



Information Systems Research

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Editorial—Evolvable Systems: Through the Looking Glass of IS

Ritu Agarwal, Amrit Tiwana

To cite this article:

Ritu Agarwal, Amrit Tiwana (2015) Editorial—Evolvable Systems: Through the Looking Glass of IS. Information Systems Research 26(3):473-479. <https://doi.org/10.1287/isre.2015.0595>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2015, INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Editorial

Evolvable Systems: Through the Looking Glass of IS

Ritu Agarwal

Center for Health Information and Decision Systems, Robert H. Smith School of Business, University of Maryland,
College Park, Maryland 20742, ragarwal@rhsmith.umd.edu

Amrit Tiwana

University of Georgia, Athens, Georgia 30602, tiwana@uga.edu

We explore how “Red Queen” competition is increasing the competitive premium on “evolvable” information systems (IS). Ephemeral market advantage coupled with relentless innovation spawning trends such as the Internet-of-Things and additive manufacturing are amplifying the importance of evolvable systems across all industries. We discuss uncharted theoretical and empirical territory for IS research on evolvable systems. The elusiveness of some of these phenomena to other disciplines offers a unique opportunity for IS scholars.

1. Introduction

In Lewis Carroll’s *Through the Looking-Glass*, Alice asks the Red Queen why she is constantly running yet remaining in the same spot. “Now, here, you see, it takes all the running you can do, to keep in the same place... to get somewhere, you must run at least twice as fast...!” (Carroll 1917, p. 39, Figure 1). This phenomenon—*Red Queen competition*—implies that an entity (such as an organization, market offering, or system) must evolve progressively faster just to keep up with its cohort of rivals (Barnett 2008; Stenseth and Smith 1984; Tiwana 2014, p. 39). Speedier evolution of one player raises the bar for all rivals. The penalty for falling behind is mortality.

The *evolvability* of man-made information systems (IS) can preordain the survival of organizations, particularly as software is increasingly embedded in nontechnology products and processes, and as the Internet-of-Things (IoT) appropriates billions of unexpected physical objects into contemporary systems. We define the evolvability of a system as its capacity to efficiently change to serve new purposes and emerging possibilities (de Weck et al. 2011, p. 187). At some level, to the extent that systems can be modified, updated, or otherwise altered, evolvability is not infeasible but it often comes at a high cost. Arguably, today, systems that are not designed to evolve are likely to irrecoverably fall behind those that are, as are the organizations that depend on them. Yet, our discourse has not yet ventured downstream enough to ask questions about creating evolvable information

systems.¹ We hope to spark a conversation around why evolvable synthetic systems and the organizations that they create are increasingly inseparable from the IS discipline, and opportunities for research where IS can play a pivotal role.

We first describe three reasons why evolvable systems command a rising competitive premium in an industry-agnostic manner. We then discuss three uncharted spaces around evolvable IS. These are summarized in Table 1.

2. A Growing Evolvable Systems Premium

Evolvable systems competitively matter more today because interminable systems can straitjacket organizations facing ephemeral competitive advantage, non-IT (information technology) industries are acquiring properties of the IT industry, and the IoT is increasing the scale and distributedness of systems.

2.1. Competitive Advantage Is Ephemeral But Systems Can Be Interminable

Competitive advantage is increasingly fleeting (D’Aveni et al. 2010) yet systems can have long, protracted lifespans. Many outlive their intended lifespans by decades, as repeated failed attempts to replace the U.S. air traffic control system built in the 1970s predating even the

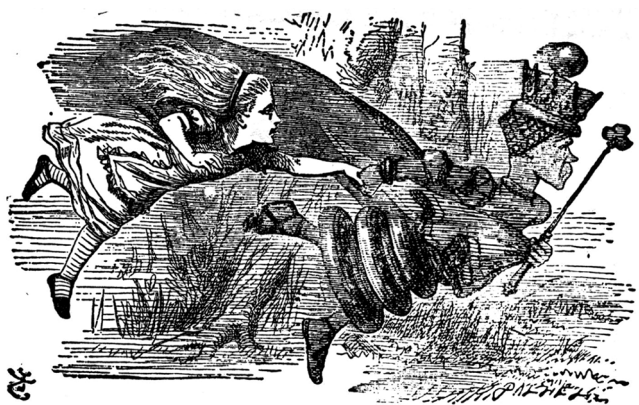
¹ We use the term systems to refer exclusively to software-based information systems (not organizational systems), architecture to refer to systems architecture, and apps to refer to software applications.

