Technology Ecosystem Governance

Jonathan Wareham
Paul Fox
Josep Lluís Cano Giner
Technology Ecosystem Governance

Jonathan Wareham  
ESADE Business School, Ramon Llull University

Paul Fox  
La Salle Business Engineering School, Ramon Llull University

Josep Lluís Cano Giner  
ESADE Business School, Ramon Llull University

January 2013

**This working paper is an updated version of: Wareham, J.; Fox, P.; Cano, J. Paradox in Technology Ecosystem Governance. ESADE Working Paper Nº 225, April 2012.**

Abstract

Technology platform strategies offer a novel way to orchestrate a rich portfolio of contributions made by the many independent actors who form an ecosystem of heterogeneous complementors around a stable platform core. This form of organizing has been successfully used in the smartphone, gaming, commercial software, and other industrial sectors. While technology ecosystems require stability and homogeneity to leverage common investments in standard components, they also need variability and heterogeneity to meet evolving market demand. Although the required balance between stability and evolvability in the ecosystem has been addressed conceptually in the literature, we have less understanding of its underlying mechanics or appropriate governance. Through an extensive case study of a business software ecosystem consisting of a major multinational manufacturer of enterprise resource planning (ERP) software at the core, and a heterogeneous system of independent implementation partners and solution developers on the periphery, our research identifies three salient tensions that characterize the ecosystem: standard-variety; control-autonomy; and collective-individual. We then highlight the specific ecosystem governance mechanisms designed to simultaneously manage desirable and undesirable variance across each tension. Paradoxical tensions may manifest as dualisms, where actors are faced with contradictory and disabling ‘either/or’ decisions. Alternatively, they may manifest as dualities, where tensions are framed as complementary and mutually-enabling. We identify conditions where latent, mutually enabling tensions become manifest as salient, disabling tensions. By identifying conditions in which complementary logics are overshadowed by contradictory logics, our study further contributes to the understanding of the dynamics of technology ecosystems, as well as the effective design of technology ecosystem governance that can explicitly embrace paradoxical tensions towards generative outcomes.

Keywords: technology ecosystems, platforms, governance, paradox, tensions.
1.0 Introduction

Technology ecosystem governance has become a focal topic for the numerous technology platforms that organise the complements and services that extend the overall value of a core product. Platform strategies are well known from smart phone platforms, but are also common in gaming consoles, as well as the social media platforms. Beyond consumer markets, platform strategies are frequently used in industrial sectors, where base products in software, manufacturing or scientific machinery nourish a larger community of repair and maintenance organisations that operate as semi-autonomous value-added resellers and service providers (Aguilar 1998, Weisenfeld et al. 2001).

In addition to nourishing a community of 3rd party service providers, platforms can also explicitly encourage innovations that go well beyond the original product portfolio. Via purposefully designed governance mechanisms, they can promote contributions of participants who are not directly employed by the platform core to create complementary products and services to address the needs of a large, globally heterogeneous group of end-users in a manner that would be prohibitively difficult for the core to do alone. This is a case that goes beyond the mere design, development, and distribution of pre-defined products, towards a strategy of purposefully cultivating an ecosystem of complementors for ‘generativity’. Generativity refers to the ability of a self-contained system to create, generate, or produce a new output, structure, or behaviour without any input from the originator of the system (Tilson et al., 2010). It includes both the technology artefacts, as well as the social meanings and behaviours that constitute their larger system of use (Avital and Te’eni, 2009).

However, the reliance on an ecosystem of autonomous partners is not without risk. Where generativity and local agility is desirable, un-controlled creative output is not always positive for the health of the ecosystem (Hagiu, 2010). Stated colloquially, letting a thousand flowers grow can often produce low-quality services and complements, resulting in a negative customer experience, and thereby seriously harming the reputation and economic sustainability of the product platform (Boudreau, 2011). So the design of governance mechanisms for technology
ecosystems is not a trivial task; the challenge is to establish governance mechanisms that appropriately bound participant behaviour without excessively constraining the desired level of generativity. This governance problem has been broadly called the ‘paradox of change’ (Tilson et al. 2010 p. 6) in the literature, referring to the need for technology ecosystems to be stable and evolvable simultaneously. Without evolvability, the platform cannot adequately adjust to changes in customer requirements, market shifts, and technology evolution. Without demonstrated stability, complementors and customers would have little assurance that their substantial financial and human resource investments can yield long-term returns.

Where the stability-evolvability tension has been addressed conceptually in the literature (Yoo et al. 2010, Tilson et al. 2010, Tiwana et al. 2010), the research lacks substantial empirical and theoretical insight into how this tension is manifested, as well as how technology ecosystem governance can address it. Stability-evolvability in technology ecosystems refer to the products and services of the platform. These outputs are produced by complementors who work autonomously to meet client needs, but must simultaneously subsume their actions and outputs to some form of direction-giving (Demsetz 1997) to limit inferior quality and encourage contributions to the public goods of the ecosystem. This suggests that the general stability-evolvability paradox is manifested across a number of underlying dimensions that produce tensions pertinent to technology ecosystems. And as argued in the literature (Boudreau 2010, Boudreau 2011, Hagiu 2010, Messerschmitt and Szyperski 2003), balancing these tensions is one of the main goals of technology ecosystem governance. It is essentially a design problem (Simons 1995), where specific governance levers, if properly implemented, can accommodate these tensions towards a sustainable equilibrium. If improperly implemented, ineffective governance can result in a degenerative evolution of the ecosystem. As existing research offers limited insight into what these underlying tensions are, the first stage of the design of effective ecosystem governance is to identify the most pertinent tensions, and thereafter consider the design of governance mechanisms to effectively accommodate them. Accordingly, we formulate our first research question.
• **RQ 1.** How are the main tensions in technology ecosystems addressed in technology ecosystem governance?

Scholars have broadly defined the term ‘paradox’ in a managerial context as meeting competing demands simultaneously (Cameron 1986, Poole and Ven 1989), implying a need to balance contradictory tensions. However, the literature suggests that there are differences in the manner by which tensions can be framed or leveraged (Smith and Lewis 2011). Conceptually, tensions can manifest as competing trade-offs that present ‘either/or’ decisions in the form of discrete alternatives, as a dualism. Alternatively, tensions can function as complementary and mutually enabling attractors in a holistic system, as a duality (Farjoun, 2010). Given the centrality of tensions in ecosystem governance, our second research question explores how complementary or competing logics become manifest in the ecosystem and how they are governed.

• **RQ 2.** Tensions can manifest themselves as either contradictory or complementary logics. Are contradictory and complementary logics present in technology ecosystems? If so, how are they governed?

One industrial sector that has successfully developed a vibrant ecosystem of 3rd party complementors for generativity is enterprise resource planning (ERP) software. Providers of commercial accounting and manufacturing software suites, such as SAP and Oracle, have long benefited from the expertise of local or regional implementers to make country specific modifications, sector specific add-ons, and company specific customisations that meet the local, distinct needs of their clients. The use of third party partners to do this is largely driven by the extreme heterogeneity that characterises the enterprise resource planning software market. ERP software is designed to be a common suite of financial accounting, manufacturing, and logistics software that is modified to function in a global market of over 100 legal & accounting regimes. Orthogonal to national differences, myriad sector differences require modifications and add-ons to meet the needs of a wide range of entities from manufacturing to service organisations in the private and public sectors. Using one standard software suite to meet the different requirements of, for example, paper processing, health care, financial services, manufacturing, or
education, is a formidable challenge. Consequently, the ERP software manufacturer wisely consigns these localisation tasks to regional partners with far greater expertise in their native markets, functioning as accounting, legal, or sector specialists (Suprateek Sarker et al. 2012, Ceccagnoli et al. 2012).

Accordingly, our analysis studies one such major vendor of ERP software. We perform an extensive analysis of the governance mechanisms developed in a broad ecosystem that includes the software vendor at the core, and a large community of independent implementers that develop and implement complements targeted towards myriad national regions and industrial sectors. Our ERP ecosystem is of interest for several reasons. Firstly, it is one of the few empirical examples of how technology platforms function in a for-profit, business-to-business context (Suprateek Sarker et al. 2012, Ceccagnoli et al. 2012). Secondly, given the severe heterogeneity across customers, complementors, and complements, ERP ecosystems represent an extreme form of technology ecosystem, where the underlying tensions arguably assume their utmost manifestation, and their appropriate management is of greatest potential consequence. As such, the manner in which ERP ecosystem governance accommodates the most salient tensions, embracing them as either contradictory or complementary logics, is likely to be one of the more illustrative and sophisticated exemplars where such effects are discernible.

This empirical context enables us to make to flowing contributions. We identify three specific tensions pertinent to technology ecosystems that are characterised by high levels of heterogeneity: standard-variety; control-autonomy; and collective-individual; and describe their interdependencies. We thereafter highlight the specific ecosystem governance mechanisms designed to simultaneously increase desirable variance and decrease undesirable variance across each tension. We further delineate how paradoxical tensions can be manifested as either complementary or contradictory logics, and describe how these opposing logics can be accommodated by governance mechanisms, exploring their causes, effects, and potential managerial responses to them as a problem of technology ecosystem governance design. These specific insights provide contributions towards understanding the actual mechanics of the stability-evolvability paradox in
technology ecosystems, as well as theoretical and normative insights into how technology ecosystem governance can be designed to accommodate these tensions and lead them effectively towards generative outcomes.

2.0 Conceptual development

This section outlines the main characteristics of technology ecosystems and describes three main tensions that underlie the stability-evolvability paradox. We develop a framework of how these tensions interact. From here, we consider how these tensions can manifest in either contradictory or complementary forms, and consider implications for technology ecosystem governance design.

Technology ecosystems are often described as product platforms defined by core components made by the platform owner and complements made by autonomous companies in the periphery. These ecosystems have two primary characteristics: 1) they should perform an important function within a ‘system of use’ or solve an important technical problem within an industry; and 2) it should be easy to connect to or build upon the core solution in order to expand the system of use and allow new and even unanticipated end-uses. The core firm’s product has important, but limited, value when used alone, but substantially increases in value when used with complements (Gawer and Cusumano, 2002, 2008).

