 2: Multiple-Case Study

Seven organizations were sampled using purposeful sampling [29], due to: (a) they have been in an active API-first transformation for at least 2 years; (b) they have managed 200 or more production APIs; (c) they have a formal governance process for managing their innovation portfolios. To gain maximum variation in context: 2 of the cases were from the financial sector, 1 case was from the telecommunications sector, 1 case was from the health tech sector, 1 case was from the logistics sector, 1 case was from the retail tech sector, and 1 case was from the manufacturing sector. The seventh case (Eta) was a negative case, an organization that had intentionally dismantled formal API governance, to provide counterfactual evidence for many governance studies that lack this type of evidence. See Table 2 for a summary of the characteristics of each case.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Case	Industry	Employees	APIs	API-first tenure	Governance model	DX maturity	Interviews	Docs
Alpha	Financial svcs	45,000	1,200+	5 yrs	Centralized CoE	Mature	8	14
Beta	Financial svcs	12,000	450	3 yrs	Federated w/ central stds	Developing	7	9
Gamma	Telecom	80,000	2,800+	6 yrs	Platform team model	Mature	9	17
Delta	Healthcare tech	5,000	320	2 yrs	Emerging centralized	Nascent	6	6
Epsilon	Logistics	28,000	680	4 yrs	Federated w/ API guild	Developing	7	11
Zeta	Retail tech	9,500	510	3 yrs	Hybrid product-platform	Developing	5	8
Eta	Manufacturing	18,000	340	3 yrs	Centralized (rolled back)	Developing (regressed)	6	7

Table 2. Overview of case study organizations.

The data collection process consisted of 48 interviews for the seven cases, with each participant being an employee in one of five different roles: Chief Technology Officer, API Product Manager, Platform Engineer, Developer Experience Lead, Innovation Portfolio Manager. Each interview lasted approximately 45 – 75 minutes and utilized a structured protocol to assess the four governance mechanisms and developer experience practices. In addition to the interview data, there were 72 total documents collected from each case, including but not limited to API Governance Policy Documents, Developer Portal Analytics, Developer Satisfaction Survey Results, and Portfolio Review Minutes. All quantitative values reported within the results (e.g., reuse percentages, defect reduction rates, integration time line improvements) were cross-checked utilizing at least two sources: interviewee responses were compared to data on organizational dashboards, internal analytics reports, or documented information from portfolio reviews provided by the case organization. When organizations supplied the authors directly with dashboard metrics, we cross-checked the metric(s) to ensure all participants reported consistent measurements, noting the definition(s) of each metric. We have marked all figures that could not be independently verified using documentable evidence as estimated figures based on interviewee responses. Transcriptions of all interviews were completed and analyzed using NVivo 14 following theoretically-informed thematic analysis [30].

4.4 Analytical Approach

Within-case analysis generated governance profiles of individual cases that mapped practices into the four mechanism categories and evaluated DX maturity as an independent moderator variable. Cross-case analysis used replication logic [29] to determine whether the hypothesized mechanisms appeared systematically and to identify contingency factors. Pattern matching examined whether observed relationships between governance practices and outcomes conformed with the theoretical propositions of the framework.

We also documented for each case the possible confounding organizational context factors, IT budget trajectory, leadership turnover, other contemporaneous transformation efforts, and industry regulatory changes that may have had an independent impact on both governance practice and outcomes from the portfolio. Although qualitative methods are incapable of statistically controlling for such confounding factors, our cross-case comparative analysis enabled us to document cases in which contextual differences suggested alternative explanations for observed patterns. We include documentation of these alternative explanations along with our primary findings in Appendix B. The use of this systematic approach to documenting alternative explanations was a deliberate methodological choice made to enhance the interpretive credibility of observed relationships between governance practices and outcomes.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

The two researchers individually coded all of the transcripts of the interviews conducted in the study and achieved an initial (pre-consensus) inter-rater reliability of 79% across all mechanism–case cells. The majority of the coding disagreements occurred at the developing/mature boundary and specifically related to M2 and M4, where distinguishing between partially enforcing mechanisms and institutionally enforcing mechanisms was particularly difficult without referring to supporting documentary evidence. Following structured consensus discussions using the Rubric for Consensus and Documented Evidence found in Appendix A, the inter-rater reliability of the two researchers improved to 91%. The remaining 9% of the disagreements between the researchers were resolved through a third-party decision-maker and documented within the audit trail.

4.5 Evidence Transparency and Temporal Plausibility

To be documented, we needed an internally generated object that was operationally used and could be evaluated independently of interview narrative data. These would include documentation artifacts such as governance policies, an architecture review template, versioning notices for all applications, configuration of CI/CD gates, analytics from the developer portal, a portfolio review pack, and an incident summary. In order for a numerical figure to be documented, it had to be supported by at least one other document artifact that contained information that described the same event as described by the interviewee, and in addition, there had to be confirmation from another source (informant) that defined the measurement and indicated a positive or negative direction of change.

We organized each of our governance outcome statements into three levels of evidence quality to limit the causal language we could use. Evidence quality was ranked using the following three-tier system: Tier A (the outcome corroborates temporal progression; corroborative artifact that has a date stamp for the outcome is available and the corroborative artifact indicates the outcome occurred after a dated governance action); Tier B (the outcome corroborates directionality, but ambiguity exists regarding timing; directionality is assumed to be consistent with but not necessarily a result of); Tier C (outcome estimates are based on interview responses alone; no corroborative artifacts exist). In an effort to minimize the effects of confirmation bias, maturity scoring was conducted before a cross-case narrative was developed, and coders relied upon mechanism definitions and artifacts and not on outcome statements when performing maturity scoring.

5. FINDINGS

5.1 Cross-Case Governance Patterns

Table 3a. Cross-Case Summary: Industry Context, Governance Mechanism Maturity, and Key Reported Outcome Indicators (All Seven Cases)

Case	Industry	Employees	Prod. APIs	API-first Tenure	Governance Model	Mechanism Maturity Ratings (Table 3)					Key Reported Outcome Indicators	Case Type
						M1 Modularity	M2 Options	M3 Boundaries	M4 Velocity	DX Mod.	*Bold = Tier A documentary evidence; plain = Tier B corroborated or Tier C informant-reported. ↓ = regressed from prior maturity level.*	
Alpha	Financial Servic	45,000	1,200+	5 yrs	Centralized CoE	**Mature**	**Mature**	**Developing**	**Mature**	**Mature**	**−34% cross-initiative integration defects (Tier A: Jira, 12-mo pre/post gate)**	**Core** (Sections 5.2–5.5)

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Case	Industry	Employees	Prod. APIs	API-first Tenure	Governance Model	Mechanism Maturity Ratings (Table 3)					Key Reported Outcome Indicators	Case Type
						M1 Modularity	M2 Options	M3 Boundaries	M4 Velocity	DX Mod.	*Bold = Tier A documentary evidence; plain = Tier B corroborated or Tier C informant-reported. ↓ = regressed from prior maturity level.*	
	es										41% new initiatives using only pre-built components (up from 12%) (Tier A: portfolio tracking) Higher mechanism uptake; reduced shadow integrations vs. Zeta (DX amplification effect)	
Beta	Financial Services	12,000	450	3 yrs	Federated w/ central standards	**Developing**	**Developing**	**Nascent**	**Developing**	**Developing**	Federated model without interoperable standards produced local optimisations at the cost of portfolio coherence (Tier B: informant consensus) "Federation without interoperable standards produced individual optimizations for the local [business unit] at the expense of overall portfolio coherency." — API Programme Manager (Tier C)	**Deviant** (Section 5.6)
Gamma	Telecom	80,000	2,800+	6 yrs	Platform team model	**Mature**	**Mature**	**Mature**	**Developing**	**Mature**	**87% of production APIs tracked in marketplace (Tier A: portal analytics)** Avg. API usage per initiative: 1.7 → 4.3	**Core** (Sections 5.2–5.5)