Table 1 Evolvable Systems: Phenomenon, Challenges, and Research Implications

Phenomenon	Challenge	Research implications
Interminable systems in an era of ephemeral competitive advantage.	Systems not constructed to be evolvable straitjacket organizations into succumbing in Red Queen competition.	• Disassemblable systems evolvable at the pace of the environment. • Dynamic app-infrastructure boundaries.
Nontechnology market offerings inherit properties idiosyncratic to software due to embedding of: (a) software in them and (b) additive manufacturing in their production.	Surviving Red Queen competition increasingly hinges on the evolvability of the software embedded in a firm's market offerings.	• Architecture as a largely irreversible yet preordaining DNA of systems. • Architecture as a mass-coordination device.
Proliferation of the IoT magnifies the distributedness and scale of systems.	Potential gridlock from an interplay of structural and behavioral complexity.	• Approaches for inferring evolvability distinctive to IS.

Note. IOT, Internet-of-Things; IS, information systems.

Figure 1 Alice Racing the Red Queen in this 150-Year-Old Illustration was Formative in the Conceptualization of Red Queen Competition



Source. Public domain illustration by J. Tenniel from the 1871 edition of *Through the Looking-Glass*.

advent of the GPS technology illustrates (Breseler 2015). Systems not built with evolvability as a core aspiration can then become the straitjacket of organizations that depend on them. Strategic exploitation of IT with a goal of creating even temporary market advantages requires systems to be inherently evolvable.

Yet, as Figure 2 illustrates, IS research has focused predominantly on the front end of systems' life cycle clustered around its first use and less on what follows. The costs of a system over its lifetime can be almost an order of magnitude larger² than its initial costs: The \$4 trillion spent globally on IT in 2014 alone could potentially cost another \$28 trillion before those systems are retired. Not accounting for these costs can instigate suboptimal decisions. Even small improvements in the evolvability of IT systems can handsomely pay off in Red Queen competition. This however requires attention to the post-adoption, post-continuance, life cycle-spanning evolvability of the

systems that we construct. How do we implement systems to be evolvable to match the transient pace of competitive advantage?

2.2. Embedded Software Is Making Nontechnology Industries Increasingly Indistinguishable from the IT Industry

Software is increasingly embedded in nontechnology market offerings—watches, doorlocks, toilets, soda fountains, vacuum cleaners, even light bulbs (Andreessen 2011). A modern car for example has the computing power of 50–100 laptops and millions of lines of software code (MIT Technology Review 2015). Software is becoming even more deeply intertwined with manufacturing processes in industries where nascent additive manufacturing (3D printing) is supplanting conventional subtractive manufacturing. This can simply be viewed as completing a cycle (Figure 3), the first half of which began in the dot com epoch as the digitization of atoms. Additive manufacturing represents the other half, physical materialization of bits. The implication is that nontechnology products have begun to inherit properties idiosyncratic to the software industry such as same-side and cross-side network effects, low reproduction costs, complementarities, capacity for noncoercive lock-ins, rapid releases, and a potential for asymmetric pricing similar to multisided markets. The survival of firms in non-IT industries in Red Queen competition will increasingly hinge on the evolvability of the software embedded in their market offerings. Failing to appreciate the extent to which they are rapidly becoming indistinguishable from the software industry can be costly for organizations, as illustrated by California's Department of Corrections accidentally releasing 450 violent felons in 2011, Canadian firm AELC's Therac-25 cancer therapy machines delivering overdoses of radiation, Toshiba's exit from the television business, and Target's struggle for survival after a massive security breach in 2013.