The term ‘ecosystem’ alludes to the biology and ecology disciplines (Folke et al. 2002, Chapin et al. 2004, Holling 1973). The central concept is ‘resilience’, an ability of an ecosystem to maintain its own inertia, yet adapt to exogenous shocks with an innate generative ability to evolve endogenously. Generative ability refers to the capacity of a self-contained system to create, generate, or produce a new output, structure, or behaviour without any input from the originator of the system (Avital and Te’eni 2009, Tilson et al. 2010). For technology ecosystems, this ability to evolve is particularly valuable when consumer patterns are heterogeneous, technologies are fragmented, and overall market trajectories are uncertain (Baldwin & Woodard, 2008; Boudreau & Hagiu, 2009). The platform must simply evolve and generate novel and relevant responses to rapidly changing consumer demands, fast
development clock-speeds and product life-cycles. Yet excessive evolvability without some degree of stability and inertia can render a system fragmented in use, unsustainable financially, and irrelevant in the marketplace. Accordingly, maintaining equilibrium between stability and evolvability is a central challenge to technology ecosystems. By making some components variable, platforms become adaptable to future technological developments, social or business trends, as well as uncertain or unanticipated environmental changes. By keeping other components stable, they encourage the re-use of standardised processes and outputs that benefit the ecosystem as fully amortised investments, leveraging economies of scale, and generating positive externalities and social goods. Hence, the ability to evolve must be simultaneously anchored in the inertia of a stable core.

Yet the management of the stability-evolvability tension is arguably more complex than a simple dichotomous continuum. Towards illuminating this complexity, we posit that the competing goals of stability and evolvability in an ecosystem generally refer to outputs; the products and services that offer value, persuading customers to adopt one specific platform instead of another. Outputs are created by complementors; autonomous actors who channel entrepreneurial instinct to address client needs with novel innovations. Actors are driven by self-interested motives to respond to market needs, yet must simultaneously contribute re-usable knowledge to the greater collective good and cohesion of the ecosystem. Consequently, we identify three distinct yet related dimensions where this stability-evolvability tension can manifest in the ecosystem, and must be embraced by technology ecosystem governance: i) outputs, ii) actors, and ii) identifications. We proceed by delineating each tension, their mechanics, and subsequent implications for technology ecosystem governance.

2.1 Outputs: standard and variety

The economic logic of platforms suggests that a high standardisation of the core enables economies of scale and the amortisation of fixed costs. Yet on the periphery, economies of scope can be realised through the creation of specialised complements and constant experimentation (Baldwin and Clark, 2000), where
complementors respond to the needs of users with a level of speed or specialisation that would otherwise be prohibitively difficult for the core. In doing so, complementors extend a well-established foundation of existing practices, processes and artefacts, to minimise cost and optimise efficacy – while offering the client solutions where it matters. In effect, stability is leveraged to maximise variety.

Variety, however, is not limited to the periphery or complements of the platform. As Baldwin and Woodward (2008) highlight, core components of a system will also need to change over time, if for no other reason than to embrace basic technological advances in underlying technologies such as processing, storage, communication, and power consumption. What remains stable through the evolution of the platform are the interfaces or thin crossing points (Baldwin, 2008) that govern interaction between the layers (Baldwin & Woodard, 2008). These architectural control points (Woodward, 2008) govern the relationships between the core and complements, creating bottlenecks where platform operators can, via property and other legal rights of exclusion, grant or deny actors access to the system (Boudreau & Hagiu, 2009; Jacobides, Knudsen, & Augier, 2006; Rochet & Tirole, 2003).

As such, the interfaces between components are one of the most important levers of mechanism governance to balance a standard – variety tension, being either loosely or tightly coupled. Loosely coupled ecosystems are characterised by very thin interfaces between components, enabling generativity to emerge at different layers of the platform architecture (Yoo et al. 2010, Messerschmitt and Szyperski 2003). Where loose coupling facilitates greater generativity, it risks greater fragmentation, inefficiency, and an inferior user experience (Messerschmitt and Szyperski, 2003). Components become more appropriable, subject to reverse engineering, and complementors may quickly become competitors (Eisenmann et al., 2011). In addition, excessive specialisation, diversity, and random competition can lead to overcrowding, high coordination costs and fragmentation (Boudreau 2010) that can inhibit rates of development and adoption of both the core and complements (Adner, 2006). By contrast, tightly coupled architectures tend to be closely nested or fixed, with closed product-specific interfaces that protect the market position of the product via lock-in and asset specificity. High levels of cohesion and integration with core components can increase an intrinsic protection
from appropriation, as well as a more holistic and cohesive user experience; yet at the same time, stifle innovation and constrain the level of platform evolvability and user adoption (Rochet and Tirole 2003, Parker and Van Alstyne 2005).

Consequently, technology ecosystem governance employs architecture to balance the tension between standard and variety in outputs. These architectural control points are embedded in the variance decreasing or variance increasing ecosystem governance (tightly or loosely coupled) and often work simultaneously across diverse dimensions to achieve an optimal, aggregate effect. Specifically, ecosystem governance must utilize variance reducing mechanisms to ensure effective standards in technological compatibility, high quality in complements and implementations, and the reuse of common and standard modules to leverage economies of scale and network effects. Simultaneously, governance must employ variance increasing mechanisms to generate agile market responses to client requirements by increasing the number and character of complements, driving platform adoption with increased options and longevity through adaptation to exogenous social, economic, and technological evolutions.

2.2 Actors: control and autonomy

The logic of technology ecosystems is based upon the idea that, while it is possible to cultivate generativity within the boundaries of a single firm, generative potential is substantially increased in a looser arrangement of heterogeneous actors who pursue self-interested, innovative activities in a distributed and scalable ecosystem (Busquets et al. 2009, Yoo et al. 2010). As autonomous actors, complementors are attracted to opportunities for economic value, invoking the speed of market mechanisms, a Hayekian response to the here and now, focusing their own portfolio of domain expertise, sector knowledge, and relational capital to create locally relevant solutions. By attracting specialists with different backgrounds and perspectives to autonomously address difficult problems presented by clients, the ecosystem can foster ‘productive friction’ (Hagel and Brown, 2005 p. 100) which increases the possibilities for innovation, learning, and competency building.
However, in letting a thousand flowers grow, unchecked variance in quality can lead to agency costs, where the platform core bears the negative costs of poor quality of the complementor or their applications (Wolter and Veloso, 2008). As an example, Hagiu (2010) found that Atari's inability to control the overcrowding and subsequent downward price pressure of low quality games in their ecosystem was central to the platform's demise. This experience suggests a need to equally distribute the economic wealth through a regime of property rights and legal protection that fosters sustainability for both the ecosystem periphery and core (Iansiti and Levien, 2004). In other words, two hundred well-designed flowers are perhaps more appropriate to the health of the ecosystem than one thousand of a variable quality. This suggests that the focus of control levers are not limited to output quality, but should also address issues of overall complement quantity, the distribution of complementor efforts across heterogeneous market niches, the frequency of complement releases, and other managerial decisions that directly or indirectly influence the quality and character of complement supply.

Accordingly, technology ecosystems must simultaneously realise a form of constrained serendipity (Faraj et al. 2011) by employing variance reducing mechanisms to control poor complement quality (both process and output), ensure professional business conduct, and constrain excess supply that can erode economic value; yet also utilise variance increasing mechanisms to leverage autonomy for innovative responses to client requirements and market evolution, generating a sufficiently large, high-quality portfolio of complements to encourage user adoption.

### 2.3 Identifications: individual and collective

In technology ecosystems, individual complementors work towards their own benefit for financial compensation, career advancement, or other extrinsic motivations. Yet where much generative work is localised with a specific client or application, the ecosystem itself still requires some sense of cohesion and consistency across niche-focused efforts; a larger social good; a canon of best-practice, and re-usable knowledge that further enables the scalable exploitation of common innovations. As
a large system of distributed innovation, governance infrastructures must be developed that embrace disparate, self-interested motivations, fragmented knowledge, diverse expertise and market contexts, yet simultaneously direct contributions to the greater collective benefits of the ecosystem (Boudreau, 2011).

Consequently, given pluralistic motivations distributed across a heterogeneous network of actors, a challenge remains as to how to implement some form of governance to harness extrinsic motivations and individual identifications, without undermining intrinsic motivations and collective identifications. Adler and Chen consider this tension between individual and collective creativity and formal control mechanisms in what they term ‘large scale collective creativity’ (Adler and Chen, 2011). They highlight that diagnostic control systems can be used in both an enabling and coercive manner (Adler and Borys, 1996). Used coercively, diagnostic control systems focus punitively on errors and negative variance and foster guilt and anxiety that undermine creativity. By contrast, an enabling use of diagnostic systems can, via procedural and outcome transparency, align individual performance with organisational or community goals, and thereby cultivate collective identifications while concurrently embracing self-interested motives (Wareham et al., 1998).

Yet the challenge of aligning individual and collective incentives is exacerbated in industrial ecosystems by the fact that complementors often work in consortia to deliver a good or service to the customer, risking problems of group or collective responsibility (Collins, 1990). In these settings, a reseller and group of implementation partners represent the product on behalf of the licence owner, franchise owner, or platform owner. And while the purchaser still has a large incentive to enforce control, the focus of the control is now distributed over more than one entity including any affiliated sub-contractor. This ‘joint action interdependence’ (Caglio and Ditillo, 2008 pg. 877) generates a team production environment where non-sequential, complementary processes are prominent, and direct lines of accountability may be difficult to define, let alone enforce (Miller et al., 2008). In these conditions, it is important to establish incentives for members to invest in complementary innovations; that is, work together to bring different capabilities and expertise to create effective and holistic solutions for clients. This
requires the simultaneous identification with multiple foci of commitment: a) the immediate task/client; b) the organisational group (e.g. partner consortia); as well as, c) the entire ecosystem. Given the prevalence of joint action interdependence in technology ecosystems, their effective governance must simultaneously cultivate community identifications to reduce undesirable variance; that is, increase contributions towards re-usable knowledge and community standards, positive externalities, complementary innovations, cohesion, and the ecosystem’s social goods; as well as individual identifications increasing desirable variance to encourage creative, explorative and entrepreneurial responses to client requirements and market developments.

2.4 Synopsis

The preceding argument can be summarised as follows: ecosystems require resilience; that is, a combination of stable and evolvable attributes. In technology ecosystems, we identify three main dimensions across which the stability-evolvability equilibrium must be managed: i) outputs; ii) actors; and iii) identifications. For technology ecosystem governance:

- Stability and variety in outputs is achieved through: a) variance reducing mechanisms to ensure standards; and b) variance increasing mechanisms to generate variety.

- Standard and variety in outputs is realised by actors, whose actions and behaviour must be simultaneously controlled and autonomous. This is enabled by: c) variance reducing mechanisms to control actors; and d) variance increasing mechanisms to leverage the autonomy of actors for innovative responses to client requirements.

