Technological Forecasting & Social Change 117 (2017) 151–159

Contents lists available at ScienceDirect

Technological Forecasting & Social Change

When the sky is the limit on scale: From temporal to multiplicative scaling in process-based technologies

Mark Lehrer [a,*], Preeta M. Banerjee [b], I. Kim Wang [a]

[a] Strategy and International Business, Sawyer School of Management, Suffolk University, 120 Tremont St., Boston, MA 02108, USA
[b] Deloitte Development LLC, 1000 Winter St. Suite 2000, Waltham, MA 02451, USA

ARTICLE INFO

Article history:
Received 29 April 2016
Received in revised form 8 December 2016
Accepted 12 December 2016
Available online 5 January 2017

Keywords:
Process-based technologies
Scale
Design
Biotechnology

ABSTRACT

The design evolution of two important process-based technologies, PCR DNA amplification and ERP software, was punctuated by discrete leaps in scale. From comparison of these technologies we distill a stage model centering on the phenomenon of increasing scale while clarifying just what the concept of scale means in the context of process-based technologies. Process-based technologies turn out to be distinctive because of the *temporal* aspect of scaling; although scaling up usually refers to spatial dimensions of scale, this research highlights the temporal dimension to scale. Temporal scaling can be complemented by multiplicative scaling, a design innovation enabling multiple processes to be performed in parallel. After highlighting different patterns of innovation from those that characterize manufactured products as conveyed by classic product-process lifecycle models, we reconcile our stage model with these classic lifecycle models: although the sequence of innovation phases is different, the overall evolution of the underlying economic logic motivating technology developers is actually rather similar.

1. Introduction

A major topic in technology management concerns the challenge of increasing the volume or scale of an innovation. For example, dominant design models of the product life cycle usually characterize the mature phases of the life cycle as one in which the innovative focus is on process innovation for the sake of greater large-scale production efficiency. From a slightly different perspective Sahal (1981, 1985) coined the expression of "learning by scaling" to describe the learning required to master the challenge of engineering larger-scale versions of products or systems. The need to scale up the size of technology products periodically leads to a sequence of multiple dominant designs over time (Frenken and Leydesdorff, 2000). As a general observation, many technology entrepreneurs and even many high-tech regions often develop innovative technology products but struggle with the challenge of scaling up production of such products once they have overcome obstacles to commercialization in their original application domain (Florida and Kenney, 1990).

However, the vast bulk of prior innovation research on the topics of volume and scale applies mainly to manufactured products, leaving the question open as to how issues of volume and scale play out in non-manufacturing domains, such as services (Barras, 1986) or – as examined here – in process-based technologies. It is important from the start not to confuse *process-based technologies* – by which we mean technologies based on a sequence of procedural steps that need to be performed – with the concept of *process innovation* used to describe improvements in the production of (mainly manufactured) items (OECD, 1997). Although the concept of process-based technologies has never been the subject of any dedicated study, this category of technologies is not new and appears to be generally understood by scholars. For example, several researchers specifically characterize nanotechnology as process-based (Linton and Walsh, 2008; Maine et al., 2012) and emphasize the process-based nature of nanotechnology as crucial for understanding innovation patterns specific to this technology.

The empirical focus of this study is on two important process-based technologies, polymerase chain reaction (PCR) DNA amplification and enterprise resource planning (ERP) software, whose development and design evolution was punctuated by multiple discrete leaps in scale. From comparison of these technologies we distill a stage model centering on the phenomenon of increasing scale while clarifying just what the concept of scale means in the context of process-based technologies. Process-based technologies turn out to be distinctive because of the *temporal* aspect of scaling. Whereas the scaling of *products* primarily alters their *size*, the scaling of *processes* primarily alters the *speed* with which they can be performed.

One of the salient performance dimensions of process-based technologies, in other words, is time compression. To make the matter less

* Corresponding author.
E-mail addresses: marklehrer@gmail.com (M. Lehrer), pbanerjee@deloitte.com (P.M. Banerjee), iwang@suffolk.edu (I.K. Wang).

http://dx.doi.org/10.1016/j.techfore.2016.12.006
0040-1625/

abstract, suppose the amount of time needed to replicate a strand of DNA is initially 48 hours. A technological design improvement reducing the time involved to 6 hours results in a temporal scaling-up factor of 8: one can perform the DNA amplification process 8 times as often in a given span of time. This is temporal scaling. Throughput is increased by accelerating the performance speeds of the underlying technological process.