² Various industry estimates range from 500% to 700%; the real costs cannot be reliably estimated because of differences across technologies and programming languages, and confounded by systems abandoned by firms that failed to survive.

Figure 2 Lesser Studied Lifetime Costs of Information Systems can Exceed by an Order of Magnitude the More Studied Initial Costs

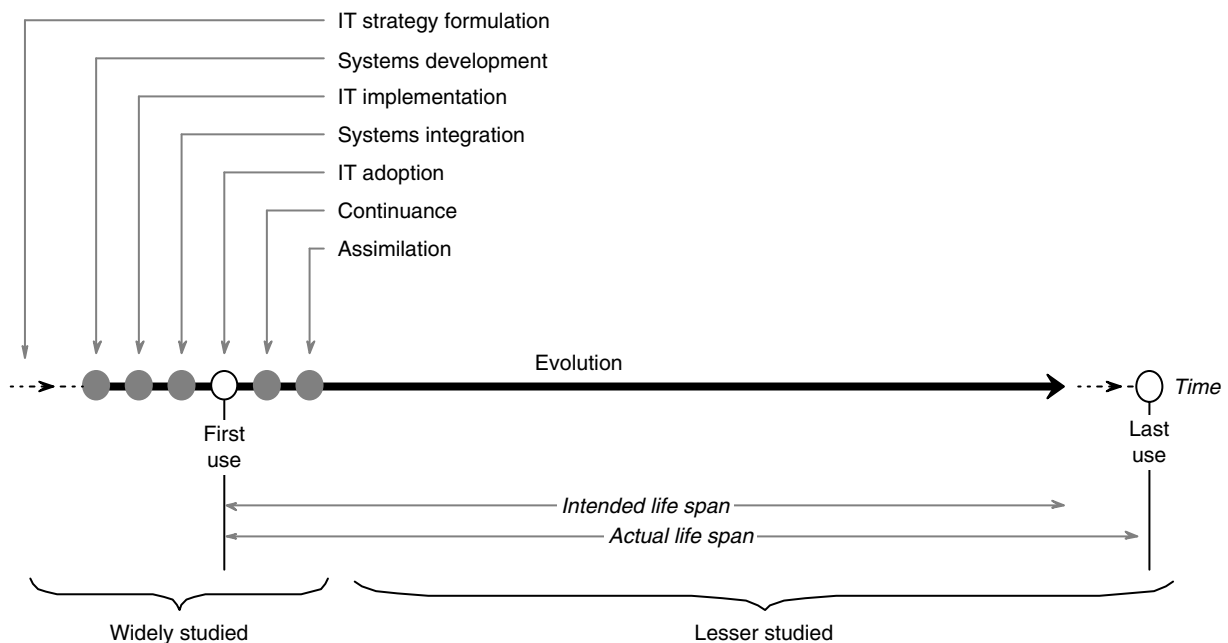
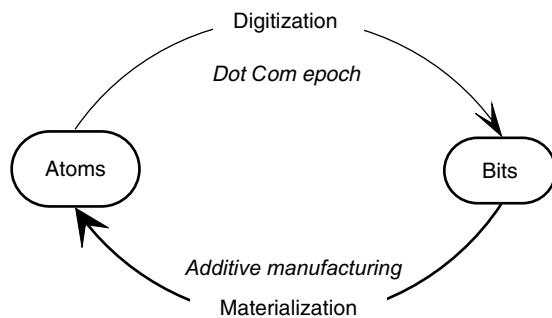


Figure 3 Materialization versus Digitization



2.3. Internet-of-Things Explodes Both Distributedness and Scale of Information Systems

The fledgling IoT will likely magnify these pressures as objects supplant devices that constitute an IS. Two hundred billion *things* are expected to be connected to the Internet in five years, including about 150 million cars, 300 million utility meters, 100 million light bulbs, and 6 billion consumer devices (Intel 2015, Press 2014). (There were 14 billion such things connected to the Internet in 2015.) But that is not the future that firms must brace for; the theoretical limit on such objects is the number of possible unique Internet Protocol (IP) addresses in the Internet protocol: 340 undecillion.³ The disruptive potential of ordinary physical objects that inexpensively communicate lies in using software to stitch together information that they gather—location, temperature, humidity, movement patterns—into con-

text. This context gives a whole different meaning to contingency theory's notion of "environment."