- Achieving an appropriate balance between controlled and autonomous behaviour by actors is enabled by a combination of collective and individual identifications, where: d) community identifications reduce undesirable variance towards contributions to the social goods of the ecosystem; and e)
individual identifications increase desirable variance to encourage explorative and entrepreneurial responses.

These three dimensions are depicted in Figure 1 below. For purposes of visualisation, we represent the dimensions as orthogonally related and dichotomous, although non-orthogonal continuums are likely to be more accurate characterisations. The relationships between each sector are either strong and direct; that is, enabling or constraining; or weak and indirect; that is, compatible or incompatible.

Table 1 describes the individual interactions between each construct in detail. To simplify the exposition, we explore the 2-way interactions across the constructs, although 3-way interactions are obviously in effect.
<table>
<thead>
<tr>
<th>Interaction</th>
<th>Relationship</th>
<th>Qualification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collective</td>
<td>Control</td>
<td>Enabling</td>
</tr>
<tr>
<td>Collective</td>
<td>Autonomy</td>
<td>Incompatible</td>
</tr>
<tr>
<td>Individual</td>
<td>Control</td>
<td>Incompatible</td>
</tr>
<tr>
<td>Individual</td>
<td>Autonomy</td>
<td>Compatible</td>
</tr>
<tr>
<td>Standard</td>
<td>Control</td>
<td>Enabling</td>
</tr>
<tr>
<td>Standard</td>
<td>Autonomy</td>
<td>Incompatible</td>
</tr>
<tr>
<td>Variety</td>
<td>Control</td>
<td>Constraining</td>
</tr>
<tr>
<td>Variety</td>
<td>Autonomy</td>
<td>Enabling</td>
</tr>
<tr>
<td>Standard</td>
<td>Collective</td>
<td>Enabling</td>
</tr>
<tr>
<td>Standard</td>
<td>Individual</td>
<td>Incompatible</td>
</tr>
<tr>
<td>Variety</td>
<td>Collective</td>
<td>Constraining</td>
</tr>
<tr>
<td>Variety</td>
<td>Individual</td>
<td>Enabling</td>
</tr>
</tbody>
</table>

This portrayal offers the following insights. The upper-front-left corner (labelled RV) represents the area with convergence of variance reducing mechanisms (standard/control/collective). Alternatively, regions labelled as IV (variety/autonomy/individual) represent variance increasing mechanisms. As interactions between the dimensions are subtle, no single combination of governance mechanisms emerges as optimal by definition. It is reasonable to suggest that technology ecosystems with disproportionate emphasis on variance reducing mechanisms may be exercising excessively tight governance, resulting in low levels of generativity and evolvability. Alternatively, technology ecosystems with excessive reliance on variance-increasing mechanisms may suffer excessive fragmentation, loss of consistency, and inertia. This trade-off is evidenced in the divergent evolutions of the Linux and UNIX operating systems, two variants of a similar technology that evolved under different types of governance. Linux, as a
more tightly governed open-source endeavour, maintained a more unified global standard and cohesive technology core. Unix, by contrast, was fractured into many competing ‘flavours’ of the core operating system by the efforts of commercial actors to differentiate a standard into proprietary products (McPherson 2005). So the Linux ecosystem might be considered governed with moderate variety, moderate control, and high collective identification. Unix had no such variance reducing governance, and became excessively fragmented to the degree that it is no longer a significant competitor to Linux in terms of market share.

This is not to say that one relative weighting is superior to the other. Rather, it simply suggests that the trichotomous interactions can be subtle, rendering normative statement difficult without grounding in a specific empirical context. Indeed, although this conceptualisation implies that some equilibrium across the three dimensions is desirable, a skewed weighting towards one extreme or the other may well be appropriate for specific types of technology ecosystems at different stages of maturity (young versus established), or with dissimilar objectives (for-profit versus open-source).

2.5 Managing tensions in ecosystem governance: framing paradox

If technology ecosystem governance must simultaneously increase and decrease variance across outputs, actors, and identifications, then it is useful to review the literature on framing paradox that explores how organisations can meet competing demands simultaneously (Smith and Lewis, 2011). Managerial research has conceptualised paradox as either dualism or duality (Farjoun 2010). The idea of dualism is the view typically associated with trade-offs, conflicting alternatives framing tensions as mutually exclusive, either-or, or exhaustive classes (Farjoun, 2010). Perhaps the most widely acknowledged dualism in the management literature is the classic view that exploration and exploitation are exclusive trade-offs (March, 1991). In simple terms, the resource ‘pie’ is fixed, and efforts spent in exploration are, by definition, efforts not spent in exploitation; the choice between two competing options is always a zero-sum game (Farjoun, 2010).
The concept of duality, by contrast, views the two options not as competing and mutually exclusive; but rather as compatible, mutually enabling, and constituent of one another (Farjoun, 2010). Exploration cannot be achieved without the economic sustainability provided by exploitation. Likewise, current exploitation was enabled by past exploration. Admittedly, this is not entirely inconsistent with the internal logic of a dualism. Dualisms, via a premise of competing alternatives and zero-sum games, also acknowledge that trade-offs require balance, and that normally an appropriate mix of both options is required. The main difference and augmentation of a duality perspective is to emphasise the enabling and interdependent characteristics of each option. Hence, a logic of zero-sum games is replaced by a logic of positive-sum games.

If technology ecosystem governance must simultaneously increase and decrease variance, then understanding the mechanics of a duality vis-a-vis a dualism are central to the governance design. The requirement of ‘simultaneity’ clearly suggests that effective ecosystem governance must direct tensions towards complementary and enabling outcomes. Paradox scholars have suggested that paradoxes render complementary and enabling outcomes when they are latent, not salient (Smith and Lewis 2011). The idea is that the process of organising produces latent tensions that exist in an interdependent, mutually enabling duality. However, certain environmental conditions may bring paradoxical tensions to the foreground as salient, contradictory dualisms. Smith and Lewis offer three factors that can render paradoxical tensions salient: 1) plurality; 2) change; and 3) scarcity (Smith and Lewis, 2011). Plurality increases options, which can be manifest as competing goals to be evaluated with limited cognitive resources or comparative abilities, resulting in uncertainty. Likewise, change also highlights new options and competing opportunities, increasing uncertainty. Finally, resource scarcity raises the well-known problems of trade-offs; a decision to allocate limited resources in one area is a decision not to use these resources in another area. This is summarised in Table 2 below.
By framing tensions as either contradictory or complementary, we can re-examine the three tensions previously identified, but emphasise their conflicting and/or enabling outcomes as a result of technology ecosystem governance design. Table 3 below presents hypothetical scenarios where tensions are constraining or enabling.

<table>
<thead>
<tr>
<th>Tension</th>
<th>Contradictory scenario</th>
<th>Complementary scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standard – variety</strong></td>
<td>Niche-oriented complementors develop customised innovations that excel at meeting customer requirements. Yet their artisanal qualities limit re-use, despite overlaps and redundant features across competing complements. A lack of common interfaces and economic incentives limits sharing. As a result, other dedicated resources must work on standard infrastructure components. Labour is specialised and dedicated to specific tasks, there are no platform-core, or complement-complement synergies.</td>
<td>Predictable standards provide complementors assurance of the eventual scalability and reuse of their innovations across different contexts and users. Efficient communication of the core evolution path allows complementors to maintain compatibility over time. As such, investments in complement quality and quantity are greater.</td>
</tr>
<tr>
<td><strong>Control – autonomy</strong></td>
<td>Diagnostic control mechanisms are employed to reduce undesirable variance in quality, yet focus punitively on errors and so foster defensive responses that discourage risk and exploration undermine creativity; excessive restrictions restrict the ability of complementors to achieve their own objectives, thus reducing the value received from participation in the ecosystem.</td>
<td>Control is not random, but purposefully targets elements that reduce undesirable process and outcome variance that can threaten the ecosystem at critical points. By reducing transactional risk in specific areas (e.g. interoperability), control enables autonomous, option creating responses to clients in other areas (e.g. functionality, services).</td>
</tr>
<tr>
<td><strong>Collective – individual</strong></td>
<td>Individual identifications drive self-interested behaviours necessary in a for-profit commercial platform. Excessive focus on individual profit results in a highly competitive situation between core and complementors, detracting from the efforts of standardisation, direct and indirect network effects and public goods (Ibarra et al., 2005).</td>
<td>Complementors knowingly or unknowingly subsume their self-interested conduct to some form of direction giving (Demsetz, 1997) that orients their individual identifications towards the desired collective outcomes of the ecosystem. Collective identifications enable communal contributions without having adverse effects on the creative efforts of individual, niche participants.</td>
</tr>
</tbody>
</table>
In the case analysis that follows, we will investigate the conditions under which paradoxical tensions are manifest as latent or salient, enabling or contradictory, and consider consequences for the design of technology ecosystem governance.

3.0 Technology ecosystem mechanisms

3.1 Case background

To illustrate how these tensions function, we analyse a case of a complex technology ecosystem characterised by significant heterogeneity and generative activity. The following section introduces the case, its general processes and characteristics, where subsequent sections address research questions one and two directly. Our case analysis originates from a provider of an enterprise resource planning (ERP) software suite (‘software vendor’). The vendor is an international organisation present in over 45 countries through Europe, Asia, North and Latin America. The software is targeted at organisations with between 50 and 5,000 users. As such, SMEs constitute the majority of the vendors’ clients. Given the scale and scope of such a heterogeneous and dispersed client base, the software vendor has cultivated a sophisticated ecosystem of third-party implementation and solution partners to configure and modify the software, as well as develop focused complements/add-ons to the base software suite.

We identified three main roles in the software ecosystem: the software vendor, partners, and customers. While we have used the generic term ‘partner’, it must be acknowledged that there are a wide variety of companies included in this group who differ in many dimensions such as size; nationality; regional vs. national vs. international focus; type (independent software vendors, value-added resellers, etc.); as well as application development partners, implementation partners, and those who focus on both development and implementation. In our subsequent analysis, we equate all partners as ‘complementors’ from the platform literature. These can be entities that write applications or add-ons, but also implementers or training partners that offer service based complements.
Activities in the sales and implementation processes fall in three main categories: a) product and channel development; b) early stage sales; and c) implementation and late stage sales. These activities are illustrated in Figure 2 below.