As the title of our paper suggests, our analysis discloses a pattern of design evolution from a temporal stage of scaling up to a "multiplicative" stage of scaling up. In contrast to temporal scaling (enabling an order-of-magnitude reduction in the time needed to perform a given technology process), "multiplicative" scaling involve a configurational method for increasing throughput. Again, the DNA example makes the basic principle plain. As an alternative to temporal scaling, let us imagine a design innovation in DNA amplification products that still require 48 h to perform the process but are able to duplicate the DNA in 8 samples simultaneously. In terms of overall throughput, a "multiplicative" design improvement of this kind yields a statistically equivalent outcome to temporal scaling-up by the same factor of 8 (Fig. 1). In terms of overall efficiency, multiplicative scaling can have the same net throughput effect as temporal scaling.

The two approaches are not mutually exclusive, of course. Temporal scaling can be complemented by multiplicative scaling. In fact, our research on two highly disparate process-based technologies is suggestive of a stage model involving three basic stages: 1) refinement of the basic process design; 2) temporal scaling; 3) multiplicative scaling.

Beyond just this simple stage model, the case studies below shed light on a number of issues related to the concept of "learning by scaling" (Sahal, 1985). For one thing, the relationship between product and process innovation differs from that in product-process lifecycle models predicated on a phase of product innovation giving way to a primary emphasis on process innovation (Abernathy and Utterback, 1978). For another, in wide-application technologies like ERP and PCR the scaling up process can accompany learning on both the supply and demand side. Most prior research on learning by scaling focuses on supply-side phenomena (Narasimalu and Funk, 2011; Sahal, 1985; Slayton and Spinardi, 2016). Our study shows how the nexus between scaling and learning operates on the demand side as well: the appetite of ERP and PCR users for greater scale was stimulated by user learning about new applications of these technologies requiring greater scale.

Discussion proceeds as follows. Section 2 surveys relevant issues in prior literature on scaling and dominant designs. Section 3 explains the methodology of the study and the selection of PCR and ERP as a logical pair of technologies for study and comparison. Sections 4 and 5 contain brief case studies on the process-based technologies of PCR and ERP and on the design evolution of commercialized processes and products embodying these technologies. Section 6 discusses the implications of the case studies and our derived stage model. As in the literature on dominant designs, the focus of the analysis is on the nexus between design evolution and industry dynamics (Murmann and Frenken, 2006). The discussion of Section 6 fleshes out the interrelated issues of temporal scaling, the seeming insatiability of demand for greater scale, and the relationship between product and process innovation. Section 7 concludes with limitations and avenues for further research.

2. Theoretical background

Leaps in innovation resulting from new dominant designs often enable a change in scale of products or of production volumes (Abernathy and Utterback, 1978; Suarez and Utterback, 1995; Tushman and Murmann, 2003). With numerous variations this pattern has been shown to hold in many different industries, e.g., aircraft for civil aviation (Frenken and Leydesdorff, 2000; Slayton and Spinardi, 2016); the typewriter, calculator, TV, vacuum tube, and transistor industries (Suarez and Utterback, 1995) and the cement, container glass, flat glass, and minicomputer industries (Anderson and Tushman, 1990).

Sahal (1981) first underlined how dominant designs expand the scale of a technology. Often scale means physical size. The relevant scaling can be a "scaling-up" of wing span and fuselage length for civil aircraft (Frenken and Leydesdorff, 2000), or a "scaling-down" of vacuum tubes, transistors, and integrated circuits for computers (Cabral and Leiblein, 2001). Sahal (1981) and Slayton and Spinardi (2016) note that scaling can occur in dimensions beyond just overall product size, however, underlining especially the challenge of scaling up key components of products. The experience of the hard disk industry exemplifies both spatial and temporal scaling. The adoption of magnetic polarization increased the areal density, i.e., the volume of information stored on a given length of track of the hard disk drive (Christensen, 1997); meanwhile, the basic measurement unit of microprocessor clock speed advanced from MHz (megahertz) in 1990s to GHz (gigahertz) in early 2000 when semiconductor manufacturers deployed a narrower line width (Cabral and Leiblein, 2001). Although an exhaustive discussion of how the term "scale" has been used is beyond the purview of