The implication of this is twofold. First, the lag between understanding this context and each adaptive move can exact compounding competitive penalties in Red Queen competition. The complexity of dealing with big data (Agarwal and Dhar 2014) potentially becomes even bigger. Second, the distributedness of an IoT-incorporating system can exacerbate both structural and behavioral complexity. Coordination challenges (aligning actions) can persist even if agency conflicts (aligning interests) are absent (Gulati et al. 2005). Splintered ownership of different parts of a single system can create an evolvability-retarding gridlock (Tiwana 2014, p. 211); the systemwide consequences of even a minuscule change in one part become harder to predict. How can a firm evolve a system with many moving parts if it owns only a fraction of it and it grows incomprehensibly complex in its entirety? If a firm has limited control over where the data populating its system (that is in the hands of consumers, in their households, or in the vehicles that they drive) comes from, and these data sources are in motion and susceptible to disruption, how can it possibly rely on these data to drive decision making? Multifaceted, often unpredictable interactions of behavioral complexity can compound the challenges of such structural complexity.⁴ Evolvable systems must tackle both.

³ An undecillion is a trillion, trillion, trillion.

⁴ Lines of authority are likely to be more ambiguous than in hierarchies or in contractual market relationships; and the number of parties involved (organizational scale) can make coordination among them formidable.

3. Uncharted Territory for IS Research

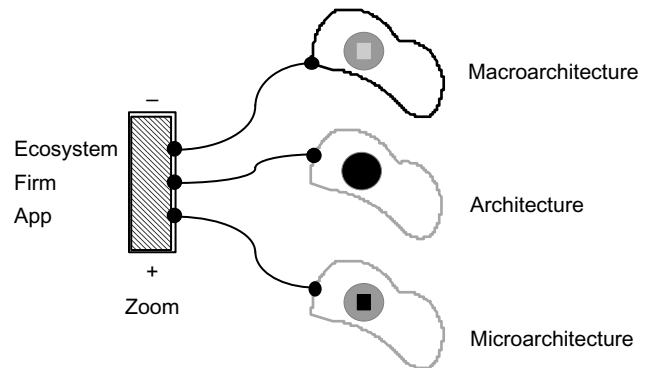
These phenomena collectively open up three uncharted spaces for IS research, all focused on the “system” as the unit of analysis. These include designing systems for disassembly rather than just integration; architecture as a mass-coordination device; and approaches for inferring systems evolvability personalized to IS research conversations.

3.1. Designing Systems for Disassembly, Not Just Integration

Considerable progress has been made in understanding systems integration and the mechanisms to facilitate it (e.g., Subramanyam et al. 2012). But systems also need the capacity for disassembly, i.e., to pull apart one subsystem without impairing the rest of the system. This requires thinking beyond the contemporary plug-and-play aspiration to ask questions about how we build systems than can unplug and still play. Facilitators of one should not be assumed to facilitate the other. Such systems must be able to readily incorporate subsystems that do not yet exist, preserve viable legacy ones, and coherently link them all without compounding complexity.