ISSN: 2456-9348
Impact Factor: 8.232

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Case	Industry	Employees	Prod. APIs	API-first Tenure	Governance Model	Mechanism Maturity Ratings (Table 3)					Key Reported Outcome Indicators	Case Type
						M1 Modularity	M2 Options	M3 Boundaries	M4 Velocity	DX Mod.	*Bold = Tier A documentary evidence; plain = Tier B corroborated or Tier C informant-reported. ↓ = regressed from prior maturity level.*	
											(Tier A: portal analytics) 23% of app portfolio from partner-developed apps (Tier B) — sole M3 evidence case M2–M4 tension documented: aggressive versioning temporarily destabilised reuse; resolved via deprecation discipline (Tier B: CTO interview)	
Delta	Healthcare Tech	5,000	320	2 yrs	Emerging centralized	**Nascent**	**Nascent**	**Nascent**	**Nascent**	**Nascent**	**~30% of dev resources consumed reconciling overlapping APIs from premature M2-without-M1 implementation (Tier B: sprint retrospectives)** "Governance fatigue" reported after simultaneous rollout of all 4 mechanisms (Tier C: CTO) No dedicated API management staff; scattered	**Deviant** (Section 5.6)

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Case	Industry	Employees	Prod. APIs	API-first Tenure	Governance Model	Mechanism Maturity Ratings (Table 3)					Key Reported Outcome Indicators	Case Type
						M1 Modularity	M2 Options	M3 Boundaries	M4 Velocity	DX Mod.	*Bold = Tier A documentary evidence; plain = Tier B corroborated or Tier C informant-reported. ↓ = regressed from prior maturity level.*	
											responsibilities → inconsistent enforcement	
Epsilon	Logistics	28,000	680	4 yrs	Federated w/ API guild	**Developing**	**Mature**	**Developing**	**Mature**	**Developing**	Reuse-based resource allocation system increased API reuse adoption (Tier B) Avg. onboarding time: 4.2 days vs. 0.5 days for Gamma — DX friction suppressed M2 despite formal governance (Tier B: informant reports) Demonstrates DX suppression effect: M2 design maturity ≠ adoption outcomes without DX quality	**Core** (Sections 5.2–5.5)
Zeta	Retail Tech	9,500	510	3 yrs	Hybrid product-platform	**Mature**	**Developing**	**Developing**	**Developing**	**Developing**	**Mature M1 but weak DX → developers bypassed governed APIs via "shadow integration" (Tier B)** No marketplace or reuse incentives → high-quality APIs remained organisationally invisible (M1–M2 analytical	**Core** (Sections 5.2–5.5)

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Case	Industry	Employees	Prod. APIs	API-first Tenure	Governance Model	Mechanism Maturity Ratings (Table 3)					Key Reported Outcome Indicators	Case Type
						M1 Modularity	M2 Options	M3 Boundaries	M4 Velocity	DX Mod.	*Bold = Tier A documentary evidence; plain = Tier B corroborated or Tier C informant-reported. ↓ = regressed from prior maturity level.*	
											independence case) Demonstrates that modularity (M1) is necessary but insufficient for option creation (M2) without curation infrastructure	
Eta	Manufacturing	18,000	340 (+60 post-rollback)	3 yrs	Centralized (rolled back)	**Developing ↓**	**Nascent ↓**	**Nascent**	**Nascent ↓**	**Developing ↓**	**Portfolio grew 280 → 340 APIs; ~40% functional duplication of new APIs after rollback (Tier A: lead architect + incident system)** Surge in cross-team coordination incidents documented in incident management system (Tier A) "I know now that when we took away what we called 'unnecessary' bureaucracy we were really taking away the glue that held all of our teams together." — CTO (Tier C) Strongest counterfactual evidence in study: governance removal	**Deviant** / Counter-factual (Section 5.6)

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Case	Industry	Employees	Prod. APIs	API-first Tenure	Governance Model	Mechanism Maturity Ratings (Table 3)					Key Reported Outcome Indicators	Case Type
						M1 Modularity	M2 Options	M3 Boundaries	M4 Velocity	DX Mod.	*Bold = Tier A documentary evidence; plain = Tier B corroborated or Tier C informant-reported. ↓ = regressed from prior maturity level.*	
											→ exact negative outcomes predicted by P1, P4, P6	

Note. Maturity ratings (Nascent / Developing / Mature) are reproduced from Table 3 and were determined by inter-rater coded analysis of interview transcripts and documentary triangulation (IRR = 91% post-consensus). Cell shading: green = Mature, amber = Developing, red = Nascent. ↓ denotes regression from a prior maturity level (Eta only). Outcome indicators use the three-tier evidence quality system from Section 4.5: Tier A = documentary evidence corroborated and temporally ordered; Tier B = corroborated but temporally ambiguous; Tier C = interview estimate only. Bold outcome text indicates Tier A documentary evidence. Yellow row shading (Beta, Delta, Eta) marks cases receiving focused deviant/negative case analysis in Section 5.6; all three are theoretically significant: Beta illustrates federated governance failure, Delta illustrates premature simultaneous rollout, and Eta provides the study's primary counterfactual evidence for P1, P4, and P6. "Prod. APIs" = number of production APIs under active governance at time of study. "API-first Tenure" = years since formal API-first programme launch. Docs = number of documentary artefacts collected and coded per case (from Table 2).

Of the seven case studies, six have demonstrated similar governance practices associated with the four proposed governance mechanisms, though at varying levels of maturity. A cross-case comparison for the maturity of each mechanism is provided in Table 3. The maturity was determined through coding of interviews and documentary triangulation, with each assigned to nascent, developing, or mature. A maturity assessment for DX was added to the primary four mechanisms, per the moderating effect discussed in Section 3.6, and an empirical pattern was found consistent with the theoretical moderation. In terms of the evidence density of the mechanisms and cases studied, the least evidence was found for boundary regulation (M3), which also had the lowest maturity of the four mechanisms, in all but one case (Gamma).

Case	M1: Modularity	M2: Options	M3: Boundaries	M4: Velocity	DX Gov.
Alpha	Mature	Mature	Developing	Mature	Mature
Beta	Developing	Developing	Nascent	Developing	Developing
Gamma	Mature	Mature	Mature	Developing	Mature

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Delta	Nascent	Nascent	Nascent	Nascent	Nascent
Epsilon	Developing	Mature	Developing	Mature	Developing
Zeta	Mature	Developing	Developing	Developing	Developing
Eta	Dev. (regressed)	Nascent (regressed)	Nascent	Nascent (regressed)	Dev. (regressed)

Table 3. Cross-case governance mechanism maturity assessment. Color coding: green = mature; amber = developing; red = nascent

5.2 Findings for Modularity Enforcement

Alpha, Gamma, and Zeta implemented mature modularity enforcement mechanisms. Mandatory design reviews were conducted by Alpha's CoE as part of their design process to assess if each initiative remained aligned within its respective domain boundaries; if each initiative maintained a consistent schema (as defined by the governing body); and if each initiative was compatible (backward-compatible) with the existing architecture of all other initiatives. This was directly related to a 34% decrease in cross-initiative integration defects (Tier A: documented via Jira-based defect tracking over twelve months prior to implementing gates and twelve months post-gate implementation). According to Alpha's Portfolio Manager: "Initiatives are able to proceed independently due to API contracts that provide guarantees for interoperability." While the Portfolio Manager attributed the ability for independent initiative progress to the API contracts, he acknowledged that this was also influenced by the reorganization of the platform team and improvements to testing infrastructure.

Delta has implemented some form of modularity enforcement; however, this is still a developing process. Unfortunately, Delta's modularity enforcement efforts resulted in many overlapping APIs, which consumed approximately 30% of the development resources available for reconciling these issues (Tier B: reported independently by three separate individuals; and while the exact percentage of development resources consumed is unknown, the trend was directionally supported by sprint retrospective summaries over the course of five quarters).

5.3 Combinatorial Option Creation Findings

Gamma and Epsilon had the most developed form of combinatorial governance in terms of how they utilized options; Gamma used its marketplace to track 87 percent of all production APIs (Tier A: as measured by marketplace portal analytics) and increased average usage of APIs from 1.7 to 4.3 APIs per initiative. Similarly, Epsilon created a reuse-based system where high reuse interface producers would receive priority for resource allocation. Alpha successfully implemented 41 percent of all new initiatives using only pre-built API components (up from 12 percent prior to implementing the marketplace formally [Tier A: internal portfolio tracking system]).