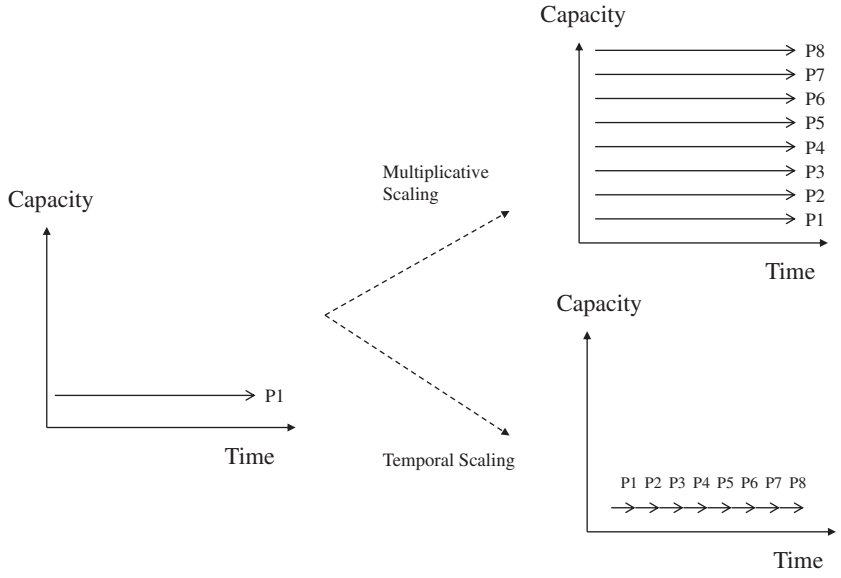

Fig. 1. Temporal & multiplicative scaling.

this research, such patterns of innovation suffice to indicate the multiple dimensions of scale characteristics of products.

Previous scholarship has also shed light on the mechanisms leading to a change of scale. Anderson and Tushman (1990) develop a cyclical model of technological change in which one cycle begins with competing product designs that converge on a dominant design, followed by a sequence of incremental changes. These changes ultimately lead to technological discontinuity and another round of product design competition; Slayton and Spinardi (2016) conceptualize such discontinuities as sociotechnical transitions. In each cycle, rivalry between product designs contributes to changes in scale. A hierarchical view of design principles helps explain how dominant designs at different system levels enable such leaps (Murmann and Frenken, 2006; Sahal, 1981; Tushman and Murmann, 2003). Such a view is again illustrated by the development of the hard disk drive (Christensen et al., 1998). Two architectural innovations (the Winchester architecture and intelligent interface electronics) and two component innovations (the under-spindle pancake motor and the voice coil actuator motor) occurring between 1973 and 1986 defined today's hard disk drives. The interplay of innovations at different system levels shrank the disk from 14 to 1.8 in.

A common denominator among different kinds of changes in scale is efficiency improvement leading to higher user utility and thus to greater market acceptance. The market often withholds the adoption of an innovation until a certain cost and/or efficiency threshold is met (Sahal, 1981). For instance, connections to the power grid multiplied when the cost of electricity decreased (Sahal, 1981). In general, dominant designs inaugurate the convergence of supply and demand at a sharply increased level (Benner and Tripsas, 2012; Murmann and Frenken, 2006). With regard to the issue of scaling in particular, one point that is underdeveloped in prior research is the conflation of efficiencies resulting from the scale of production operations (e.g., factory size) and efficiencies resulting from the scaling-up or scaling-down of products (e.g. product performance gains) as seen in aircraft (Sahal, 1981) and disk drives (Christensen et al., 1998).

Prior studies on scaling focus mainly on product innovation for which some form of the product-process industry lifecycle model is relevant (Abernathy and Utterback, 1978; Christensen et al., 1998; Frenken and Leydesdorff, 2000; Murmann and Frenken, 2006; Tushman and Murmann, 2003). Nonetheless, a dichotomous classification of product vs. process innovation overlooks certain technologies that are process-based and inseparable from the product itself. A novel R&D process enabled the scaling-up of the volume of commercial bioreactors, which for decades had been plagued by the inability to scale up (Linton and Walsh, 2008). In many very important technologies – such as nanotechnology – product and process innovations are intertwined (Linton and Walsh, 2008; Maine et al., 2012). Many of the most important commercialized outputs of (process-based) innovations in nanotechnology are products using nanotechnology components like carbon nanotubes and fullerenes (Maine and Garnsey, 2006; Youtie et al., 2008).