The process begins with the marketing efforts of partners, which are normally managed by the individual partner, but may be undertaken with the support of the software vendor in the case of certified partners. Once the customer chooses the implementation partner, the partner and customer decide which software products to implement and how to implement them. The software’s open source-code enables partners to configure and/or customise the core software to match customer requirements. The licences for core products are owned by the software vendor and licence revenues are shared between the software vendor and partners, while the licence for any partner solutions or add-ons are owned by the developing partner. Multiple partners often collaborate, including single module implementations as well as mixed-module implementations composed of complements from multiple partners. In these cases, revenue sharing is negotiated on a case-by-case basis with guidelines provided by the software vendor. The software vendor encourages partners to use a standardised implementation methodology. There are significant potential benefits to standardising this process, including improved quality, fewer errors, and lower total cost of ownership.
Customer training and support are the responsibility of the partner, while the software vendor provides support to partners depending on their level of certification. Likewise, maintenance and upgrade of both core software and partner solutions are the responsibility of partners. The software vendor plays a more passive role by providing a core software product that is modular, highly configurable, and customisable. The software vendor offers some support to the implementation partners via sales support for larger clients, technical expertise, online tools, and templates for pre-sales proposal preparation, project management, and implementation tools. The eventual solution is the result of choosing certain modules, configuring and customising the core software when appropriate, integrating the core software with partner vertical solutions and third-party software, and changing the customer’s business processes when necessary.

We identified substantial differences in the nature of ecosystem participants in terms of the volume of transactions, annual revenues, number of employees, etc. Partners may engage in different roles related to the software, such as software development, sales and installation, training, or technical support. Partners may focus on a single role, or may develop competencies in multiple roles. Finally, partners may specialise in one or more industry sectors, and develop complements and perform software implementations for these specific sectors.

We illustrate this heterogeneity in Table 4 below which depicts possible activities and domains. Complementors can have a combination of geographical focus, functional focus, or specific expertise of an industrial sector, and perform support, training, implementations, and add-on development across these domains.
Consider the following examples.

- **Complementor A** may be a German partner with a broad profile, specialised in accounting, finance, tax and HR modules. As such, the firm could be considered a generalist partner providing support, training, and implementation in the German market, with deep knowledge of the accounting, tax, and HR regimes.

- **Complementor B** may be an American partner that has developed a global CRM add-on to support advanced on-line marketing, segmentation, and pricing. It does not handle generalist implementations, but simply develops and supports its product across all regions and industries.

- **Complementor C** may be a vertical specialist with deep knowledge of the food processing industry. It would perform general implementations for clients across a broad geographical region (e.g. South America), and sell and support several of its own vertical add-ons for specific industries.

It is evident that the profiles, activities, and markets of these complementor archetypes differ substantially, rendering a level of heterogeneity that, according to the software vendor, often leads to excessive variance in the quality of both complementors and complements.
3.2 Methods

Our analysis is based upon 31 semi-structured interviews conducted between November 2007 and June 2010. We employed a theoretical sampling approach, selecting subjects for their similarities as well as their differences. Software vendor employees (N=16) were selected from a wide selection of functional areas within the software vendor in order to acquire an understanding of the breadth of mechanisms and programs in place. These employees include channel management partners, as well as development managers and R&D directors. In addition, we interviewed a selection of independent software development and implementation partners (N=10). We also attended two international partner conventions (lasting three days), and participated in a variety of presentations and workshops designed to support implementation partners. Partners were chosen on initial recommendations from software vendor management and subsequent recommendations from the interviewees, with the objective of interviewing a representative cross section (size and industry, as well as geographical span). After this initial group, a snowball sampling approach was pursued, with future respondents chosen based on recommendations from the partners themselves. Finally, we interviewed a selection of customers (N=5) in distinct industry sectors to incorporate the end-user perspective.

We used an inductive, qualitative method in our research design in accordance with the exploratory nature of the study. The data was analysed using a grounded theory approach by identifying general concepts, organising the concepts into categories, and identifying properties for these categories and the relationships among them (Glaser and Strauss, 1967). Categories were chosen by triangulating four main sources: the software vendor’s partner program guidelines (archival data); interviews with representatives from the software vendor; interviews with partners and customers; and prior literature and theory reviewed in Section 2. This was an iterative process, with several rounds of category addition and consolidation. Substantial archival data from the software vendor was used to complement data gathered from the respondents. The initial interviews were semi-structured with the primary objective of identifying the various mechanisms that form the business model of the ecosystem. As the various platform mechanisms revealed themselves,
multiple follow-up interviews were staged with the interview subjects in order to ensure as complete a response as possible from each interviewee.

By the end of the interview process, the interviews became more formal, with the codes presented in Appendix A providing the basis for discussion. Furthermore, multiple validation incidents were staged in order to cross-check information provided by partners and customers with representatives from the software vendor, and vice versa. The interview process was concluded when no significant additional insights were obtained from additional data points and theoretical saturation was achieved. Additional cross-validation of the findings was obtained from industry literature, particularly competing software platforms.

3.3 Findings

In the following section we provide an operational description of specific ecosystem governance mechanisms and how they are designed to accommodate the three main tensions identified.

**Outputs: standard-variety**

Initial entry into the ecosystem as a registered partner is not particularly difficult; it requires approval and some validation by the software vendor. Partners seeking to develop skills and experience in the development and implementation of the software application can contact new partner support and training centres run by the software vendor or third-party training entities, although this is not obligatory. The open source-code is consistent with the positioning of the software toward SMEs, who typically do not welcome large changes to their own business processes, but would rather change the software to fit their existing business structure. Consequently, the acquisition or development of coding skills required for advanced implementations requires substantial effort. To address this requirement, the software vendor has created a number of utilities and training support centres to enrich creative responses to client demands, including: extensive code libraries,
technical support, integrated development environments with proprietary coding language, financing facilities, centralised sales and marketing support, as well as a community based platform that enables the sharing and recombination of tested code, modules, or vertical applications.

The mechanisms developed to reduce undesirable variance in process and output quality include an advanced regime of certification of both employees and partners for technology competencies that affect higher quality standards across outputs. Complements and solutions are also tested and certified. These certifications require that the partners utilise development and implementation templates that help ensure that requisite quality levels in functionality and inter-component compatibility are maintained. Additional reinforcement of the best development and implementation practices are achieved through training centres and community platforms with code libraries and consortia-based project management templates.

Table 5 below identifies the specific governance mechanisms for increasing desirable variance for output variety, and decreasing undesirable variance for output standardisation. All of these mechanisms are designed and enacted by the ecosystem core for use by the ecosystem partners.

| Table 5. Governance mechanisms enabling standard and variety in outputs |
|---|---|
| **Concept** | **Case data** |
| **Mechanisms enabling output variety (increasing desirable variance)** | • Open source-code of core applications/ protected source-code for partner solutions<br>• Specialised integrated development environments and development language<br>• Ease of customisation<br>• Code libraries<br>• Integrated development utilities<br>• Technical support<br>• Financing facilities<br>• Sales and marketing support<br>• Community solution catalogues |
| **Mechanisms enforcing standard output (reducing undesirable variance)** | • Configuration tools and templates<br>• Multi-partner project management templates<br>• Certification of partner solutions<br>• Certification of personnel<br>• Education and training facilities<br>• 3rd party verification of solution compatibility<br>• Required use of implementation tool/process<br>• Training centres<br>• Community platforms to facilitate code application re-use |
A theme that emerges in our analysis is that many of the mechanisms serve to both constrain undesirable variance and facilitate desirable variance simultaneously, working across different dimensions of the same process. For example, the integrated development tool for the software serves to increase complementor productivity, functional exploration, and creative problem solving; yet at the same time, constrains specific technological attributes that ensure quality and compatibility. The embedded, dual function of these mechanisms is a commonly found characteristic in many of the governance mechanisms we identified.

**Actors: control-autonomy**

The main mechanism use to negotiate control-autonomy tension is self-selection. Stated simply, partners are granted very high levels of autonomy upon entering the ecosystem. Via self-selection, they choose the level and nature of control that is appropriate for their business portfolio based upon the perceived benefits realised by adopting higher control levels. Table 2 identifies five levels of elective control in the software vendor ecosystem across the dimension of partner certification level, describing the infrastructure mechanisms and incentives available to participants at each level. Entry into the software vendor ecosystem begins at level 1. Partners have the option of remaining at this level perpetually. However, if they find advancement to higher levels beneficial, they can self-select themselves for higher levels by meeting the requirements.

These levels are, with limited exceptions, inclusive (level 3 is a small business specialist), meaning that the requirements and benefits of a higher level build upon those of the previous level. Advancement to levels four and five are permitted by the acquisition of internal points, which are awarded for a variety of achievements and are valid for a limited time. Points are awarded for many activities, including technical certifications of personnel, specialist competency designations, demonstrated competencies in specific application areas, validated customer references, technical testing of applications, successful participation in customer satisfaction surveys, and the volume of licences and/or revenue generated by the sales of the core platform product in specific markets.
Note that the key concept in the design of governance mechanisms is that each level of control imparts some additional value to both the periphery and core; that is, a value proposition is subsumed in the control infrastructure. For example, partner support and productivity centres offer support and training to newer partners who benefit from stimulus to achieve critical mass. This additional training also improves quality control in the final products and services. Likewise, higher levels of certification for both solutions and partners require more stringent control (e.g. testing by a third party, documented industry level implementations of solutions), yet simultaneously offer value to the periphery (higher quality solutions/services, co-branding with the software vendor), and the core (expanded catalogue of high quality partner solutions, greater scale and scope of ecosystem, additional revenue). Table 6 summarises the basic requirements for advancement to each level, as well as the potential value propositions for both periphery partners and the platform core.
<table>
<thead>
<tr>
<th>Partner Level</th>
<th>Requirements</th>
<th>Value to periphery members</th>
<th>Value to core</th>
</tr>
</thead>
</table>
| **Fifth (highest level)** | • Maximum point threshold  
• Partner personnel certified  
• Core vertical solutions certified  
• Demonstrated competency in focused strategic area  
• Participation in customer satisfaction surveys  
• Dedicated software vendor account manager for large clients  
• Validated customer references | • Highest level of co-branding  
• Maximum suite of software licences  
• Access to development libraries  
• Priority listing in solution finder catalogues  
• Priority real time technical support  
• Financing facility  
• Dedicated software vendor account manager for large clients | • Core product licence revenue  
• Expanded core software offering and vertical solutions  
• Specialist competencies  
• Increased licence revenue |
| **Fourth** | • Medium point threshold  
• Partner personnel certified  
• Partner solutions certified  
• Customer references  
• Partner subscription fee | • Expanded suite of licences for development and infrastructure software  
• Practice management support  
• More qualified personnel  
• Use of software vendor logo/branding  
• Inclusion of partner solutions in solution finder catalogue  
• Additional free online training  
• Additional sales and marketing support | • Core product licence revenue  
• More qualified implementation personnel  
• Fewer software errors  
• Improved software interoperability  
• Expanded portfolio of partner solutions |
| **Third** | • Organisational assessment  
• Partner personnel tested/certified | • Expanded suite of development tools  
• Licences for infrastructure software  
• Use of software vendor logo and branding  
• Priority technical support  
• Specialist community access  
• Financing facility | • Core product licence revenue  
• Subscription revenue |
| **Second** | • Subscription fees  
• Implementation templates | • Development tools  
• Productivity software licences  
• Online training platform  
• Marketing tools and support  
• Community technical support | • Core product licence revenue  
• Subscription revenue |
| **First (entry level)** | • Registration | • Access to peer network  
• Access to technical support communities  
• Marketing and sales tools | • Core product licence revenue |
In order to obtain higher certification status, partners must typically demonstrate a variety of achievements (to obtain points), or willingly submit to greater levels of control over their processes and outputs. In this respect it is similar to a market transaction, where something is offered in exchange for some expected value. Partners essentially self-select themselves for higher partnership status.