Zeta is an example of the M1-M2 analytical independence discussed above. Although Zeta has matured into an M2 environment (with regard to modularity enforcement), it remains at the M1 level of option creation due to the lack of both marketplace curation and reuse incentives. Therefore, although Zeta's team is creating high-quality, modularly designed APIs, they remain organizationally invisible -- other teams can build good interfaces but do not have the ability to find them or are not incentivized to reuse those APIs. Thus, this supports the assertion that modularity provides the necessary but insufficient conditions for option creation.

5.4 Boundary Governance and Velocity Research Findings

Preliminary findings of boundary governance (boundary regulation): Only Gamma exhibited a mature M3 that maintained the three-tiered (Internal/Partner/Public) governance structure to differentiate boundaries for each tier. Twenty-three percent of the application portfolio was generated by partner-developed applications (Tier B: Consistent with partner portal analytics); both Alpha and Epsilon had some evidence of developing boundary governance (nascent partner programs). There is limited evidence of boundary regulation outside of Gamma (See Table 4), which is an empirical limitation requiring additional research to investigate further.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Therefore, we recommend that M3 be viewed as a contingent extension of the core framework until there are empirical studies demonstrating that boundary regulation is necessary for all organizations with mature external API programs.

Velocity Calibration: Alpha used a very strict semantic versioning policy with a 90-day deprecation window; Epsilon supplemented their versioning policy with a DX quality gate that prevented APIs without documentation from registering on the marketplace. Gamma's velocity was high (2.1 major version releases per year per API), and this high rate of release decreased the combinatorial option value of the system. This illustrates the tension between M2-M4 (M2 = Speed; M3=Versioning; M4= Reuse): "We optimized for speed and paid for it in reuse."

5.5 Developer Experience as Empirical Moderator

The cross-case comparison showed that organizations with mature DX (Alpha, Gamma) had better results in their use of the mechanisms than organizations with similar mechanism maturity but less mature DX. Zeta had mature M1 but developing DX; the developers used "shadow integration" to bypass the governed APIs since they were inadequately documented and did not provide adequate SDK support. The mature M2 for Epsilon was limited by its developing DX: The average time it took to onboard was 4.2 days compared to .5 days for Gamma; informants attributed this to the quality of the documentation and availability of a sandbox.

5.6 Deviant and Negative Case Analysis

The theoretical model was enhanced by a deviant/negative case study on three companies (Beta, Delta, and Eta), whose governance maturity scores were the lowest, and/or whose progress toward governance maturity was the least consistent with expectations. The company that exhibited the least mature governance across all the mechanisms, and was at the earliest stage of developing a digital platform (DX), was Delta. Three factors explained why Delta was the worst performer in terms of governance. Firstly, Delta simultaneously implemented all four governance mechanisms as part of a top-down directive, instead of incrementally implementing them one after another. This resulted in what its CTO referred to as "governance fatigue" and the perception among teams that governance represented bureaucratic overhead. Secondly, there were no full-time employees assigned to the role of managing APIs for Delta; responsibilities for API management were scattered across different application teams that had conflicting priorities, and therefore produced inconsistent levels of enforcement. Thirdly, due to the regulatory environment in which Delta operated, it was forced to devote a significant amount of resources to compliance-based governance requirements, thereby diverting resources away from investments that would have contributed to innovation and the development of a digital platform.

Beta, a medium-sized financial service provider, took a different path than Alpha. Although Beta had been using an API-first approach for three years and had a federated model with centralized standards, it still had very little development going on in all but one area. Beta's primary constraint was organizationally-based as the federated model distributed authority across the various business units, each with its own API catalog (with incompatible standards), which undermined the ability to create combinations of options that could be utilized together. Beta's API program manager stated: "Federation without interoperable standards produced individual optimizations for the local (business unit) at the expense of overall portfolio coherency."

Eta represents the study's only full negative case: an organization that intentionally dismantled its formal governance structure. Following a CTO transition, the new leadership concluded that governance was impeding feature delivery velocity, and over an 18-month period systematically unwound its governance structures. This included making design review optional, de-prioritizing catalog maintenance, and moving from a strict version policy to an advisory version policy. As expected in the short term, Eta saw positive results with regard to the company's ability to deliver new features; however, this continued for only 6-8 months before the company experienced severe consequences as a result of the lack of governance. These consequences included significant integration issues throughout the company, duplication of API development across the different departments of the company (the number of APIs available to the company portfolio increased from 280 to 340, with the lead architect estimating that there was approximately 40% functional duplication of the new APIs), and a significant increase in the number of cross team coordination incidents documented in the companies incident management system (Tier A). Eta's CTO said afterwards "I know now that when we took away what we called 'unnecessary' bureaucracy we were really taking away the glue that held all of our teams together, which allowed them to build on top of one another's work" and at the time of the study, Eta had just begun to rebuild its governance because it realized that without a governing structure, the negative effects of an un-governed portfolio grow exponentially as scale grows. This is a clear example of a counterfactual to demonstrate how removing governance led to the exact problems that the proposed governance framework stated would be negatively

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

impacted by a lack of governance, and demonstrates both theoretical plausibility and acknowledges that, at the same time, there was a transition of leadership and a change of strategy.

These deviant cases demonstrate three boundary conditions for governance maturity. Firstly, it is apparent that governance maturity will involve developing capabilities incrementally in order to be implemented comprehensively simultaneously; this suggests an ordering of M1, M2, M4, and M3 as necessary dependencies for governance development. Secondly, the type of governance model used (federated, centralized, or hybrid) impacts the performance of mechanisms to provide a coherent portfolio across units, where federated models will require the explicit coordination of activities at a cross-unit level to be effective. Lastly, the experience of Eta illustrates how governance benefits can be reversed and how the costs of reversing governance will compound over time.

Boundary Conditions and Applicability

The boundaries for applicability of the framework appear to be: (a) The size of the API portfolio is large enough and heterogeneous enough to require management of the portfolio (>~200+ actively used production APIs, i.e., APIs with recurring usage by >~2 independently operating teams within the last 90 days); (b) There are many different innovation projects that share some common API dependencies, therefore there are portfolio level interdependencies; (c) Strategic goals will be achieved by building composite digital products, not monolithic ones. At or below these thresholds, the costs associated with governing an API portfolio using formal governance methods may be greater than the potential benefits, as demonstrated by Delta's experience with investing in governance before its portfolio was complex enough to justify those investments. These are only provisional hypotheses based on limited data, and should be tested and validated against the experiences of organizations of all sizes and in all industry sectors.

Proposition	Mechanism	Supporting Cases (evidence tier)	Strength	Status
P1	M1 Modularity	Alpha (A), Gamma (A), Zeta (B), Delta (C), Eta (B)	**Strong**	Supported
P2	M2 Options	Gamma (A), Alpha (A), Epsilon (B), Zeta (B)	**Strong**	Supported
P3	M3 Boundaries	Gamma (B); limited corroboration	**Thin**	Preliminary
P4	M4 Velocity	Alpha (A), Epsilon (B), Gamma (B, tension)	**Moderate**	Supported
P5	DX moderator	Zeta (B), Epsilon (B), Alpha vs. Delta	**Moderate**	Supported

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

P6	Inter-mech.	Gamma (B, M4–M2 tension); limited	**Thin**	Exploratory

Table 4. Proposition evidence strength assessment. Tier A = corroborated + temporally ordered; Tier B = corroborated, temporally ambiguous; Tier C = interview estimate only

Table 4b. Proposition–Evidence Alignment: P1–P6 Mapped to Cases, Mechanisms, and Portfolio Outcomes