To summarize, the interplay of scale and dominant design has received ample attention in prior research on manufactured technology products. However, our knowledge is more limited of scaling, and of the relevant dimensions of scaling, in process-based technologies. Given their evident importance, the design evolution of process-based technologies merits investigation.

3. Methodology: sample selection, framework of analysis, and data collection

The two process-based technologies studied here were polymerase chain reaction (PCR) DNA amplification, a core technique in biotechnology, and enterprise resource planning (ERP) software, an IT-based technology. Both PCR and ERP were at first idiosyncratic, specialized technologies whose vast commercial potential became apparent only after decades. Although PCR and ERP can be termed general-purpose technologies from the standpoint of widening applications over time (Gambardella and McGahan, 2010; Lehrer et al., 2016), the following analysis focuses less on the generality of applications per se and instead more on the phenomenon of increasing scale in design on the supply side and, in tandem, the seeming insatiability of users on the demand side, as ever more intensive applications of these process-based technologies were discovered.

Predicated on a research design involving most-different cases (Gerring, 2007: 139–145), the two technologies were selected in part because of their disparity: IT (ERP) vs. biotechnology (PCR), engineering (ERP) vs. science (PCR), software (ERP) vs. hardware (PCR), etc. Gerring (2007) cites studies in which only one single independent variable was the same (all the others being different), the idea being to provide evidence of some important generalizable impact of this key variable.

Similarly, in our sample the common independent variable is the process-oriented nature of the core technology: these "most-different cases" are meant to yield at least some face evidence for the generalizability of findings concerning process-based technologies. Process-based technologies can be defined commonsensically as technologies that 1) consist of *a procedure for transforming certain inputs into certain outputs*, 2) involve a certain number and sequence of *steps* to perform, and 3) take *a certain amount of time to perform* (i.e., taking up time rather than just space). Previously the term has certainly been used in a technical sense (Linton and Walsh, 2008; Maine et al., 2012) but to our knowledge never explicitly defined.

ERP and PCR technologies exemplified two contrasting paths by which a specialized process evolves into a commercializable technology via design innovations. As summarized in Table 1, ERP originated as an exploit of engineering, whereas PCR originated in the R&D laboratory. The developers of ERP software took a specialized and idiosyncratic software development process and re-engineered it into a one-size-fits-all design of standardized software. Parallels in the realm of more tangible products can be seen in 19th-century machine tools (Rosenberg, 1963) such as the development of the so-called universal milling machine (1861) and universal grinding machine (1868). Similar patterns characterize, much more recently, the development of universal control systems (Thoma, 2009). In such a process of "technological convergence" (Rosenberg, 1963, 1976), commercial developers engineer a highly versatile, "universal" embodiment of an initially non-standardized product or process (Pavitt, 2003; Rosenberg, 1963).

In contrast, PCR DNA amplification was a science-based technological process. PCR began as a laboratory discovery that could be performed only by trained scientists. Such a pattern is common among technologies developed in the R&D laboratory such as lasers, nanotechnology-based materials (Maine and Garnsey, 2006) and mathematical algorithms used in software (Gambardella and Giarratana, 2013). In such cases, the initial gap between the technology and marketable products is substantial and technology providers have to design "user-friendly" embodiments of the technology in order to make inroads on commercialization (Gambardella and Giarratana, 2013).

Despite differences in genesis, one common feature of development facilitated comparison. Both PCR and ERP featured a "real time" design

Table 1
Two contrasting origins of process-based technologies.

	Engineering-based technology	Science-based technology
Origin	A pre-existing process utilized for a specialized purpose	Invention or scientific discovery, e.g., in the R&D laboratory
Basic design challenge	Engineering of a generic version of the technology for broader applications	Embodiment of the technology in commercializable and "user-friendly" products
Examples	ERP software; machine tools (Rosenberg, 1963); continuous chemical processing (Pavitt, 2003)	PCR DNA amplification; software algorithms (Gambardella and Giarratana, 2013); nanotechnology (Maine et al., 2012)

phase. The first bona fide commercialized ERP software product, R/2, stood for Realtime/2, and while one important version of PCR technology, QRT-PCR, stood for Quantitative Real Time-PCR. As an initial means of organizing the data, we built timelines of technological and design evolution featuring three phases: a pre-real time phase #1, a real time phase #2, and a post-real time phase #3.