Two research opportunities stem from this challenge. First, we need to develop a deeper understanding of the dynamic reapportionment of an IT system between apps and infrastructure over time. The distinction between apps and infrastructure is more widely used in practice than in theory development (Agarwal and Sambamurthy 2002, Tiwana and Kim 2015). Infrastructure is the substrate on which apps are built; necessary but rarely a differentiator. Contemporary infrastructure spans beyond our historical conception of IT infrastructure, e.g., platforms such as iOS and their application programming interfaces (APIs) that apps can use to access it (Tiwana et al. 2010). Research questions at the discrete “system” level include how this boundary is initially chosen, how it evolves over time,⁵ and its evolutionary consequences. The endogenous choice of the boundary where a discrete system ends and infrastructure begins is associated with an evolvability trade-off: the functionality of an app implemented via infrastructure can economize costs and potentially reduce implementation complexity but may also simultaneously constrain its evolution. For example, the more a mobile app exploits the Android platform’s infrastructure, the less the code its developer must write from scratch but the more dependent it becomes on the platform. In this sense, the app-infrastructure boundary choice can be viewed as an attention-conserving,

Figure 4 Systems Architecture Is Conceptually Agnostic to Unit of Analysis



bounded-rationality coping mechanism with yet-to-be-understood consequences for systems evolvability.⁶ It is potentially attention-conserving because it can reduce the internal complexity and size of the focal system to a point where a developer might be able to grasp it in its entirety.

A second research opportunity arising from the notion of disassembly is the potential to explain adaptation in a manner unique to the IS discipline. Strategy and organization theory for example emphasize the behavioral impediments to organizational adaptation, and the structural impediments implicated in their literature are about *organizational* design (e.g., centralization, formalization, and decision rights). The evolvability of an organization’s IS increasingly constrains organizational adaptation. An evolvable systems perspective can contribute distinctive insights into how the *technological* structure of a system enables and constrains organizational adaptation, and into its interplay with previously studied organizational properties. Such a system-level approach is conceptually agnostic to the organizational unit of analysis; it can zoom-in to a micro level (e.g., a product) or zoom-out to a macro level (e.g., an ecosystem or platform). This is illustrated in Figure 4.⁷ This can potentially add a novel perspective to the enduring Lawrence and Lorsch (1967) integration-differentiation tension that organizations of all shapes and sizes struggle with.

⁶ The question of boundaries also demands a sharper definition of what constitutes a “system.” We believe that Simon’s (1962) idea of defining a boundary of a discrete system as one with strong interactions within it and weak interactions outside is a legitimate approach for defining this boundary.

⁷ As we zoom out, these levels range from the microarchitecture of an individual app, architecture of a firm’s IT portfolio, to the macroarchitecture of an ecosystem. We must not however assume (e.g., Tiwana et al. 2010) that architectural properties desirable at one level of zoom are also desirable at others in Red Queen competition.

⁵ For example, over time, functionality originating in an app can become part of infrastructure and elements of infrastructure can move into an app. For a narrower technical description of this phenomenon in consumer-grade platforms, see Tiwana (2014, p. 225).

3.2. Architecture Is the DNA of Systems, Unchangeable Yet Preordaining

Architectural choices are frequently irreversible, endogenous choices that can preordain systems' evolvability by opening and foreclosing viable evolutionary paths (Tiwana 2014, p. 97). Organizations often relegate them as technical decisions despite their yet-to-be-understood strategic consequences. Consider three examples: Facebook's incredibly high bandwidth costs and Google's need to run about half a million servers just to power search contrast starkly with Skype's almost infinite ability to inexpensively scale; all artifacts of their early architectural choices. Facebook and Google's architectures are host-based architectures that concentrate most functionality on the server side; Skype uses a peer-to-peer architecture. These choices are reversible in theory but the costs are likely prohibitive in practice.