Prop.	Proposition Label	Proposition Statement & Proximal Indicators	Mechanism(s)	Supporting Cases — Role & Key Empirical Evidence	Outcome Addressed (Proximal / Distal)	Outcome Type	Evidence Strength & Tier
P1	**Modularity Enforcement → Initiative Independence**	Enforcement of modularity in a system is associated with a decrease in inter-team coupling and cross-initiative coordination shocks, enabling greater initiative independence. *Proximal indicators: (a) few cross-team change approvals per release; (b) low consumer breakage rates; (c) high deprecation compliance; (d) low mean time to remediate*	**M1** (Architectural Modularity Enforcement)	**Alpha** *(Tier A)* [Primary] −34% cross-initiative defects (Jira, 12-month pre/post gate) **Gamma** *(Tier A)* [Corroborating] Mandatory design reviews; schema governance **Zeta** *(Tier B)* [Corroborating] Mature M1 despite weak DX (shadow integrations suppressed P1 gains) **Delta** *(Tier A)* [Negative] Premature M2 without M1 → coupling failures; validates M1 prerequisite	**Proximal:** Integration defect rate; cross-team change approvals; deprecation compliance rate **Distal:** *Not directly tested (cross-sectional design precludes attribution)*	**PROXIMAL**	**STRONG** (5/7 cases; Tier A documentary evidence in 3)

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Prop.	Proposition Label	Proposition Statement & Proximal Indicators	Mechanism(s)	Supporting Cases — Role & Key Empirical Evidence	Outcome Addressed (Proximal / Distal)	Outcome Type	Evidence Strength & Tier
		integration defects.		role **Eta** *(Tier A)* [Counterfactual] M1 rollback → 40% API duplication, surge in cross-team incidents			
P2	**Option Creation → Innovation Throughput**	Combinatorial option creation is associated with a decrease in marginal build costs for new initiatives and an increase in innovation throughput, evidenced by greater API reuse rates and reduced time-to-first-successful-call. *Proximal indicators: (a) share of new projects reusing pre-existing APIs; (b) decreased time-to-first-*	**M2** (Combinatorial Option Creation)	**Gamma** *(Tier A)* [Primary] 87% APIs tracked in marketplace; avg. usage 1.7 → 4.3 APIs/initiative (portal analytics) **Alpha** *(Tier A)* [Primary] 41% new initiatives using only pre-built components (up from 12%); internal portfolio tracking **Epsilon** *(Tier B)* [Corroborating] Reuse-based resource priority system; increased reuse adoption	**Proximal:** API reuse rate; time-to-first-call; APIs used per new initiative; marginal build cost per initiative **Distal:** *Innovation throughput (reported by informants; not independently measured)*	**PROXIMAL** (primarily) + DISTAL (informant-reported)	**STRONG** (4/7 cases; Tier A documentary evidence in 2)

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Prop.	Proposition Label	Proposition Statement & Proximal Indicators	Mechanism(s)	Supporting Cases — Role & Key Empirical Evidence	Outcome Addressed (Proximal / Distal)	Outcome Type	Evidence Strength & Tier
		successful-call; (c) expansion rate of new project scope from reuse.		**Zeta** *(Tier C)* [Absence] DX friction suppressed reuse uptake despite M2 investment — separates M2 design from DX moderation			
P3	**Boundary Regulation → Controlled Scope Expansion (preliminary)**	Boundary regulation is associated with a higher number of complementary participants at a lower risk-adjusted cost, enabling controlled scope expansion through external or autonomous-internal ecosystems. *Proximal indicators: (a) faster tier-based partner onboarding; (b) lower incident rate per active key; (c) consistent SLA and*	**M3** (Ecosystem Boundary Regulation)	**Gamma** *(Tier B)* [Primary (sole)] 23% of app portfolio from partner-developed apps (partner portal analytics; temporally ambiguous) **Alpha** *(Tier C)* [Nascent] Developing partner program; insufficient maturity to test P3 **Epsilon** *(Tier C)* [Nascent] Some boundary governance evidence; not sufficient for P3 testing	**Proximal:** Partner onboarding time; incident rate per active key; % portfolio from complementors **Distal:** *Scope expansion through complementors (reported by Gamma; not independently measured)*	**PROXIMAL** (limited) + DISTAL (informant-reported only)	**WEAK** (1/7 cases; Tier B only; preliminary — requires targeted future study)

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Prop.	Proposition Label	Proposition Statement & Proximal Indicators	Mechanism(s)	Supporting Cases — Role & Key Empirical Evidence	Outcome Addressed (Proximal / Distal)	Outcome Type	Evidence Strength & Tier
		predictable support demand. Explicitly flagged preliminary — limited evidence base.					
P4	**Velocity Calibration → Time-to-Market with Stability**	More effective velocity calibration is associated with faster and more predictable time-to-market while maintaining code compatibility and reuse. *Proximal indicators: (a) higher release frequency with stable change failure rate; (b) lower upgrade latency; (c) controlled version sprawl.*	**M4** (Innovation Velocity Calibration)	**Alpha** *(Tier A)* [Primary] Cadence governance linked to stable reuse; faster delivery reported **Epsilon** *(Tier B)* [Corroborating] Velocity policy reduced version sprawl; improved predictability **Gamma** *(Tier B)* [Tension case] M2–M4 tension: aggressive versioning temporarily destabilized reuse (CTO interview); resolved via deprecation discipline **Eta** *(Tier A)* [Counterfact	**Proximal:** Release frequency; change failure rate; upgrade latency; version sprawl metric **Distal:** *Time-to-market improvement (informant-reported; cross-sectional design limits causal attribution)*	**PROXIMAL** (primarily) + DISTAL (informant-reported)	**MODERATE–** STRONG (4/7 cases; Tier A in 2; mixed evidence from Gamma)

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Prop.	Proposition Label	Proposition Statement & Proximal Indicators	Mechanism(s)	Supporting Cases — Role & Key Empirical Evidence	Outcome Addressed (Proximal / Distal)	Outcome Type	Evidence Strength & Tier
				ual] Removal of M4 → uncontrolled version proliferation; validates P4 via negation			
P5	**DX as Cross-Cutting Moderator of P1–P4**	Developer experience (DX) positively moderates P1–P4: good DX reduces the marginal cost of adoption and compliance, amplifying mechanism effectiveness; poor DX suppresses effectiveness regardless of governance design maturity. *Moderation indicators: (a) documentation quality; (b) SDK completeness; (c) feedback loop responsiveness; (d) onboarding time; (e) shadow-*	**DX** (Cross-cutting Moderator) Operates across M1, M2, M3, M4	**Alpha** *(Tier A)* [Amplification] High DX maturity → higher mechanism uptake; reduced shadow integrations **Gamma** *(Tier A)* [Amplification] High DX; marketplace UX investment directly increased M2 option usage **Zeta** *(Tier B)* [Suppression] Mature M1 but poor DX → developers bypassed governance ("shadow integration"); P1 gains partially offset	**Proximal:** Adoption rate per mechanism; shadow-integration incidence; onboarding time; voluntary compliance rate **Distal:** *Governance effectiveness at portfolio level (moderated by DX; not independently measurable in cross-section)*	**PROXIMAL** (moderation effect only)	**MODERATE** (4/7 cases; Tier A in 2; cross-sectional design limits isolation of moderation)

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Prop.	Proposition Label	Proposition Statement & Proximal Indicators	Mechanism(s)	Supporting Cases — Role & Key Empirical Evidence	Outcome Addressed (Proximal / Distal)	Outcome Type	Evidence Strength & Tier
		integration rate as inverse proxy.		**Epsilon** *(Tier B)* [Suppression] Onboarding friction suppressed M2 reuse uptake despite formal governance; DX isolated from M2 design			
P6	**Inter-Mechanism Tension**	Option creation (M2) and velocity calibration (M4) can negatively impact modularity enforcement (M1) and boundary regulation (M3) where governance alignment is weak. Good governance alignment reduces inter-mechanism tension. *Tension indicators: (a) negative correlation between M4 release frequency and M2 API stability; (b) consumer*	**M1 × M2 × M4** (and M3 when activated) Governed by Governance Alignment	**Gamma** *(Tier B)* [Primary] CTO interview: aggressive M4 versioning temporarily destabilized M2 reuse; resolved via deprecation discipline (Tier B — temporally ambiguous) **Delta** *(Tier A)* [Corroborating] Weak M1 + premature M2 → coupling failures; documents M1–M2 prerequisite tension **Eta** *(Tier A)* [Extreme case]	**Proximal:** M4 release frequency vs. M2 API stability correlation; breakage rate under high M4 velocity; time-to-first-call variance **Distal:** *Portfolio coherence and coordination cost at scale (Eta counterfactual; informant-reported)*	**PROXIMAL** + DISTAL (Eta counterfactual provides strongest distal evidence)	**MODERATE** (4/7 cases; Tier A in 3 for tension presence; alignment resolution less well evidenced)