Our research began with expert interviews. In researching the design evolution of PCR we conducted nine interviews in the US life sciences sector. The interviewees were product development managers at biotech firms, lab scientists, and financial experts in the life sciences. Attendance at two different BIO International events enabled us to conduct brief interviews with a half dozen other participants. On ERP development we conducted 23 interviews within the German IT industry, including with directors of two different Fraunhofer Institutes specializing in software, managers of three IT companies in Germany (IBM Germany, Nemetschek and SAP), and participants at a major customer event of SAP and the meeting of a German IT industry association.

We followed up our interviews with secondary research. On PCR's development we consulted major published sources, notably Rabinow (1996) and Mullis (1990). On ERP we consulted books devoted to the history of SAP, the firm that essentially invented ERP software and has been the leading provider of the technology since its inception (Meissner, 1997; Plattner, 2000). A key technique of data analysis was grounded theory (Glaser and Strauss, 1967) in which theoretical constructs are derived in iterative comparisons of the data with pre-existing theoretical constructs. Key concepts derived from comparison of the PCR and ERP cases were temporal scaling, which can be defined as an order-of-magnitude compression in the time needed to perform a technology process, and multiplicative scaling, a method for achieving an order-of-magnitude increase in throughput by performing a given technology process on multiple batches simultaneously as opposed to one batch at a time. A strength of grounded theory underlined by Glaser and Strauss (1967) lies in its internal validity, since theoretical constructs are derived from the data under analysis. As for external validity, it has become common in more recent discussions of grounded theory to emphasize concerns of transferability (Gasson, 2004). Acting on such concerns, analysis of the PCR and ERP cases below is followed by consideration of how the concepts of temporal and multiplicative scaling mesh with patterns observed in established research based on the classic lifecycle models of technology development in the tradition of Abernathy and Utterback (1978).

Based on the brief case studies below, we derive a simple stage model of technology development roughly following the segmentation of events into: 1) an initial phase of process refinement; 2) a "real-time" design phase of temporal scaling; 3) a post-real-time design phase focusing on greater processing volumes and hence multiplicative scaling. Repeated design improvements resulted in the ability of user firms to perform ERP and PCR processes more and more rapidly over time and thereby to integrate them into daily operations of ever greater scale. Although obvious in retrospect, the nexus between deployment speed and scaling up was by no means predictable in advance and constitutes one of the central empirical findings. By subsequently showing that the underlying patterns of technology development broadly resemble those identified in many studies of the product-process lifecycle (Murmann and Frenken, 2006), our analysis endeavors to close the gap between spatial (product-based) and temporal (process-based) conceptions of scaling up.

4. Emergence and development of PCR DNA amplification

4.1. Overview

One pervasive process utilized across a wide range of fields and applications of biotechnology is the replication of DNA, technically called "DNA amplification." Three procedural steps define the PCR DNA amplification process. The first step uses heat to unwind the targeted DNA strands in the genetic material. The second step is to anneal the single-stranded DNA. The third is to apply a special enzyme called a polymerase to reconstitute double helixes from both strands, leading to a doubling of the target DNA. As documented below, PCR became embodied in a sequence of dominant designs that delivered first reliability, then higher speeds, and then finally vastly increased volumes. The organization of findings reflects this basic sequence of changing design foci.

4.2. Initial design phase: process refinement

In the 1980s, Cetus Corporation's Kary Mullis envisioned the use of heating-cooling cycles in conjunction with a polymerase to replicate targeted strands of DNA. Repetition of the process resulted in the target DNA being doubled, quadrupled, and eventually "exponentially amplified." Once other scientists at the firm refined procedures to perform the "chain reaction" in a reliable fashion, Cetus possessed an invaluable process for generating large quantities of the specific genetic material that biotech scientists and others sought. Cetus patented this process and began selling licenses.

During its first decade, PCR was a reliable but still complicated process within the research community. The patented PCR technique was painstaking and slow, requiring manual transfer between water baths at different temperatures and the extraction of enzymes from bacteria. Interviews with people in the industry illuminate why it remained a niche technology at first: "PCR was too complex to operate … requiring a complex mix of reagents, primers, chemistries, and thermal cycling … all too complex to operate outside research labs." PCR was costly to perform because of the time and effort required to conduct. A run of 30–35 replications cycles typically involved a two-day marathon and a cost of $8000 to $12,000.