An analysis of the ecosystem mechanisms suggests that the majority simultaneously impart value to the individual partners, as well as the collective ecosystem. For individual partner benefits, we can identify a number that are consistent across all five levels of partners. These include: access to technical development tools and software libraries, sales and marketing support, status and legitimacy via co-branding, customer service and technical support, improved quality of add-ons, implementation skills, and customer satisfaction. The benefits accrue to solution partners in increasingly higher levels commensurate with the stricter requirements placed upon the solution providers.

The main benefit imparted to the platform core is the direct incentive for the partners to sell additional licences of the core products. Points are explicitly awarded for increased licence revenue for a variety of products. The secondary, but perhaps more important, benefits to the ecosystem are the indirect network effects of a larger, more vibrant ecosystem of qualified implementation partners; as well as the availability of software applications and other subsidiary complements (e.g. training, support, user communities) that offer potential value to clients.

Figure 3 below shows how partners can obtain similar control levels, but do so via diverse means, with differing sizes, markets, and expertise or business portfolios.
Identification: collective-individual

Participation in a for-profit industrial ecosystem is largely premised upon profit seeking behaviour. As such, self-interested, individual identifications will be predominant amongst partners as they work to address the challenges of financial sustainability common in any service organisation. Yet at various points, identifications can – and must – expand from strictly individual to include the: i) client and task; ii) a larger group or project consortia; or even iii) the greater collective ecosystem based upon reciprocal exchange where partners both contribute to – and benefit from – the greater ‘social goods’ made available on the platform. For the ecosystem to foster economies of scale and scope, to generate novel responses to client niche requirements while simultaneously cultivating cohesion within a fragmented portfolio, ecosystem governance must speak to both individual and community identifications concurrently. This is facilitated by operational components of the governance infrastructure, as well as less formal mechanisms of social governance.

Collective identifications are cultivated at the preliminary control levels. The main partner web portal for the software vendor identifies several hundred websites which are independently maintained and intended for use by partners and customers. These provide such services as discussion forums, blogs, white papers,
listings of partner solutions (with comments and ratings), job listings, events listings, downloads, code samples, bug reports, tutorials, and advice. In this forum, end users are able to share best practices in areas such as partner selection, implementation processes, and how to minimise total cost of ownership.

Greater levels of interdependence are cultivated through inter-partner collaboration for multiple partner projects, facilitated by the partner-finding platform. This is a directory of several hundred thousand partners who publish profiles on practice and technical specialities, thousands of registered add-on applications, and numerous broadcast opportunities for collaboration on existing jobs or future tenders. The typical motivations for partner-partner collaboration include access to specialised skills, building a suite of complementary applications, or expanding geographic reach. For example, a partner with expertise in a particular industry segment may turn to another partner for their expertise in a ‘horizontal’ application such as tax reporting. Alternatively, they may wish to expand internationally and find a partner in another country who could help localise their vertical application. While the technicalities of partner-to-partner business relationships will vary from legal region and job profile, the software vendor certainly encourages these collaborations (while not directly regulating them), and provides a number of generic best practice templates and tutorials on the commercial structure of such partnerships.

True collective identifications also require informal governance to communicate legitimacy and status. Low barriers to ecosystem entry can present challenges for complementors. On the outskirts of the periphery, new entrants can suffer from low status and illegitimacy due to newness (Stinchcombe, 1965). Given a lack of reputation and experience, they may choose to compete directly with incumbents, or attempt to carve a defendable niche and differentiate themselves as specialists. Economic theory argues that uncertainty exists concerning the quality of a firm’s products in market exchanges (Akerlof, 1970). As such, high status can serve as a signal or proxy of quality to help mitigate the problems of information asymmetry and uncertainty in economic exchanges (Podolny, 1994). Hence, a mechanism must be designed to: a) validate the quality of the products, services, and organisations on the periphery, and; b) where warranted, transfer the reputational stock to the periphery partners in a manner that is commensurate with the quality
achievement of the complementor’s organisation or products, yet does not expose the core to excessive risk of reputation deterioration. In this respect, status serves as a currency that the core can award to partners on the periphery via mechanisms of affiliation based on merit (Castellucci and Ertug, 2010). Our evidence clearly indicates that status was awarded to periphery partners based on evaluations of performance, achievement, or quality. The most obvious mechanism is the graduated certification levels, where higher levels of certification confer greater legitimacy on the complementors. Moreover, co-branding is also a graduated device, where partners with higher certification levels are permitted to use logos and other co-branding tools more liberally than lower level partners. As partners achieve greater certification levels, their overall image often assumes a higher degree of similarity with the core in terms of colour or appearance; we found several partner applications that were virtually indistinguishable from the core applications. Note as per the previous description, advancement to higher levels is permitted by a system of internal points that opens many avenues to higher status, thereby accommodating individual identifications while channelling them towards collective benefits. The results are summarised in Table 7 below.

<table>
<thead>
<tr>
<th>Table 7. Governance mechanisms enabling collective and individual identifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Concept</strong></td>
</tr>
</tbody>
</table>
| Mechanisms enabling individual identifications (increasing desirable variance) | • Business support services offered to partners (training, productivity, financing).  
• Internal currencies to enable: i) common measurement standards to assess heterogeneous accomplishments; ii) multiple paths through higher graduations of recognition; and iii) horizontal mobility through the ecosystem.  
• Process definitions to enable heterogeneous identifications to include the: i) client and task; ii) a larger group or project consortia; or even iii) the greater collective ecosystem. |
| Mechanisms encouraging collective identifications (reducing undesirable variance) | • Communal technical utilities, partner, solution, call-for-tender directories, code repositories, socialisation and training opportunities, and collaboration platforms.  
• Multi-partner collaboration templates and best practice norms.  
• Implicit incentives in hierarchical control infrastructure to facilitate reciprocity and institutionalise common behavioural patterns, expectations, and norms to improve the quality and cohesion of the ecosystem.  
• Status; the potential for increased status and legitimacy serve as an incentive for individual actors to contribute to greater collective efforts in the ecosystem. |
3.4 RQ 1 - Summary of main findings

Tension 1. Outputs: standard-variety

Variance-reducing mechanisms focus on technical compatibility and quality standards of the implementations and complements for a common product evolution, interfacing and compatibility standards, as well as the amortisation of common development costs. Variance increasing mechanisms include the open source-code enabling full flexibility in fulfilling functional requirements, and addressing different legal regimes, or operational requirements. Technical specialisation permits substantial freedom of choice in complementor positioning and service portfolios offered. Although distinct, standard-variety mechanisms are complementary; they enable increased creative freedom in the exploration of functions and services, yet simultaneously constrain and enforce standard technical requirements. As such, standard-variety mechanisms are often interdependent and mutually enabling.

Tension 2. Actors: control-autonomy

Technology ecosystems are quasi-markets with autonomous, profit-seeking agents who choose to participate in generative activity. Complementors consciously choose to submit their creative outputs to process and output based controls based upon the perceived value, similar to a market transaction. Ecosystem participants thus forfeit some generative liberty in exchange for a commensurate value imparted by the control mechanism, making self-selection a central lever in managing this tension. Analogous to a market exchange, the discretionary adoption of variance reducing control mechanisms by partners concurrently addresses disparate requirements, simultaneously imparting distinct values to complementor and core. Similar to the standard-variety tension, variance increasing mechanisms are often embedded in the variance decreasing mechanisms.
Tension 3. Identifications: individual-collective

For-profit ecosystems leverage self-interested behaviour and individual identification. As such, governance infrastructures must aggregate heterogeneous, random, and specialised contributions into a more cohesive ecosystem enabling the re-use and amortisation of complements through community contributions and collective identifications. These can be cultivated through explicit incentives and variance-reducing infrastructures such as communal technical utilities, code repositories, socialisation and training opportunities, and multi-partner collaboration platforms. Likewise, ecosystem governance can employ implicit incentives to facilitate reciprocity and institutionalise common behavioural patterns, expectations, and norms to improve the quality and cohesion of the ecosystem. One important mechanism is status; the potential for increased legitimacy serves as an incentive for individual actors to contribute to greater collective efforts in the ecosystem, either directly through community contributions, or indirectly, via higher quality products and services. At the same time, variance increasing effects can be cultivated by internal currencies that embrace autonomy and individual identifications, enabling: i) common measurement standards to assess heterogeneous accomplishments; ii) multiple paths through higher graduations of recognition; and iii) horizontal mobility through the ecosystem.

4.0 RQ2 - Paradox transitions between complementary and contradictory logics

Research question 2 states that tensions may manifest themselves as either contradictory or complementary logic. It is therefore worth asking if contradictory and complementary logics are present in technology ecosystems and how they are governed. To address this question, we analyse three cases that exemplify each primary tension: a) standard-variety; b) control-autonomy; and c) collective-individual. We consider the interaction with the secondary tension as well. After presenting each example, we summarise and discuss the implications in Section 4.4.
4.1 Complement generativity (standard-variety).

One of the main goals of a successful platform is to obtain substantial scale, longevity, and potential for wealth generation to attract complementors. Specifically, a potential complementor needs to be convinced that investments made in the platform will yield adequate long-term returns. Towards this goal, as the statement below indicates, there is clear evidence to suggest that the software vendor’s brand value and common marketing strategies have substantial positive effects for the partners.