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Prop.	Proposition Label	Proposition Statement & Proximal Indicators	Mechanism(s)	Supporting Cases — Role & Key Empirical Evidence	Outcome Addressed (Proximal / Distal)	Outcome Type	Evidence Strength & Tier
		breakage increase from accelerated M4 deprecation; (c) variance in time-to-first-call when versioning velocity exceeds migration framework capacity.		M4 removed entirely → exponential M1 failures at scale; extreme version of M4–M1 tension (Tier A documentary) **Alpha** *(Tier A)* [Resolution case] Mature governance alignment resolved M2–M4 tension; provides evidence that alignment moderates tension			

Note. Evidence tiers follow Section 4.5: Tier A = documentary evidence corroborated and temporally ordered; Tier B = corroborated but temporally ambiguous; Tier C = interview estimate only. "Proximal" outcomes are directly observable portfolio metrics (defect rates, reuse rates, onboarding time, version sprawl). "Distal" outcomes are portfolio-level innovation performance indicators (throughput, time-to-market) reported by informants and not independently measured; the cross-sectional study design does not support causal attribution to distal outcomes. Case roles: Primary = principal evidence source for the proposition; Corroborating = independent confirming evidence; Absence = proposition mechanism absent, enabling contrast; Negative/Counterfactual = mechanism removed or failed, providing strongest distal validation via negation. Row shading: blue = core mechanism propositions (P1, P2, P4); amber = contingent/preliminary (P3); green = moderator (P5); lavender = inter-mechanism tension (P6).

6. DISCUSSION

6.1 Theoretical Contribution

This research provides four theoretical contributions to innovation management theory. First, this study identifies API portfolio governance as an example of how dynamic capabilities theory can be instantiated in a specific area of innovation management, and integrates lifecycle decision-making authority, enforcement procedures, and portfolio economics (option values, reuse incentives, cost-of-change allocations) into a single governance model that relates capability platform architecture to outcomes for innovation portfolios.

The second part of this study expands on the application of dynamic capabilities theory (DCT) to the governance of internal platforms by illustrating how these mechanisms can interact with each other to create

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

emergent portfolio effects. The four mechanisms utilized in this study, based on Teece's [5] triad of sensing through modularity enforcement (M1), which monitors structural integrity and consumption signals across the portfolio; seizing through combinatorial option creation (M2), which exploits reusable modules to lower marginal innovation costs; and reconfiguring through innovation velocity calibration (M4), which adjusts release cadence and compatibility standards in response to changed opportunity and risk profiles. Ecosystem boundary regulation (M3) operates as a contingent extension that, when activated by external-exposure scope conditions (S1–S4), amplifies demands on all three of the core sensing/seizing/reconfiguring mechanisms.

Table 2b. Mechanism-to-Teece Triad Mapping: M1–M4 Alignment, Status, and Empirical Support.

Mech.	Governance Mechanism	Teece Triad Alignment	DCT Function at Platform Layer	Framework Status	Activation Condition	Theorized Proximal Outcome	Empirical Support (Cases)
M1	**Architectural Modularity Enforcement**	**SENSING**	Monitors structural integrity, contract stability, and cross-team coupling signals across the portfolio	**CORE**	Always active; prerequisite for M2 and M4	Greater initiative independence; reduced integration defect rate (P1)	Alpha (Tier A: −34% defects), Gamma, Zeta; Delta (negative — premature M2 without M1). 6/7 cases.
M2	**Combinatorial Option Creation**	**SEIZING**	Exploits modular capabilities via marketplace curation and reuse incentives to reduce marginal innovation cost	**CORE**	Always active; requires M1 as precondition	Increased innovation throughput; lower build-cost per initiative (P2)	Gamma (Tier A: 87% APIs tracked; 1.7→4.3 APIs/initiative), Epsilon, Alpha (Tier A: 41% reuse); Zeta (absence case). 6/7 cases.
M3	**Ecosystem Boundary Regulation**	**SENSING / SEIZING / RECONFIGURING (amplifier)**	Amplifies governance demands on M1, M2, and M4 when external exposure	**CONTINGENT**	Scope conditions S1–S4: external API exposure, high complementor	Controlled scope expansion via complementors; risk-adjusted partner	Gamma only (Tier B: 23% apps from partners). Nascent evidence in Alpha, Epsilon.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Mech.	Governance Mechanism	Teece Triad Alignment	DCT Function at Platform Layer	Framework Status	Activation Condition	Theorized Proximal Outcome	Empirical Support (Cases)
			creates dependencies outside organizational control		autonomy, regulatory/ privacy constraints, or monetization risk	growth (P3 — preliminary)	Single-case support — targeted future validation required.
M4	**Innovation Velocity Calibration**	**RECONFIGURING**	Reconfigures the capability platform by adjusting release cadence, compatibility standards, and deprecation timelines in response to changed risk/opportunity profiles	**CORE**	Always active; stability demands intensify when M3 is activated	Faster, more predictable time-to-market with maintained reuse stability; optimal exploration–exploitation balance (P4)	Alpha (Tier A), Epsilon; Gamma (M2–M4 trade-off case). Eta negative case: rollback → 280→340 APIs, +40% duplication (Tier A). 6/7 cases.
DX *(Moderator)*	***Developer Experience*** *(Cross-cutting Moderator)*	*MODULATES ALL THREE (adoption-cost dimension)*	Reduces the marginal cost of developer adoption and governance compliance, amplifying or dampening the effectiveness of	**CORE MODERATOR**	Always active across all mechanisms; not a mechanism itself	High DX amplifies P1–P4 effects; low DX suppresses all mechanisms regardless of governance design maturity (P5)	Alpha, Gamma (amplification — high DX); Zeta (shadow integration), Epsilon (onboarding friction — suppression). All 7 cases

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Mech.	Governance Mechanism	Teece Triad Alignment	DCT Function at Platform Layer	Framework Status	Activation Condition	Theorized Proximal Outcome	Empirical Support (Cases)
			M1–M4				show measurable DX effect.

Note. Shading key: blue (M1, M2, M4) = core validated mechanisms (present in ≥5/7 cases); amber (M3) = contingent extension (1/7 cases); green (DX) = cross-cutting moderator. "Core" status indicates that the mechanism was empirically associated with governance outcomes across at least 5 of 7 cases. "Contingent" indicates the mechanism was salient in only one case (Gamma) and requires targeted future validation before elevation to co-equal status. P-numbers correspond to propositions in Section 3.8. Tier designations (A/B/C) follow the evidence quality framework in Section 4.5: Tier A = documentary evidence corroborated and temporally ordered; Tier B = corroborated but temporally ambiguous; Tier C = interview estimate only.

A significant feature of this study is that the inter-mechanism dynamics demonstrate that the efficacy of governance mechanisms is a function of the configuration of those mechanisms as opposed to their individual maturity levels. This addresses the call for more detail in DCT [32] and introduces the idea of governance alignment as a theoretical framework to explain why organizations that have the same level of individual mechanism maturity will result in different portfolio outcomes.

Third, the identification of DX as a cross-cutting governance moderator is a theoretical contribution that connects the internal developer platform literature with the literature on innovation governance. In addition to being studied as a software engineering issue [35, 37], the role of DX as a moderating factor for innovation governance has not been theoretically addressed. Therefore, while the empirical findings indicate that DX either enhances or weakens each mechanism (i.e., Zeta's "shadow integration" and Epsilon's onboarding friction), governance research needs to consider the adoption dimensions, not only design.