In terms of applications, PCR first got traction as a core technology in academic research and criminal forensics labs as a method for detecting and amplifying DNA when samples volumes were low, i.e. when there was insufficient DNA or when a researcher wanted to take a closer look at a particular fragment of DNA like a specific gene fragment or mutation. Yet Cetus was commercially puzzled about what to do with its PCR technology. Cetus had been founded to produce biotechnology *products*, not laboratory processes like PCR. Many Cetus executives urged discontinuation of the PCR project as lying outside its commercial comfort zone (Rabinow, 1996).

4.3. Product innovations for increased processing speed

Cetus perceived that the ability to automate the PCR process was critical to lowering the operating cost and thus diffusing the technology. One significant bottleneck was the fact that each heating cycle of 95 degrees rendered the polymerase inoperative, requiring a change of chemicals for each successive cycle. Thus, Cetus needed a polymerase enzyme that could withstand high temperatures and multiple cycles. The discovery of such a polymerase in the hot springs of Yellowstone Park formed the basis of a commercially valuable product, the Taq polymerase (Fore et al., 2006). The Taq product set the stage for use of thermal cyclers, machines that automated the alternate heating and cooling cycles of DNA. Automation using Taq and thermal cyclers enabled scientists to conduct a full 35 cycle run in just 2 hours rather than two days.

Taq polymerase and thermal cyclers were crucial design breakthroughs facilitating temporal scaling of the PCR process and were, at the same time, easy-to-commercialize *products*. Roche, which acquired Cetus' patents in 1991, quickly realized that such tangible products were also the key to resolving the commercialization quandary surrounding PCR. Recognizing that it was cumbersome to sell and monitor the use of PCR licenses, Roche simply bundled the license for the PCR process in the equipment. While many scientists groaned under the high cost of equipment and supplies, the low transaction costs and automatic legal protection afforded by the Roche arrangements allowed

the PCR business to boom in the 1990s as applications began to extend beyond the original core areas of biomedical research and forensics.

4.4. Real-time design phase: QRT-PCR

The accidental discovery that certain fluorescent dyes would bind to PCR samples made it feasible to measure the degree of PCR amplification after each cycle by simply measuring fluorescence. Real-time quantification of the DNA target strand could even be automated by connecting the detection apparatus to a computer that recorded the cycle-by-cycle increase: real time in PCR refers to the ongoing display of results as the PCR process is being performed. Of course, the question is why users would want to merely "quantify" the amount of DNA material replicated in a closed apparatus.

In fact, PCR was gaining traction as a *diagnostic* tool and not merely as an incubator of DNA material. Through DNA amplification and quantification one could infer the amount of target DNA in the original sample. PCR provided a new means for detecting the presence of viruses, bacteria, and other pathogens. Quantitative real-time PCR began to displace even traditional diagnostic procedures like culture and serology. Beyond the advantages of real-time display of results and lesser need for post-reaction analyses, "closed" QRT-PCR amplification reduced the probability of variability and contamination.

Yet even in the already established areas of PCR use such as biomedical research and forensics, such temporal scaling was advantageous. In particular, "real time" records of DNA amplification were valuable for ensuring and documenting *accuracy* in legally sensitive areas. One of the disadvantages associated with the iterative nature of PCR is that if errors occur in one cycle (for example, due to depletion of the polymerase or other reagents), such errors will become magnified in successive cycles. Applied Biosystems was the first firm to offer real-time PCR products, later followed by Qiagen, Roche, and Agilent.

4.5. Post real-time "multiplicative" design phase: M-PCR

Although QRT-PCR offered a rapid process for real-time analysis, its cost-efficiency in use was limited by the fact that it was still applied just to single samples. To overcome these limitations, technology developers explored a range of different design improvements. Among them, M-PCR was the most significant development.

M-PCR stands for multiplex polymerase chain reaction, meaning that multiple samples are processed at once. M-PCR involves "multiplicative" scaling, that is, the simultaneous performance of operations previously performed sequentially (Chamberlain et al., 1988). Each M-PCR run of a 100-row sample, for example, performs the PCR DNA amplification process 100 times during a given single heating-cooling cycle. By allowing large number of samples to be run in parallel, M-PCR increased overall *throughput*. One interviewee emphasized that "by targeting multiple genes at once, additional information may be gained from a single test run that otherwise would require several times the reagents and more time to perform … It is cost-effective … it is high-throughput."