The to-be-explored opportunity for IS research is conceptualizing IT architecture as a plausible coordination device among autonomous entities. Coordination devices derived from organization theory appear unscalable to the fluid coalitions of many autonomous firms (not to mention consumers) that can exceed even the largest interfirm networks by *over an order of magnitude*.⁸ The IoT will likely amplify this challenge. To the extent that other disciplines tend to focus primarily on *organizational* structure, control, and incentives as coordination devices, we cannot rely solely on the knowledge generated there. Satisficing levels of coordination are observable in organizations of precisely the sort where Ouchi (1979) deemed organization theory's traditional devices infeasible. How does IT architecture—at any level in Figure 4—enable a motley of small firms to replicate the coordination advantages of a hierarchy and the efficiencies of a market? What determines whether architectural choices at the system level become a springboard or an iron cage for new offerings? Understudied also is the interplay among architectural choices at different levels of a system's hierarchy in Figure 4, and its strategic consequences. (See Bresnahan and Greenstein 2014 for a recent resurrection of systems hierarchy.) Further, even if architecture could align actions, it cannot align incentives between autonomous parties (firms, departments); how can that void be filled by IT governance?

We view architecture as an opportunity for bringing back systems into IS thinking. Insights from two theoretical perspectives offer potentially useful building blocks. First, we draw attention to the lens of modularity that originated in 1972 in IS (Parnas 1972), inspired by Simon's (1962) notion of nearly-decomposable systems. We call for a thoughtful reduction of two conceptual

homonyms, architecture and modularity; simplicity in our theoretical building blocks can be potent in grasping the complexity of the systems that we confront.⁹ The second is leveraging real options thinking (Fichman 2004, Gamba and Fusari 2009) to theorize how architectural choices can endogenously embed different types of future flexibilities that might be differentially valuable under qualitatively different types of uncertainties unique to Red Queen competition among rival systems.

Tensions worthy of research attention for Red Queen competition include how architectural choices might facilitate systems integration but also create asset-specificity-like lock-ins among cooperating firms; how they might shift the balance of power among them; and how they might exacerbate mutual obliviousness. Theoretically advancing our understanding of systems architectures can also help separate signal from noise in our sensemaking of new IT trends.¹⁰

3.3. Inferring Systems Evolvability

How do we gauge systems evolvability? Evolvability cannot be directly observed but it can be inferred through changes over time in some evolution-oriented outcomes. They span the short- and long-term as well as strategic and operational outcomes (Tiwana 2014, p. 164; Tiwana et al. 2010).¹¹ Figure 5 illustrates some potentially relevant outcomes, using the system as the unit of analysis. Explicitly thinking in terms of systems evolvability can thus expand the repertoire of outcomes that we attempt to explain, nomologically and temporally downstream to the ones that have traditionally dominated IS research (italicized in Figure 5).

In Red Queen competition, the performance of one system is always *relative* to a cohort of rival systems. This cohort must be identified explicitly to avoid erroneous apples-to-oranges comparisons in using any evolvability proxies. (A payroll system, an email system, and Uber's matching platform would belong to

⁹ A first step is disentangling modularity as a systems property from modularization as a deliberate process, and from their upstream antecedents and downstream consequences. A second step is to eschew the assertion of modularity as desirable at all levels in Figure 4, at least until we better understand its myriad consequences.

¹⁰ Consider the fact that cloud computing in vogue today is architecturally identical to host-based computing popular in the 1970s in that a thin, relatively-dumb client (today, a tablet or smartphone) is merely an interface to functionality heavily concentrated on the server side. The only difference is the reason for its resurrection: a confluence of TCP/IP ubiquity and declining costs of computational processing. Public and private APIs fueling platforms like iOS and Android are simply hooks into such host-based architectures in an object-oriented paradigm.

¹¹ Dynamics in the short-run might offer little insight into those in the long run (Simon and Ando 1961). Simon and Ando (1961) suggest that models for explaining the latter can sometimes be much *simpler* if near-decomposability characterizes a system; again a direct connection to architecture.

⁸ To illustrate, the iOS ecosystem's 300,000 app developers (who have produced 1.5 million apps) versus Li-and-Fung's network of 8,000 firms.