Fourth, the findings disrupt the notion that portfolio governance is mostly project-level[14]. All the mechanisms identified are at a capability platform layer below the project level but above the Infrastructure Management Layer. This intermediate layer influences portfolio performance, yet this layer has not been conceptualized within previous theoretical frameworks. The distinct contribution to the construct is that it is located at the crossroads of Architecture Governance, Product Management, and Portfolio Strategy - and is thereby distinct from COBIT's enterprise-wide principles[34]; TOGAF's architectural development governance[38] and emerging IDP literature[35, 36, 39].

6.2 Practical Implications

For professionals, we see 4 implications of our results. Organizations need to have a dedicated API portfolio governance role (which includes both API product management from a technical perspective and strategic portfolio management). The Maturity Framework (Appendix A) can serve as a diagnostic tool for professionals to identify areas where the organization's governance has holes in it and prioritize which investments to make with it. We do not intend for the Maturity Framework to be used as a psychometrically valid measure, but rather as a structured way of thinking about what governance elements are missing from an organization's portfolio. Economic governance mechanisms, like reuse credit systems or internal chargeback models, should receive at least the same level of consideration as technical governance mechanisms because they have been shown to increase the speed of products through an organization's portfolio. Finally, governance related to digital transformation should be viewed as one of the top governance priorities of an organization. Our data shows that the maturity of an organization's digital transformation will moderate the returns of all other governance

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

investments. If an organization ignores its digital transformation governance, then it may build governance structures that are technically correct but effectively useless.

The data from our cross-case study indicates a typical sequence of development in an organization's level of maturity in governance (see Figure 2). An organization will develop the ability to enforce modularity first as a way to protect itself, and then it develops its ability to create combinatorial options based on reuse. The latter is developed as the benefits of reuse become apparent. The development of velocity calibration and boundary regulation is typically a later-stage ability and requires a foundation of internal governance. The analysis of the deviant case shows that the sequencing of these governance abilities reflects a necessary dependency structure rather than simply being a common pattern of development; Delta's attempt to implement a comprehensive governance system at the same time resulted in governance fatigue instead of increasing governance maturity. In addition, we found that governance alignment (the extent to which the overall configuration of the governance mechanisms coheres) moderated the degree of maturity that can be achieved in governance; while Beta had sufficient levels of maturity in each of the various governance mechanisms (a "federated" model), it did not have the necessary cross-unit coordination to achieve coherence at the portfolio level. Thus, this knowledge regarding the sequencing and alignment of governance mechanisms provides organizations with a basis to make investment decisions and avoid the temptation to invest in all mechanisms at once.

Stage	Mechanism	Success Requirements	Failure Mode
Stage 1	**M1 Modularity Enforcement**	Standards + automated gates + fast feedback	Reviews become bottlenecks; teams bypass standards
Stage 2	**M2 Option Creation**	Discovery + self-service onboarding + reuse incentives	Good APIs, low reuse; shadow integrations
Stage 3	**M4 Velocity Calibration**	Versioning + deprecation + migration tooling	Breaking changes or frozen releases; consumer debt
Stage 4*	**M3 Boundary Regulation**	Tiers + contracts + policy-as-code (when S1–S4 apply)	Over-restrictive gates; inconsistent tiers

Figure 2. Typical governance sequencing pathway with dependencies and failure modes. *M3 conditional on scope conditions S1–S4.

6.3 Feedback Loops and Path Dependencies

While the propositions are presented as one-way associations for analytical purposes, it is highly probable that the causal relationships contain feedback loops that result in path dependency. There are three feedback mechanisms proposed. First, a governance-legitimacy loop: The governance investments made at the beginning of the project that lead to measurable portfolio performance (i.e., lower defect rates, faster onboarding), create organizational legitimacy and provide additional justification to invest more in governance activities, thereby creating a positive feedback loop. Conversely, if the governance does not show measurable performance results early on (similar to what happened to Delta Airlines), then there will be a negative feedback loop resulting in loss of legitimacy and reduced resources. Second, a reuse-investment loop: As the number of combinatorial options created increases, so too does the amount of reuse generated by production teams. This increased usage provides greater consumption signals that justify continued investment in reusable interfaces for the consuming team(s). Consequently, the reuse loop indicates that any reuse barriers experienced in the early stages of an Option Creation process may serve as a critical mass barrier below which no new options can be created. Third, a DX-compliance loop: Better Developer Experience (DX) quality leads to greater voluntary compliance with governance standards (and therefore lower costs to enforce those standards), thus freeing up more resources to improve DX. Conversely, poorer DX quality results in developers finding ways to circumvent governance standards (which increases the cost to enforce those standards) and subsequently reduces resources available to continue improving DX.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

The nature of feedback within these governance dynamics implies that the trajectory of a firm's governance maturity is to some degree path dependent; i.e., an organization's initial choices regarding the sequence in which it introduces governance functions (as implied by the staging logic in Figure 2) will disproportionately impact its long-term ability to govern effectively. As such, although this study uses a cross-sectional design and therefore cannot provide empirical evidence of the feedback loops inherent in the virtuous cycle of governance, Eta's experience of rolling back their governance has provided a suggestive example of how the removal of governance can compound a decline consistent with the disruption of virtuous cycles. Future longitudinal studies would be well-positioned to test for both threshold effects and hysteresis in the development of a firm's governance maturity.

6.4 Limitations and Future Research

Limitations of the Sample. This study only included six of seven cases drawn from large, high-technology firms (5,000–80,000 employees), and therefore, there was a potential for selection bias based on organization size and technological sophistication, and thus, the ability to implement formalized API governance. Although the inclusion of Eta as a negative case addresses some issues of what might have been (counterfactual evidence), this study lacked cases of organizations that had never formally implemented an API governance program. Thus, the study could not assess how well the theoretical model applied to smaller firms (small and medium enterprises), whose informal processes and resource limitations may lead to very different dynamic effects than those studied. Furthermore, the study could not evaluate whether organizations in relatively less mature API use sectors (e.g., manufacturing, agriculture, public administration) would experience similar forms of API governance.

Cross-Sectional Design: The cross-sectional approach will capture governance structures at one time point and thus precludes making causal statements. Therefore, associations found should be viewed as theoretical patterns of association as opposed to well-established causal relationships. It is also possible for reverse causality to occur; firms that are performing well (in terms of portfolio outcomes) could possibly put more resources into establishing good governance structures. A longitudinal study examining the temporal relationship between implementing governance and the subsequent portfolio outcome changes would provide a much stronger basis for causal statements.

Assessment of Maturity: The use of validated quantitative scales (e.g., Likert-type) for maturity ratings has not been implemented to date. However, as a first step towards developing such scales, an explicit rubric was developed, and two independent coders applied this rubric to all participants' responses. Initial agreement among these two coders was 79%. After a subsequent discussion to reach a consensus, the agreement reached was 91%. These ratings are examples of a "diagnostic judgment" rather than a "psychometric measurement." Future studies should develop valid quantitative scales for assessing maturity based on a formative index methodology, and should examine predictive validity in longitudinal study designs by investigating whether changes in mechanism maturity occur before changes in proximal portfolio indicators.

Evidence Base for M3 (Boundary Regulation): There is very little data outside of the Gamma study that supports boundary regulation. P3 is also listed as preliminary. We think M3 can be viewed as an optional mechanism until there are targeted studies in companies that have mature external API programs. Organizations with established partner or public API programs should be over-sampled in future case studies so as to create the necessary evidence base for M3.

Confounding and alternative explanations: In addition to organizational culture and leadership commitment to IT, other factors that may individually affect governance and its outcomes include IT budgets and/or current transformations within the organization in which governance is being implemented. The methodology of this study to address potential confounds is documented in Appendix B; for each of the cases, the governance interventions implemented were identified, along with those concurrent changes that might represent alternative explanations, and how the data were used to determine the extent to which governance influenced outcomes versus the degree to which confounds impacted outcomes. This methodology does not involve statistically controlling for confounds but rather provides a way for researchers to assess the likelihood of an association existing between governance and outcomes. As such, future studies on governance can use methodologies that allow for the statistical control of confounders (i.e., structural equation modeling/quasi-experiments – e.g., staggered implementation of policies/platform reorganization).