In terms of new applications, a scientist can use reference samples and compare multiple DNA molecules at the same time – a task rendered difficult by single samples. M-PCR is a useful tool for conducting research on diseases like cancer that are triggered by changes in multiple genes, thus necessitating a system for analyzing large numbers of gene patterns. Responding to such demand-side needs, M-PCR became a platform of choice for medical research. Focusing on specific genes or small mutations in specific genes (biomarkers), researchers and academics could furthermore screen patients for particular diseases. Use of M-PCR intensified further in conjunction with the Human Genome Project, as vast new amounts of data on genes and gene-related diseases became available. Academic and clinical researchers availed themselves of gene libraries to explore genes of interest for specific diseases.

M-PCR thus became common in research institutes, diagnostic laboratories, hospitals, and pharmaceutical companies. The development of M-PCR was not the brainchild of any single firm or inventor; instead, M-PCR devices and products were introduced by multiple companies. Multiplex PCR products have made PCR adaptable to high-throughput applications in fields such as diagnostics, pathogen detection, cancer research, biodefense and industrial applications involving food and water. Pharmaceutical and biotechnology firms likewise deployed M-PCR on an increasingly industrial scale.

5. Emergence and development of ERP software

5.1. Overview

Business application software is software used by firms to manage their own operations. For decades developing such software was usually a customized activity carried out by programmers working in or for user firms. Such software long defied efforts at standardization, leaving the business application software market highly fragmented (Campbell-Kelly, 2003). The engineering of standardized products able to accommodate the needs of heterogeneous firms and industries constitutes one of the primary commercial challenges and defines a natural trajectory of technological evolution for business software in general. Intriguingly, the evolution of software design discussed below revealed a pattern broadly similar to that seen in PCR DNA amplification: a sequence of dominant designs that delivered first reliability, then higher speeds, then increased volume.

Among the various types of business application software that arose, the category of enterprise resource planning (ERP) software has been recognized as one that created a whole new product segment by defragmenting the business software market (Gambardella and Giarratana, 2013; Gambardella and McGahan, 2010). ERP is a systems integration technology that provides a "unified view of the enterprise that encompasses all functions and departments and ensures the integrity of global databases of all business records and reports" (Dey et al., 2010: 565). Arguably a misnomer, ERP essentially refers to cross-functionally integrated business application software running on a centralized database. Until the advent of ERP, the standardized business application software that existed was predominantly function-specific (e.g. marketing, finance, etc.).

ERP software is essentially a means of managing an IT-coordinated *process* of ensuring cross-functional integration, as suggested by the "planning" aspect of enterprise resource planning. Two different key processes are involved, in fact: implementation of the software and its daily operation. The implementation of ERP is intricate because adopting firms need to codify all the key company routines in their different operational areas before they can even begin writing and/or installing the software.

ERP software is an unusual technology to the extent that a single firm (SAP AG of Germany) pioneered dominant designs in multiple eras of computing. SAP is exceptional among all major IT companies in the fact that it maintained market leadership in both the mainframe and client/server eras within its product area (Hidding et al., 2011). SAP did not use the term ERP (this was a later US coinage) but simply referred to its product as "integrated business application software." For reasons stemming from idiosyncratic German market demand (Lehrer and Behnam, 2009; Schmidt, 2014), functionally integrated software originated in Germany and other providers of ERP software were largely followers of SAP. SAP has consistently held a first-mover advantage and the highest global market share in this IT market segment (AMR Research, 2006). Both of the two major versions of ERP software that SAP developed, R/2 for mainframe computers and R/3 for client-server networks, were dominant designs in the sense of being the lead products in the market which rivals endeavored to imitate. The following reconstructed account of design evolution in ERP software is therefore SAP-centric.

5.2. Initial design phase: process refinement

The German programmers who left IBM to found SAP in 1972 began by writing software to support all of the firm's major functions linked to a common database. The innovative potential lay not only in its integrated design, but also in the way it could be used. Cross-functionally integrated software did not merely share data across functions but turned out to be a useful instrument for promoting cross-functional coordination in the firm's daily operations. The characteristic of cross-functional integration turned out to be particularly useful for coordinating operations in large, vertically integrated settings like chemicals and petrochemicals, which were leading adopters (Lehrer and Behnam, 2009).