[Software vendor]’s brand is helping partners to sell [software application] ..... The consolidated position of [software vendor] is recognised by the market... Now partners have more opportunities, especially access to big deals (implementation partner).

More complements suggest more options and potential value for clients. However, we also find some indication that liberal governance of the complements leads to some overcrowding, both within industrial sectors and geographic regions:

[Software vendor] has decided to allow as many verticals in the market as the partners wish to create…There really should be four or five broad verticals in [region], but instead there are about 300. The problem right now is lack of demand for the verticals (add-ons), because each is so specific (implementation partner).

The extreme heterogeneity of the regional niches can prevent implementation partners from achieving adequate size and scale of economies to develop high quality complements:

The problem is that the small to midsize partners who are regional or national neither have the size to dedicate programmers to developing vertical solutions, nor do they have enough implementations to amortise the cost of developing the solutions. The investment needed is too high compared to the eventual benefit since the partner absorbs all costs related to development,
registration, certification, and compatibility with new version of [application] (implementation partner).

Liberal governance of complements has additional negative consequences. Complementors must develop complements to the core, but some insecurity about the evolution of the core plagues their development efforts.

*Partners don’t clearly know where products are going, so partners don’t know if they are developing the products in the right way, because new versions of [application] can absorb the partner’s developments... The partners develop an add-on for a specific version of [application], but when a new version appears, it is difficult and expensive for the partner to make the add-on compatible with the new version (implementation partner).*

As a result, some partners even expressed a desire for tighter governance and reduced heterogeneity in order to avoid what they perceived as excessive redundancy and lack of financial amortisation across common complements.

*The situation where each partner creates their own verticals is a way for them to differentiate one from another, but in the end there are many versions of a particular vertical for a client to choose from, and in the end the client ends up paying for the costs of development. Having [software vendor] put their seal of approval on a handful of partner solutions in each vertical segment would result in lower total cost of ownership for the client and greater standardisation, but it would mean less differentiation for each partner. [Software vendor] would also have to compensate the partners for their versions of the verticals. So, while it would mean lower fees for the partners, it would also mean less investment in product development (implementation partner).*

The combination of platform evolution insecurity and overcrowding was summed up succinctly by one implementation partner:
Since the evolution of products is not clear, partners invest in solutions with no future or with a lot of competition, and they are competing in the same accounts with similar solutions (implementation partner).

Despite these criticisms, the software vendor does not attempt excessive regulation of the complements. There is a clear tendency to accept that locally embedded partners are best suited to address the market's extreme heterogeneity. However, they also acknowledge the implied agency problem.

We have to believe that our partners understand the local market conditions best. It would be impossible for [software vendor] to determine these conditions better than them ...So officially [software vendor] does not have any commitment to the customer. However, it always comes back to [software vendor] anyway (software vendor).

Permitting numerous complements to serve a specific market reflects a logic of generativity, market responsiveness, and sensitivity to unique client needs. Yet excessive supply and redundancy of complements creates an overcrowding problem, with limited common amortisation and expertise sharing, and overall higher total cost of ownership.

4.2 Standardised implementation strategies in heterogeneous markets
(control-autonomy)

ERP ecosystems are characterised by extremely high levels of heterogeneity, arguably more so than any other type of technology ecosystem, driven by the various accounting and legal regimes applicable in the country of use. The very high cost of localising software is generally the greatest single cost borne by the ecosystem collectively; core, complementor and therefore customers. As stated by a core channel manager:

Localisation affects three main areas: translation, legal requirements, and local business practices... It's not easy to manage the localisations at the
In an effort to reduce these localisation costs, the software vendor has developed a standardised implementation method that all implementation partners are recommended to use, and required to use for higher certification levels. The software vendor has attempted to embed a single, general-use methodology in the tool:

*With our tool* there is a clear trade-off between quality control, PM discipline, and speed and agility. So what we have tried to do is embrace this flexibility in the tool-via filters and project profiles. Partners can use the templates more or less as they like... (software vendor).

Yet the one-size-fits-all ideal of the method is admittedly utopian:

*Are we generating paper, or are we parameterising? Well – this is the ideal, but hard to achieve. We think that it is good to have a method to show to customers. On the negative side, it is still not easy enough to handle a customisation. There is a risk of over-automation and it could ruin the project (implementation partner).*

This relates not only to the heterogeneity of projects, but also to that of partners.

*It is important to stress the differences between the different partners. Some of the large partners will operate completely differently to the smaller partners. Especially considering the transition from sales to implementation (implementation partner).*

As a result, the implementation partners resort to pragmatic workarounds based on context and need:
Partners think it is difficult to use a methodology completely, because there are differences between countries, markets, customers, specialisation, etcetera that can affect the methodology implementation (software vendor).

Partners use methodologies provided by [software vendor] when it makes sense. They use what they need and ignore what they don’t. Their technicians develop their own tools to fill in the gaps of what is lacking from [software vendor] (implementation partner).

The software vendor recognises the trade-off, and further acknowledges the difficulties and costs of implementing extensive control regimes. They therefore attempt to leverage market mechanisms to supplement their control requirements:

You cannot test for the business functionality of the product. It is totally impossible. This is why we require the customer references – let the market speak. This is a very risky strategy of course, because as mentioned above, at the end of the day it all comes back to [software vendor] (software vendor).

The final statement, ‘it all comes back to [software vendor]’ refers to the potential agency costs borne by the core. They prescribe standardised implementation methods and client based testimonials as a market validation of complements and complementors to mitigate this risk. Yet, complementors adopt the standardised methods selectively and use their own techniques where necessary.

4.3 Competitive balance between partners (collective-individual)

A significant issue in the ecosystem is the competitive balance between partners. This relates to the number of partners in general:

There are no barriers to becoming a platform partner if you comply with their certification requirements. For example, they do not limit the total number of partners in the channel due to anti-trust regulations (implementation partner).
This can lead to high levels of competition between partners in specific markets. The frustration from the implementation partners is evident:

[Software vendor] is too focused on sales of licences. Every month there is tremendous pressure from [software vendor] for the partners to report results. Most partners are losing money due to this excessive pressure and short-term focus (implementation partner).

The competitive pressures between partners have a variety of undesirable consequences. For example, complementors work together in terms of training and cooperation on projects, but at other moments, they become competitors.

Training is handled by external centres, certified by [software vendor] which are managed by third parties. The main problems with this external training model is that it creates a conflict of interest for these ‘training centres’, who also provide other services such as programming support for other partners, creation and implementation of add-ons... The partners doing the training are my competitors, and if I send my people there, well... (implementation partner).

Hence, there is a perception that the software vendor lacks loyalty to individual partners, and this breeds a certain defensive posture amongst some:

If you ask a partner who their biggest competitor is, they'll say 'another partner'. The implication is that there is more competition between partners offering [application] than with competitors offering other software packages, like [competing packages]... If there are three partners competing for the same work, [software vendor] always wins (implementation partner).

Frequently, partners work together successfully to combine areas of expertise to solve complex projects. But the competitive tendencies remain.

Where collaboration between partners can work is in situations where you need special expertise for a particular project, and you create a joint venture
with another specialised partner. However, the large partners don't have any incentive to help a small partner grow (implementation partner).

The competitive situation between partners can often result in conflicts within the same project or vertical. Common country-level modifications should ideally be shared and amortised over as wide a market as possible. However, this is not always the case.

[Software vendor] tries to integrate the local regulations of each country into their software, but there are always areas which are lacking, which need improvement, or which have errors. And each partner makes their own changes, but there's no incentive to share this with other partners or even with [software vendor]. All of the localisations created by the individual partners are different and not necessarily compatible. So all of the partners have the standard version of [application] and their own ‘standard plus’ version. This becomes a point of differentiation, where partners claim that their standard plus is better than the others (implementation partner).

The statements suggest that the extreme heterogeneity of the ERP market manifests itself in legitimate modifications to the software which are driven by localisation requirements. However, there is some evidence of overcrowding in the complementor market that constrains cooperation between complementors in instances where the sharing and common development of the localisations would be optimal.

4.4 RQ2 summary

The examples presented exemplify related yet distinct paradoxical tensions. Our analysis is premised in the idea that tensions exist as part of organising socio-economic activity. They cannot be avoided or permanently resolved, rather, harnessed or directed towards enabling or disabling outcomes (Faraj et al. 2011, Farjoun 2010).
Example 1 demonstrates how extreme competitive behaviour can produce redundant investments in lower quality complements, overcrowd markets, and reduce prices. Where the primary tension is standard-variety, a secondary collective-individual tension also manifests contradictory effects, where partners create redundant complements for similar markets. The main conditions causing the dominance of a contradictory logic are plurality (excessive supply of competing complements) and change (uncertainty about the technological evolution of the platform core.)

Example 2 identified that resistance to the use of a standardised implementation method is limiting technological compatibility across complements and keeping implementation costs high. The primary tension is control-autonomy, with secondary effects across the standard-variety tension. Here, the main impetus of a contradictory framing is plurality, specifically the extreme market heterogeneity inherent in the ERP market (environmental) combined with some emotional resistance to centralised control regimes.

Example 3 showed how the liberal access to the ecosystem by partners, combined with un-coordinated, self-interested behaviour limits cooperation and the sharing of common regional standards, and accelerates downward price pressure in the ecosystem. Where a collective-individual tension was dominant, the effects were also manifest across standard-variety. In this instance, the conditions of a contradictory framing are plurality and scarcity, specifically the extreme concentration of complementors in some markets, with scarce coverage in others.

The examples and potential managerial responses are summarised in Table 8.
<table>
<thead>
<tr>
<th>Primary tension</th>
<th>Complement generativity</th>
<th>Standardised implementation strategies in heterogeneous markets</th>
<th>Competitive balance between partners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secondary tension</td>
<td>Collective-individual</td>
<td>Standard-variety</td>
<td>Standard-variety</td>
</tr>
</tbody>
</table>

**Duality (complementary logic)**

Librally governed complementors use market mechanisms to respond to local market requirements. A greater number of complements imply increased choice and potential value for clients, subsequent growth of the ecosystem, and greater value for complementors.

<table>
<thead>
<tr>
<th>Complement generativity</th>
<th>Standardised implementation strategies in heterogeneous markets</th>
<th>Competitive balance between partners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liberally governed</td>
<td>Standardised control methods enable consistent, standardised quality control of implementations. Second order effects include increased technical compatibility of complements across core implementations.</td>
<td>Platform core wants to generate as much growth as possible in the ecosystem. This includes growth of overall licence revenues via liberal access by partners to the ecosystem. Market mechanisms will dictate allocation of resources by partners, and facilitate quality control. More partners generate greater choice, equitable pricing, and potential value for customers.</td>
</tr>
</tbody>
</table>

**Dualism (contradictory logic)**

Overcrowding and excessive competition between complements. Consequent downward price pressure limits market value and resource allocation to individual complements thereby constraining quality. Lack of coordination across similar complements creates redundancy, hindering amortisation of functions that are common at legal- or sector-level, increasing localisation costs and total cost of ownership.