Evolving context: The evolving environment of this research framework was established over a time frame (2015-2025), where there were many AI-Powered APIs that started to dominate the marketplace. The governance requirements will likely continue to evolve as agentic AI systems continue to create new forms of consumption [33]. Using ethnography to observe the governance in practice would complement the interview

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

process. Finally, the reference list utilizes both practitioner-based publications and academic-based publications; Practitioner-based publications are utilized only for providing contextual operationalization, and all of the propositions presented are supported by academic-based theory.

7. CONCLUSION

In this paper, we have argued that API portfolio governance is an underexplored yet strategically important dynamic capability that links a firm's platform architecture to its innovation portfolio performance. Utilizing a multiple phase qualitative methodology including systematic literature review and a 7-Case Empirical Study (one Case was a Negative Case), we identified four governance mechanisms that firms can use to convert API level governance into portfolio level outcomes through the architectural modularity enforcement, combinatorial option creation, the boundary regulation of their ecosystem (dependent upon their externalization strategy) and the calibration of innovation velocity. The four governance mechanisms were influenced by the cross-cutting effect of developer experience governance.

The results demonstrate that advanced levels of maturity in API governance are directly related to greater independence for initiatives (the ability to work independently), greater acceleration of throughputs (processes), greater scope of services offered by an organization via APIs (scope), and better overall balance between exploring new possibilities (exploration) and exploiting existing ones (exploitation). However, these outcomes will need to be confirmed as causal relationships over time through future research involving longitudinal studies [41]. Additionally, these outcomes appear to require careful design and implementation of governance processes that align both the technical management of an organization's API products with strategic portfolio-level oversight of those products. Further, there needs to be particular consideration given to how governance mechanisms should be sequenced and aligned, and what DX quality is necessary to support the successful implementation of those governance mechanisms. The diagnostic framework used to analyze and compare cases offers practitioners a systematic way to assess and develop capabilities in this area.

For innovation management research, the primary implications of this are that for an organization to have effective portfolio governance, it must include the underlying architectural structure on which its digital innovations are built. The construct of API portfolio governance represents a unique analytical area that falls in-between the governance of Enterprise Information Technology (IT) portfolios and the management of portfolios at the level of projects; as such, it provides a theoretical basis for how architectural governance results in innovation performance. As organizations continue to pursue "API-first" strategies and as the API economy continues to mature, the governance of these digital building blocks will become increasingly important in determining whether organizations can sustainably achieve innovation performance or if they will experience architectural fragmentation and/or capability sprawl.

8. ACKNOWLEDGMENTS

The authors thank the participating organizations and the interview respondents who were very cooperative in this study.

9. APPENDIX

9.1 APPENDIX A: DIAGNOSTIC RUBRIC

In this appendix, the final version of the qualitative evaluation scoring rubric used to determine developmental stages (nascent, developing, mature) for each mechanism (M1-M4) and for DX is documented. This document is the culmination of previous summary descriptions made in the body of this report and is to be referenced by all parties interested in knowing the definitive scoring anchors. While the rubric is provided with the intent of providing transparency and assisting in replicating the findings contained within this report, it has not been validated or shown to have any reliability characteristics of a psychometrically valid test.

Beginning with the scoring process for each case, coders gathered all relevant excerpted evidence (e.g., interview quotes, documentary artifacts) and assigned a maturity level to each dimension. A mechanism was considered "mature" if at least three of its four dimensions met the defined mature criteria and no dimension was developing, nascent, or otherwise. A mechanism could have up to two dimensions that were deemed to be nascent, provided that boundary regulation (dimension M3) was either present and in use or absent (and thus not used) by virtue of the organization's strategic decision to forego an external API strategy; this absence was viewed as a strategically intentional decision and was analyzed in conjunction with both the broader industry and regulatory environment. The four scoring dimensions were: (1) the presence and breadth of the existence of governance-related artifacts (standards, gates, tiering policies), (2) the method of enforcement (manual review

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

versus automated control mechanisms), (3) the ability to view a portfolio (registry completeness, usage analytics) and provide telemetry (usage data), and (4) evidence of the adoption and/or compliance with standards (use reuse patterns, deprecation compliance). Each dimension for each case was coded independently by two coders using a three-level coding scheme. Disagreements were resolved through consensus meetings held after a pilot calibration session for two cases.

Dimension	Nascent (ad hoc)	Developing (partial)	Mature (institutionalized)
M1 Modularity	Local standards; limited automated checks; exceptions unmanaged	Portfolio standards exist; partial CI/CD enforcement; review boards for high-risk APIs	Portfolio-wide contract standards + automated conformance gates; bounded exceptions with expiry; consumer mapping used in approvals
M2 Options	APIs discoverable only via tribal knowledge; ownership unclear; reuse not incentivized	Catalog exists for key domains; some ownership metadata; reuse encouraged, case-by-case	Curated marketplace with taxonomy and analytics; clear product ownership; portfolio funding aligns with reuse
M3 Boundaries	Exposure decisions are ad hoc; inconsistent access controls	Tiering model defined; partner onboarding checklists; partial monitoring	Explicit exposure strategy; standardized legal/technical onboarding by tier; automated key management + monitoring
M4 Velocity	Breaking changes frequent; deprecations untracked; version sprawl	Versioning guidance, some deprecation timelines, and releases coordinated in hotspots	Semantic versioning + governed deprecation windows; change calendars + migration playbooks; predictability metrics
DX (moderator)	Docs/SDKs inconsistent; no feedback loop; onboarding slow	Portal/docs standards emerging; basic templates; some telemetry	Internal dev portal + paved paths; documentation/SDK quality gates; continuous feedback loops tied to backlog

Table A1. Maturity scoring rubric with operational anchors

9.2 APPENDIX B. RIVAL EXPLANATIONS

This appendix will explicitly detail the two forms of data used in the scoring of mechanisms and their maturation (i.e., what kinds of data are being used to assess the levels of each mechanism) and identify potential rival explanations for observed trends in outcomes within cases. We want to be transparent about how our analysis was conducted versus providing a statistical model of control. Table B1 will summarize the information that has been documented for each of the case studies regarding: the primary governance interventions during the time frame of the study; the relevant organizational changes that may explain the changes observed in the case; and the specific triangulation strategies employed to isolate governance from the confounding variables.
Evidence from the design review of the application is in the form of a template for use by reviewers as well as an exception log to track when the reviewers have allowed exceptions to the design principles (M1). Evidence that the CI/CD pipeline can automatically test each component of the application includes test configuration for contract testing as well as schema validation using lint tools (M1). The taxonomy of APIs that are available for developers to consume, ownership of those APIs, and how often they are consumed are all pieces of evidence

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

that the API catalog has been populated (M2). Documentation about what permissions developers require to perform various actions against the services offered by the platform, including a list of the partners with whom the company currently works and templates for the contracts that need to be signed before a new partner can onboard, is evidence of the tiered access policy (M3). Documentation related to deprecating features, documenting changes, and tracking whether upgrades have completed successfully across all environments is evidence of the versioning and deprecation policy as well as the change calendar (M4). Documentation that outlines the standards for developing and publishing content, as well as standards for creating and maintaining onboarding workflows for new developers, SDK release notes, and records of the developer feedback loop (DX), is additional evidence of how the organization has established a culture of innovation and experimentation.

Case	Primary Governance Moves	Concurrent Changes (Rivals)	How Addressed
Alpha	Mandatory design review; portfolio-wide deprecation rights; portal analytics	Platform reorg; testing infra upgrades; modernization program	Metrics corroborated via defect dashboard; rivals noted as contributing factors
Beta	Partial standards; limited enforcement; ad hoc incentives	Leadership turnover; budget constraints	Used as deviant case; outcome claims treated as directional
Gamma	Automated contract testing; tiered access; partner onboarding	Cloud migration; identity/access modernization	Partner timelines cross-checked with onboarding docs and analytics
Delta	Simultaneous mechanism rollout via top-down mandate	Heavy compliance workload; early-stage transformation	Counterfactual for low-maturity; governance fatigue triangulated across CTO, architects, portfolio manager
Epsilon	Developing marketplace; modest reuse economics; improving lifecycle policy	Shift to product-based funding; parallel data-platform program	Consumption analytics corroborated adoption; timeline comparison separates effects
Zeta	Mature modularity gates + option-creation structures; conservative velocity	Business-unit consolidation; release-train changes	Upgrade metrics corroborated via dashboards; conservative interpretation
Eta	Governance rollback: reviews optional; catalog deprioritized; versioning relaxed	Leadership transition; strategic pivot to rapid expansion	Incident data corroborated 2.5× coordination increase; rollback timeline confirmed via governance charter

Table B1. Case-level rival explanations and triangulation notes.