Figure 5 Some Proxies for Systems Evolvability

Type of outcome	Time frame	
	Short-term	Long-term
Strategic	Plasticity <i>Alignment, assimilation</i>	Mortality Market Durability Mutation Pleiotropy <i>Tobin's Q, business value</i>
Operational	Boundary Morphing <i>Performance, adoption, use</i>	Interoperability <i>IT payoff (e.g., ROI, ROA)</i>

Note. Italicized variables illustrate historically popular nonevolutionary outcomes.

different cohorts.) This cohort defines what constitutes a sufficiently long observation interval for any such outcome (and short- versus long-term in Figure 5), none of which should be used in isolation to strongly infer the absence of systems evolvability. At least some argue—in a Red Queen ethos—that sheer intensity is far more consequential than quality of evolutionary steps (Eisenhardt and Tabrizi 1995, Nelson 1995). Perhaps the most meaningful long-term outcome of evolvability is a system’s *mortality*. Its meaningful use in Red Queen competition demands attention to two empirical nuances. First, the marker for a system’s mortality—whether it is termination of updates or its last use in Figure 2—must be explicit. System mortality and survival are not simply two sides of the same coin, unlike previously assumed (e.g., Tiwana et al. 2010). Second, to mitigate prematurely declaring a slower-evolving system as dead, a cohort of rival systems should be the basis for the observation window. The *market durability* of a system (Tiwana et al. 2010), conceptualized in Red Queen competition as the proportion of adopters who subsequently remain active users vis-à-vis its cohort of rival systems, is another competitive outcome. By directly observing usage, we can tangibly measure the psychometrically-elusive notion of competitive advantage *sustainability*. Similarly, *boundary morphing*, representing a temporal change in the proportion of system functionality implemented within it versus “outsourced” from another firm’s system via APIs can offer insights into temporal *shifts* beyond static boundary *choice* in transaction cost economics. Contemporary systems do not just affect firm boundaries, they define them (see also §3.1). A biological analog is system *mutation* (Tiwana et al. 2010), defined as the reproduction of a system in an unrelated environment or application where it competes with a different cohort of rival systems. For example NextStepOS mutated into OSX, iOS, and then watchOS

over the course of 25 years. Observing NextStepOS over say five years would erroneously conclude the absence of mutation; this is why the choice of observation interval is critical. “Forking” in open source projects is another manifestation of mutation, with potentially *pleiotropic* consequences. Other evolvability proxies include the capacity to function with other organizations’ systems even if they were not designed to work together (*interoperability*¹²), the ease of making corrective repairs (*maintainability*), and a system’s capacity to serve its original purpose and a new function (*plasticity*).¹³

4. Conclusion

The increasing prevalence of Red Queen competition is enlarging the premium on evolvable IS. Such evolvability has yet to receive much attention in IS. An inability to construct evolvable systems will increasingly become a competitive disability. Nascent IT trends coupled with ephemeral market advantages imply an industry-agnostic reward for such evolvability: survival. We highlighted uncharted territory where IS researchers can contribute insights into phenomena of enduring interest but elusive to other disciplines.

We speculate that evolution manifested over half a century in nonsoftware industries and over three to four centuries in some natural sciences can be observed in about a decade in the software industry. Such *time dilation* offers potential opportunities to use software systems as a laboratory for studying evolutionary phenomenon generalizable to other disciplines.

A powerful analogy potentially exists between evolvable systems and machine learning. If algorithms can be constructed to learn and modify their behaviors based on the data that they process, what will it take for “systems” to be able to do the same? To paraphrase Chuck Palahniuk (2005) in *Fight Club*, on a long enough timeline, the survival rate for every system drops to zero. Constructing systems for evolvability can lengthen that timeline, with critical implications for the fate of organizations that depend on them.

References

Agarwal R, Dhar V (2014) Big data, data science, and analytics: The opportunity and challenge for IS research. *Inform. Systems Res.* 25(3):443–448.
Agarwal R, Sambamurthy V (2002) Principles and models for organizing the IT function. *MIS Quart. Executive* 1(1):1–16.

¹² The healthcare sector provides a compelling example of how a lack of interoperability is becoming a competitive liability. Electronic Health Record systems were not designed with interoperability in mind. Today, healthcare organizations face financial penalties and struggle to comply with the new regulations that demand electronic exchange of patient health data across organizational boundaries.