<table>
<thead>
<tr>
<th>Complement generativity</th>
<th>Standardised implementation strategies in heterogeneous markets</th>
<th>Competitive balance between partners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overcrowding and excessive competition amongst partners. Consequent downward price pressure limits potential revenue, cooperation and re-use of common regional modifications across partner consortia.</td>
<td>Implementation partners ignore standardised implementation methods, either selectively or completely. Increased fragmentation. Localisation costs remain high, technological compatibility across complements constrained.</td>
<td>Overcrowding and excessive competition amongst partners. Consequent downward price pressure limits potential revenue, cooperation and re-use of common regional modifications across partner consortia.</td>
</tr>
</tbody>
</table>

**Conditions of contradictory logic**

Plurality and change. Plurality. Plurality and scarcity.
<table>
<thead>
<tr>
<th>Diagnosis</th>
<th>Possible governance response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Diagnosis</strong></td>
<td><strong>Possible governance response</strong></td>
</tr>
<tr>
<td>Over-crowding of complements. Large numbers of partners generate similar complements for individual clients. Where possible, re-use and complement ‘best-in-class’ development is constrained by redundant efforts.</td>
<td>Interventions to limit # of complement types in defined areas. Increase coordination to reduce redundancies.</td>
</tr>
<tr>
<td>Standard methods insufficiently comprehensive to accommodate extreme heterogeneity resulting from disparities across legal, geographic, or industrial sectors, and diverse genesis of implementation partners. Disdain for centralised control.</td>
<td>Increase comprehensiveness of standard method. Allow for graduated use, legitimising variance from standard use in certain cases. Establish feedback mechanisms to incorporate legitimate variance into future versions.</td>
</tr>
<tr>
<td>Over-crowding of complementors. Liberal complementor entry renders hyper-competitive conditions in some markets, with limited coverage in others. Limited re-use and collaboration across markets constrains social goods in the ecosystem.</td>
<td>Elect market-winners in regions with over-capacity. Seed markets with under-capacity.</td>
</tr>
</tbody>
</table>

Examples 1 and 3 specifically resemble a classic 'market failure' from the economics literature, where the outcomes are socially sub-optimal due to self-interested behaviour, bounded rationality, uncertainty, and information asymmetries (Bator, 1958). As such, potential managerial responses to both of these problems are analogous to public policy responses; namely, some form of quasi-regulation. For example 1 (complement over-crowding), governance could be developed to limit the number of entrants into specific functional areas, awarding some protected status to complements in exchange for commensurate levels of increased investment and quality in the complement. The existence of several very high quality complements could be diffused through the ecosystem. In example 2, the extreme heterogeneity of the market may inevitably limit the adoption of standard implementation methods. In which case, a pragmatic response may be to legitimise variance in use, and leverage the variance as a legitimate source of improvement to the standard process to make it more comprehensive. Finally, in example 3 (over-crowding of complementors), governance could be developed to limit complementor entry into highly competitive markets, while seeding new complementor entry into under-served markets.
Considering the conditions making contradictory logic salient, vis-à-vis complementary logic, it is noteworthy that plurality is such a dominant source of transition between complementary or competing tensions. Smith and Lewis (Smith and Lewis, 2011 p. 390) state that ‘plurality expands uncertainty and surfaces competing goals and inconsistent processes (Cohen and March, 1974)’. Our case analysis confirms this position. Interestingly, where we found evidence that suggested that while change did create uncertainty in our interview respondents, it was not a major factor. This may be due to the fact that technology ecosystems are designed for evolvability, and the participants are accustomed to this reality.

A second noteworthy observation is that we represent the three tensions as basically orthogonal in Figure 1. On one level, our data agrees with this representation, in that we find that no single tension operates separately from the others in our analysis. But on another level, there seems to be some hierarchical nesting of the tensions consistent with our argument that stability and evolvability refer first and foremost to: i) outputs, which are created by; ii) actors, whose behaviour is influenced by; iii) identifications. This is fully consistent with example 1, where the control-autonomy tension (actors) subordinately contributes to the standard-variety tension (outputs). In examples 2 and 3, the primary tensions of control-autonomy (actors) and collective-individual (identifications) are important to the degree that they have consequences for outputs (standard-variety). However, our data does not support a perfect 3 level hierarchical nesting. Rather, we may think of them as mediating and moderating relationships; actors create outputs; actors are influenced by identifications.

5.0 Discussion

Technology ecosystems must manage the unwieldy challenge of simultaneously realising stability and evolvability through a combination of variance-increasing and variance-decreasing mechanisms (Yoo et al. 2012, Tilson et al. 2010). Specifically, ecosystems must increase serendipity and creativity in products, solutions, and services, yet constrain inferior quality outputs and over-abundance (Tiwana et al.
They must simultaneously harness autonomous behaviour for generativity, yet subsume it to direction-giving and control (Demsetz, 1997). Finally, ecosystems must harness individual identifications for agile, entrepreneurial responses to client needs (Faraj et al. 2011), yet cultivate collective identifications for the larger social goods of the ecosystem (Ibarra et al. 2005).

RQ 1 asked how these main tensions in technology ecosystems are addressed in technology ecosystem governance. We identified three main tensions salient across outputs, actors, and identifications – and thereafter identified governance mechanisms used to either increase desirable variance or decrease undesirable variance across each tension. For outputs, undesirable variance is constrained via processes and tools focused on technical standards and quality, with an emphasis on the programmatic compatibility of heterogeneous complements with the core and across and between complements. Desirable variance is fostered through the open source-code of the core application that permits customisation of the application to client requirements. For actors, variance-reducing control is manifested through graduated control regimes that offer some perceived value for the complementors. Variance-increasing autonomy is embraced through self-selection; actors essentially choose their desired control levels based upon the perceived utility of the control mechanisms to their business as implementation and development partners. For identifications, undesirable variance is constrained through communal utilities and socialisation to common behavioural norms, as well as status conferred by the core to the complementors for direct and indirect contributions to the social goods of ecosystem. Desirable variance is enabled through business support services to aid entrepreneurial action and internal currencies that accommodate heterogeneous accomplishments and multiple trajectories through control mechanisms.

Table 9 summarises the variance increasing or decreasing mechanisms for each tension.
Table 9. RQ 1 Main Findings

<table>
<thead>
<tr>
<th>Variance decreasing mechanisms</th>
<th>Variance increasing mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outputs: standard-variety</td>
<td>Open source-code permits full functional freedom in tailoring client solutions, accommodating different legal regimes and other client or operational requirements.</td>
</tr>
<tr>
<td>Process and output tools</td>
<td>Specialised integrated development environments and development language (variance decreasing mechanisms embedded in variance increasing mechanisms.)</td>
</tr>
<tr>
<td>focused on technical compatibility and technical quality of the implementations and complements (development templates, certifications, compatibility tests, market testimonials).</td>
<td></td>
</tr>
</tbody>
</table>

| Actors: control-autonomy     | Self-selection through control processes is similar to a market transaction; ecosystems participants offer or forfeit something in exchange for additional value. Self-selection allows freedom to choose certain aspects of control portfolio while rejecting others. |
| Self-selection: graduated control infrastructures, with greater value offered to complementors in exchange for submitting to higher levels of control by the core. | |

| Identifications: collective-individual | Communal technical utilities, code repositories, socialisation, training opportunities, and collaboration platforms. |
| Implicit incentives and infrastructure to facilitate reciprocity and institutionalise common behavioural patterns, expectations, and norms to improve the quality and cohesion of the ecosystem. | Business support services offered to partners (training, productivity, financing). |
| Status: the potential for increased status and legitimacy serve as an incentive for individual actors to contribute to the greater collective efforts in the ecosystem. | Internal currencies to enable: i) common measurement standards to assess heterogeneous accomplishments; ii) multiple paths through higher graduations of recognition; and iii) horizontal mobility through the ecosystem. |

RQ 2 further posited that tensions can manifest themselves as either contradictory or complementary logics. It therefore questions if contradictory and complementary logics are present in technology ecosystems, and if so, if so, how are they governed? Our study provides substantial evidence to the argument that tensions are manifest as both complementary and contradictory logics in technology ecosystems. The design of ecosystem governance acknowledges this possibility, and explicitly aims to leverage these tensions as mutually enabling forces. However, environmental, operational, and cognitive conditions cause transitions between complementary and contradictory logics. Consistent with theoretical
arguments, we find that plurality is the main source of the dominance of contradictory logics, where change and scarcity make similar, but lesser, contributions. In operational terms, plurality is expressed in the extreme heterogeneity of the ecosystem across geographic and functional markets renders standardized processes and outputs problematic. The effects of plurality are manifest in liberal access for complementors and limited regulation of complements that create problems of excess supply and over-crowding in complementor markets, as well as variable quality and redundancy. Finally, limited coordination rendered some geographic regions underserved, where others experienced hyper-competitive conditions, further discouraging collaborative efforts to generate common best-practice. Table 10 highlights these findings.

<table>
<thead>
<tr>
<th>Duality (complementary logic)</th>
<th>Dualism (contradictory logic)</th>
<th>Cause</th>
<th>Potential governance response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outputs: standard-variety</strong></td>
<td>Liberal access for complements. Greater value and options.</td>
<td>Excessive supply, redundancies, fragmentation.</td>
<td>Plurality &amp; change: over-supply, hyper-competitive conditions.</td>
</tr>
<tr>
<td><strong>Identifications: collective-individual</strong></td>
<td>Liberal complementor access. Greater expertise and choice.</td>
<td>Overcrowding and hyper-competition. Constrained collaboration and social goods.</td>
<td>Plurality and scarcity: hyper-competitive conditions in some markets, limited coverage in others.</td>
</tr>
</tbody>
</table>

In an overall evaluation, the predominance of contradictory logic is not necessarily an indication of the inadequate design of the governance mechanism. It may simply be a property of the system that can be purposefully managed at appropriate levels, not considered innately negative, but rather symbiotic and necessary at some levels (Chua and Wareham 2008). As such, a valid managerial focus may not be concerned with the simple existence of contradictory logics, but rather extreme imbalances that cause undesirable outcomes. On this dimension, technology ecosystems function as semi-regulated marketplaces, where standard public policy and regulatory debates offer analogous recommendations. Clearly, attention should
be paid to the problem of overcrowding amongst complementors. This may include imposing tighter restrictions on entry as a complementor, or other types of quasi-regulatory mechanisms. Likewise, greater coordination of complements would help reduce excessive redundancies and overcrowding in the complement market. Towards this aim, our analysis highlighted many similarities between contradictory logics and the classic market failure problem.