9.3 APPENDIX C. MEASUREMENT ANCHORS.

In this appendix, we transform the four conceptual mechanisms (M1-M4) and the moderating role of DX into measurement-based indicators for replicability purposes and for testing longitudinally in the future. The indicators illustrated in the next section are merely examples rather than an empirically tested scale; therefore, researchers will need to adjust the definitions and provide the exact formulas used to compute each indicator.

Guidelines for Reporting: To ensure that there is enough information provided for other researchers to replicate the results from their study(ies), studies utilizing the indicators listed below should include the following: (i) identification of the level of analysis utilized (firm versus business unit portfolio), (ii) indication of the time window(s) used to collect data (and/or to determine the time window(s) used to compute the metrics), (iii) clear

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

distinction between the system metrics supported by objective evidence and those estimated based on informant perceptions, and (iv) documentation of any concurrent changes (e.g., organization-wide reorganization, modernization program) which may be confounding variables that could explain why a firm's innovation performance changed over time. Future research can test and refine the measurement framework developed herein by correlating the scores for the mechanisms with the proximate measures identified in this appendix and with the innovation portfolios' performances through longitudinal, panel, or quasi-experimental designs.

Mechanism	Observable Governance Artifacts	Proximal Metrics	Distal Outcomes
M1 Modularity	Contract standards; automated lint/contract tests; exception logs; consumer mapping	Consumer breakage rate; change failure rate; integration defect MTTR; upgrade latency	Initiative independence; reduced coordination overhead
M2 Options	Catalog/marketplace taxonomy; ownership metadata; reuse funding/chargeback; consumption analytics	Reuse share of new initiatives; time-to-first-call; duplicate capability rate	Innovation throughput; improved cost-to-build
M3 Boundaries (conditional)	Exposure tiering; onboarding checklists; partner contracts/SLAs; key management	Partner onboarding time by tier; abuse/incident rate per key; support tickets/API	Risk-adjusted ecosystem growth
M4 Velocity	Versioning/deprecation policy; change calendar; migration playbooks; readiness gates	Release frequency vs. predictability; deprecation backlog age; version sprawl; upgrade effort	Faster time-to-market with stability
DX (moderator)	Docs/SDK standards; internal dev portal; paved paths; feedback loops linked to backlog	Onboarding time; doc completeness; SDK adoption; developer satisfaction (SPACE)	Higher mechanism uptake; reduced shadow integrations

Table C1. Operationalization and measurement anchors

10. REFERENCES

[1] Postman. 2025. 2025 State of the API Report. Postman, Inc.
[2] De, B. 2023. API Management: An Architect's Guide, 2nd ed. Apress.
[3] Lerch, M. and Spieth, P. 2013. Innovation Project Portfolio Management. IEEE Trans. Eng. Mgmt., 60(1), 18–29.
[4] Teece, D.J., Pisano, G. and Shuen, A. 1997. Dynamic Capabilities and Strategic Management. Strategic Mgmt. J., 18(7), 509–533.
[5] Teece, D.J. 2018. Dynamic Capabilities as (Workable) Management Systems Theory. J. Mgmt. & Org., 24(3), 359–368.
[6] Teece, D.J. 2017. Dynamic Capabilities and (Digital) Platform Lifecycles. In Advances in Strategic Mgmt. 37, 211–225.
[7] Chirumalla, K. et al. 2023. Moving from Servitization to Digital Servitization. J. Business Research, 158, 113668.
[8] Parker, G., Van Alstyne, M. and Choudary, S.P. 2016. Platform Revolution. W.W. Norton.
[9] Gawer, A. and Cusumano, M.A. 2014. Industry Platforms and Ecosystem Innovation. JPIM, 31(3), 417–433.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

[10] Tuominen, S. and Martinsuo, M. 2025. Alternative Approaches to Innovation Project Portfolio Governance. PMJ, 56(2), 143–161.
[11] Jacobson, D. et al. 2024. APIs: A Strategy Guide, 2nd ed. O'Reilly.
[12] Medjaoui, M. et al. 2021. Continuous API Management, 2nd ed. O'Reilly.
[13] Axway. 2024. State of Enterprise API Maturity Report. Axway Inc.
[14] Cooper, R.G. et al. 2001. Portfolio Management for New Product Development. R&D Mgmt., 31(4), 361–380.
[15] Kester, L. et al. 2011. Exploring Portfolio Decision-Making Processes. JPIM, 28(5), 641–661.
[16] Nambisan, S. et al. 2017. Digital Innovation Management. MIS Q., 41(1), 223–238.
[17] Salonen, A. et al. 2024. Portfolio Management of Digital Innovation Initiatives. Proc. SCIS2024.
[18] Baldwin, C.Y. and Clark, K.B. 2000. Design Rules: The Power of Modularity. MIT Press.
[19] Henderson, R.M. and Clark, K.B. 1990. Architectural Innovation. ASQ, 35(1), 9–30.
[20] McGrath, R.G. 1997. A Real Options Logic for Technology Positioning. AMR, 22(4), 974–996.
[21] Henfridsson, O. et al. 2018. Recombination in the Open-Ended Value Landscape of Digital Innovation. I&O, 28(2), 89–100.
[22] Ghazawneh, A. and Henfridsson, O. 2013. Balancing Platform Control and External Contribution. ISJ, 23(2), 173–192.
[23] Eaton, B. et al. 2015. Distributed Tuning of Boundary Resources. MIS Q., 39(1), 217–243.
[24] O'Reilly, C.A. and Tushman, M.L. 2013. Organizational Ambidexterity. AMP, 27(4), 324–338.
[25] Müller, R. et al. 2019. A Framework for Governance of Projects. IJPM, 37(6), 780–797.
[26] Creswell, J.W. and Plano Clark, V.L. 2023. Designing and Conducting Mixed Methods Research, 4th ed. SAGE.
[27] Yin, R.K. 2018. Case Study Research and Applications, 6th ed. SAGE.
[28] Landis, J.R. and Koch, G.G. 1977. The Measurement of Observer Agreement. Biometrics, 33(1), 159–174.
[29] Eisenhardt, K.M. 1989. Building Theories from Case Study Research. AMR, 14(4), 532–550.
[30] [30] Braun, V. and Clarke, V. 2021. Thematic Analysis: A Practical Guide. SAGE.
[31] Cusumano, M.A. et al. 2019. The Business of Platforms. Harper Business.
[32] Schilke, O. et al. 2018. Quo Vadis, Dynamic Capabilities? AMA, 12(1), 390–439.
[33] Postman. 2025. AI Agent Security Concerns in the API Ecosystem. Postman, Inc.
[34] ISACA. 2019. COBIT 2019 Framework: Governance and Management Objectives.
[35] Forsgren, N. et al. 2021. The SPACE of Developer Productivity. ACM Queue, 19(1), 20–48.
[36] Humanitec. 2024. State of Platform Engineering Report.
[37] Greiler, M. et al. 2022. An Actionable Framework for Developer Experience. IEEE TSE, 48(9), 3468–3485.
[38] The Open Group. 2022. TOGAF Standard, 10th ed.
[39] Mandic, D. and Terzic, Z. 2023. Internal Developer Portals as Innovation Enablers. Proc. EuroPLoP 2023.
[40] Storey, M.-A. and Zagalsky, A. 2016. Disrupting Developer Productivity One Bot at a Time. Proc. FSE 2016, 928–931.
[41] Kong Inc. 2024. API Maturity Model. Kong Inc.
[42] SmartBear. 2024. State of API Quality Report.
[43] Tiwana, A. 2014. Platform Ecosystems. Morgan Kaufmann.