¹³ We do not view scalability, robustness, and resilience as evolutionary outcomes.

- Andreessen M (2011) Why software is eating the world. *Wall Street J.* (August 20). Accessed August 19, 2015, <http://www.wsj.com/articles/SB10001424053111903480904576512250915629460>.
- Barnett W (2008) *The Red Queen Among Organizations: How Competitiveness Evolves* (Princeton University Press, Princeton, NJ).
- Breseler S (2015) Why 40-year-old tech is still running America's air traffic control. *Wired*. Accessed August 14, 2015, <http://wired.com/2015/02/air-traffic-control>.
- Bresnahan T, Greenstein S (2014) Mobile computing: The next platform rivalry. *Amer. Econom. Rev.* 104(5):475–480.
- Carroll L (1917 reprint) *Through the Looking Glass* (Rand McNally and Co., Chicago).
- D'Aveni R, Dagnino G, Smith K (2010) The age of temporary advantage. *Strategic Management J.* 31(13):1371–1385.
- de Weck O, Roos D, Magee C (2011) *Engineering Systems* (MIT Press, Cambridge, MA).
- Eisenhardt K, Tabrizi B (1995) Accelerating adaptive processes: Product innovation in the global computing industry. *Admin. Sci. Quart.* 40(1):84–110.
- Fichman R (2004) Real options and IT platform adoption: Implications for theory and practice. *Inform. Systems Res.* 15(2):132–154.
- Gamba A, Fusari N (2009) Valuing modularity as a real option. *Management Sci.* 55(11):1877–1896.
- Gulati R, Lawrence P, Puranam P (2005) Adaptation in vertical relationships: Beyond incentive conflict. *Strategic Management J.* 26(5):415–440.
- Intel (2015) Internet of things. Accessed August 12, 2015, <http://intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html>.
- Lawrence P, Lorsch J (1967) Differentiation and integration in complex organizations. *Admin. Sci. Quart.* 12(1):1–47.
- MIT Technology Review (2015) Rebooting the automobile. *MIT Tech. Rev.* (July/August):54. Accessed August 19, 2015, <http://www.technologyreview.com/featuredstory/538466/rebooting-the-automobile>.
- Nelson R (1995) Recent evolutionary theorizing about economic change. *J. Econom. Literature* 33(1):48–90.
- Ouchi W (1979) A conceptual framework for the design of organizational control mechanisms. *Management Sci.* 25(9):833–848.
- Palahniuk C (2005) *Fight Club: A Novel* (WW Norton, New York).
- Parnas D (1972) On the criteria to be used in decomposing systems into modules. *Comm. ACM* 15(9):1053–1058.
- Press G (2014) Internet of things by the numbers: Market estimates and forecasts. *Forbes* (August 22). Accessed August 12, 2015, <http://www.forbes.com/sites/gilpress/2014/08/22/internet-of-things-by-the-numbers-market-estimates-and-forecasts/>.
- Simon H (1962) The architecture of complexity. *Proc. Amer. Philos. Soc.* 106(6):467–482.
- Simon H, Ando A (1961) Aggregation of variables in dynamic systems. *Econometrica* 29(2):111–138.
- Stenseth N, Smith J (1984) Coevolution in ecosystems: Red queen evolution or stasis. *Evolution* 38(4):870–880.
- Subramanyam R, Ramasubbu N, Krishnan M (2012) In search of efficient flexibility: Effects of software component granularity on development effort, defects, and customization effort. *Inform. Systems Res.* 23(3):787–803.
- Tiwana A (2014) *Platform Ecosystems* (Morgan Kauffmann, Waltham, MA).
- Tiwana A, Kim S (2015) Discriminating IT governance. *Inform. Systems Res.* Forthcoming.
- Tiwana A, Konsynski B, Bush A (2010) Platform evolution: Coevolution of architecture, governance, and environmental dynamics. *Inform. Systems Res.* 21(4):675–687.