5.5 Contributions to theory

If managing paradox requires meeting competing needs simultaneously, then the question of what is ‘simultaneity’ and how it can be leveraged by the governance mechanisms is pertinent. Our analysis demonstrated how ecosystem governance can address competing requirements simultaneously. Specific mechanisms such as integrated development environments and coding libraries can serve variance-increasing objectives on dimensions of functionality and customisation, yet concurrently constrain undesirable variance in terms of technological quality and compatibility standards. Likewise, similar to any economic transaction, governance mechanisms can simultaneously impart different values to different constituents. The self-selection of control regimes allows complementors on the periphery to realise one type of benefit (speaking to individual identifications), in exchange for some other benefit realised by the core, as well as the greater ecosystem (speaking to collective identifications and communal benefits). Hence, governance mechanisms cannot only address multiple constituents simultaneously, they can concurrently function across multiple tensions, consistent with our orthogonal representation. Stated in more general nomenclature, the requirement of ‘simultaneity’ was evidenced: i) within individual tensions (e.g. increasing variance on characteristic alpha, while decreasing it on characteristic beta); ii) across multiple tensions (addressing them concurrently); and iii) across multiple constituents (addressing the divergent needs of core and periphery actors concurrently). This property of ‘simultaneity’ appears particularly pertinent to any system characterised by severe paradoxical tension. The explicit observation that effective governance mechanisms can, or should, simultaneously impart different values to different constituents, across distinct tensions and sub-dimensions thereof, can guide their
design and enactment – be they technology ecosystems or other types of social systems. In this respect, the use of internal currencies and elective mechanisms that direct efforts towards general outcome, but permit pluralistic, self-selected avenues towards it, can be useful.

As mentioned previously, there is a large debate within the paradox discourse as to whether paradoxical tensions exist as inherent features of the system, or whether they emerge as social constructions resulting from human cognition, rhetoric, and social behaviour (Cameron 1986, Poole and Ven 1989, Smith and Lewis 2011). We find evidence that both positions are valid. Furthermore, if we agree that theories are human constructs; as are languages and vocabularies used to understand a phenomenon (Smolin 2006), then such an argument suggests that duality and dualism exist, but as theoretical languages that guide the empirical interpretation of the researcher.

Consequently, one can ask for which situations are the languages of dualism and duality appropriate. Dualism may be the language best suited to describe the manner in which actors frame decisions; trade-offs that emerge in the face of plurality, change, and scarcity. If individuals frame problems as trade-offs, then this is an empirical observation that demands a theoretical language that appropriately explains and predicts the micro-level decision making of individual social actors. An attempt to suggest that duality offers a morally superior way of framing a problem may be a naively utopian gesture of normative theorising, or simply, an inappropriate application of one theoretical language, when another would be more appropriate. However, we may consider that duality is an appropriate language of ecosystem design for the desired macro-level, that of emergent outcomes in collective systems. Should we then posit that observations concerning the desired collective outcomes of ecosystems are best described as a duality, where the decision making behaviour of individual actors is best described as a dualism? Perhaps. The question alludes to the common problem that occurs when specific phenomena are best described by one language at one level – and another language at a different level. Both languages are successful within the range of their domains, yet remain internally inconsistent, and uncomfortably embedded – like poorly fitted Russian dolls. At the risk of infinite regress, such inconsistency is
largely because duality-dualism framing is, in fact, a dualism. As such, the observation does speak to the argument that dualisms are social constructions with strong consequences for how researchers interpret data.

Additionally, there are several thematic areas where the control literature is less complete, but where our study may contribute. One area is the notion that control archetypes are not static, but rather dynamic in nature. Traditional theorising on transaction governance has emphasised variance theories, where transactional, institutional, or environmental antecedents change and governance forms and control outcomes respond in some linear fashion (Klein et al., 2011). If evolvability is considered a focal property of the ecosystem, then governance mechanisms also need to embrace changing maturity levels of the ecosystem through time for the same set of antecedent conditions, oscillating between several archetypes, or evolving with the maturity of the ecosystem (Thrane 2007). As an example, the preliminary phases of the platform’s lifecycle may require a loose control that facilitates the recruitment of complementors to the ecosystem and the subsequent generation of complements and scale; a classic seeding process. It is assumed that there will be competing platforms, and given the permeating logic of natural monopolies, achieving critical mass is important. In intermediate stages, stricter control may be appropriate as achieving critical mass and network effects become less imperative, and the ecosystem can mature and evolve at controlled rates.

Finally, there is a substantial lack of research on control mechanisms in networks that resemble technology ecosystems, including re-seller, partnership, and franchising networks where responsibility is distributed across more than one entity. This problem is not only more severe from the perspective of the final purchaser, but also between other members of the distribution networks where ‘joint action interdependence’ (Caglio and Ditillo, 2008 p. 877) generates a mix of non-sequential processes that obscure accountability (Miller et al., 2008). Given the emergence of affiliate and social marketing, and other Internet-based marketing structures where direct lines of responsibility are latent (Fox and Wareham, 2010), additional research on this topic is desirable.
6.0 Concluding remarks

The use of ecosystems to drive generative responses to market demands is growing, well beyond pure technology sectors. Yet harnessing the efforts of heterogeneous actors in an ecosystem to meet the needs of diverse markets requires some balance; if one thousand flowers grow, inevitably, some will be undesirable and harmful to the ecosystem. In the extreme, the unconstrained growth of low quality innovations can kill a platform - governance is necessary.

Our analysis identified three constituent tensions of the stability-evolvability equilibrium in technology ecosystems that need to be managed: standard-variety, control-autonomy, and collective-individual. We subsequently analysed the governance of a commercial software ecosystem that accommodates these tensions, increasing and decreasing variance simultaneously within and between different tensions and constituents. We describe the specific mechanisms, and delineate conceptual properties that may be generalisable to other technology ecosystems. Our evidence demonstrates the presence of enabling and constraining logics, identifying their causes, effects, and potential responses in the technology governance design. Technology ecosystem governance can be designed to harness tensions as enabling forces that serve the overall needs of the platform. However, circumstances can cause a disabling realisation of these tensions as a ‘market failure’. Potential responses to these risks in the design of the technology ecosystem governance are offered, as well as the properties making them relevant to technology, or other forms of industrial ecosystems, that are characterised by high degrees of paradox.
References


## Appendix A

### Table 11. Coding Scheme

<table>
<thead>
<tr>
<th>Constructs</th>
<th>Codes</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standard</strong></td>
<td>Variety</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Variance reducing</td>
<td>(Tilson et al. 2010, Yoo et al. 2012)</td>
</tr>
<tr>
<td></td>
<td>Variance increasing</td>
<td>(Tilson et al. 2010, Yoo et al. 2012)</td>
</tr>
<tr>
<td></td>
<td>Positive N/W effects</td>
<td>(Katz and Shapiro 1994)</td>
</tr>
<tr>
<td></td>
<td>Negative N/W effects</td>
<td>(Katz and Shapiro 1994)</td>
</tr>
<tr>
<td></td>
<td>Cohesion</td>
<td>(Hagel and Brown 2005, Messerschmitt and Szyperski 2003, Evans and Schmalensee 2007)</td>
</tr>
<tr>
<td></td>
<td>Fragmentation</td>
<td>(Evans and Schmalensee 2007, Messerschmitt and Szyperski 2003, Hagiu 2010)</td>
</tr>
<tr>
<td></td>
<td>Scale economies</td>
<td>(Panzar and Willig 1981, 1977)</td>
</tr>
<tr>
<td></td>
<td>Scope economies</td>
<td>(Panzar and Willig 1981, 1977)</td>
</tr>
<tr>
<td></td>
<td>Ecosystems</td>
<td>(Faraj et al. 2011, Yoo et al. 2010, Tiwana et al. 2010)</td>
</tr>
<tr>
<td></td>
<td>standards/homogeneity</td>
<td>(Faraj et al. 2011, Yoo et al. 2010, Tiwana et al. 2010)</td>
</tr>
<tr>
<td></td>
<td>Ecosystem innovations/heterogeneity</td>
<td></td>
</tr>
<tr>
<td><strong>Control</strong></td>
<td>Autonomy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enabling</td>
<td>(Adler and Borys 1996)</td>
</tr>
<tr>
<td></td>
<td>Coercive</td>
<td>(Adler and Borys 1996)</td>
</tr>
<tr>
<td></td>
<td>Diagnostic</td>
<td>(Adler and Borys 1996, Caglio and Ditillo 2008)</td>
</tr>
<tr>
<td></td>
<td>Outcome</td>
<td>(Adler and Borys 1996, Caglio and Ditillo 2008, Demsetz 1997)</td>
</tr>
<tr>
<td></td>
<td>Boundary</td>
<td>(Adler and Borys 1996, Caglio and Ditillo 2008)</td>
</tr>
<tr>
<td></td>
<td>Distributed</td>
<td>(Adler and Chen 2011, Caglio and Ditillo 2008, Collins 1990)</td>
</tr>
<tr>
<td><strong>Collective</strong></td>
<td>Individual</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Motivations:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Singular</td>
<td>(Ibarra et al. 2005)</td>
</tr>
<tr>
<td></td>
<td>Plural</td>
<td>(Ibarra et al. 2005)</td>
</tr>
<tr>
<td></td>
<td>Client/task identification</td>
<td>(Adler and Chen 2011, Ibarra et al. 2005)</td>
</tr>
<tr>
<td></td>
<td>Group/team identification</td>
<td>(Adler and Chen 2011, Ibarra et al. 2005)</td>
</tr>
<tr>
<td></td>
<td>Community norms</td>
<td>(Ibarra et al. 2005)</td>
</tr>
<tr>
<td></td>
<td>Self-selection</td>
<td>(Borjas 1987)</td>
</tr>
<tr>
<td></td>
<td>Status</td>
<td>(Castellucci and Ertug 2010, Podolny 1994)</td>
</tr>
<tr>
<td></td>
<td>Legitimacy</td>
<td>(Stinchcombe 1965, Podolny 1994)</td>
</tr>
</tbody>
</table>