SAP, like many other software companies, economized on coding costs by taking the code and algorithms developed for one client and reutilizing it in work performed for other customers. SAP built ever more adjustable parameters and settings into its core code in order to minimize the amount of bespoke coding required for any given client installation. This phase of ERP software development revealed clear "economies of repetition" (Davies and Brady, 2000). Nonetheless, SAP kept this early recycling and standardization secret, portraying each installation as a fully customized activity (Meissner, 1997: 48). By incorporating ever more parameters to accommodate the needs of different application environments, SAP's business software became viable for use in an ever wider variety of firm and industry settings. Such standardization of the software was a steady ongoing process at SAP during the 1970s.

5.3. Real-time phase: R/1 and R/2

SAP's first suite of integrated business applications (R/1) was not marketed as packaged software; the core code was standardized but not yet a commercial product because SAP still had to adapt each installation to the user's specific requirements. A significant innovation was the systematic incorporation of real-time entry and analysis of data on time-sharing systems, that is, on mainframe computers featuring computer terminals as input-output devices in contrast to batch mainframes. R/1 stood for Realtime/1 and designated a major innovation in the daily operation of ERP systems.

R/2, introduced in 1979, represented SAP's first truly packaged software suite. By "packaged" interviewees meant a standardized product outfitted with the means for corporate users and/or external IT service companies other than SAP to enter the parameters necessary for site customization (parameterization). What set SAP apart from other German rivals also offering integrated software and ultimately ensured SAP's market dominance was the fact that SAP partnered with IT consultancies and engineered its product so that external IT service companies could install its software on corporate premises (Leimbach, 2009; Schmidt, 2014).

R/2 also featured real-time output of data and results. Computer-generated reports could be produced at any time rather than, as was customary, at the end of the month when printouts from the different company functions (finance, HR, etc.) were assembled into reports. The once-a-month or once-per-quarter compilation of company reports was such a deeply ingrained custom that only over time were customers able to grasp the benefits of being able to generate a snapshot of operations and financial flows at any time.

While ease of data entry and data sharing across functions were the primary original technical selling points of ERP systems, further advantages of "real-time" software were discovered by users in the course of the 1980s. Among these was the capability to empower top management to engage in direct monitoring of operations. A real-time ERP system built upon a centralized database allowed content to be accessed at any time and from any terminal attached to the system. This feature reduced the opportunities for information hoarding by specialized functional areas and gave top managers the ability to monitor details of firm operations at any moment. ERP systems were therefore appealing to top managers, and SAP targeted top managers rather than IT departments as their specific buyers. During the 1980s, R/2 gradually came to dominate the large-enterprise software market in Germany, with 2200 installations at its height. In contrast, the adoption of R/2 abroad was mainly limited to specific industries like chemicals and petrochemicals.

To summarize, the temporal scaling of ERP software provided multiple benefits to users, many of which were only slowly discovered in the course of time. Interviewees noted that the fad of "re-engineering the corporation" (Hammer and Champy, 1993) proved to be a great boon to the adoption of ERP in the 1990s: implementing a companywide ERP system provided a natural context (and pretext) for re-engineering corporate processes.

5.4. Post real-time "multiplicative" design phase: R/3 and client-server architecture

The follow-up product to R/2, namely R/3, was designed to run in distributed computing environments. SAP continued to sell R/2 installations for mainframe environments, mainly in large corporations, while targeting medium-sized as well as tech-savvy, Silicon Valley-like firms with R/3.

The client-server architecture of R/3 enabled multiplicative scaling in ERP software. R/3 systems were scalable, permitting the user firm's ERP system to be augmented at will by adding servers and clients. The client/server version of ERP software thus ultimately expanded the volume of user operations by allowing parallel operations to be performed simultaneously. Such simultaneous processing of applications allowed ERP systems to be operated on a large scale. The upshot was that although R/3 had been conceived as an ERP solution for medium-sized firms, its distributed computing capabilities made it attractive precisely to large multinational firms looking for an IT tool to coordinate their globally dispersed operations. Oddly enough, it was a major SAP customer in the US, Chevron, that first investigated the feasibility of adopting R/3 rather than R/2 to integrate globally dispersed operations. Plattner (2000: 101) related: "Chevron was the first to break out and understand that client-server was also suited for large companies. Chevron examined, measured, and invested heavily in the R/3 system and finally determined that R/3 had the potential to surpass the mainframe."

In addition, multiplicative scaling palliated certain technical limitations of ERP installations on mainframes. One limitation was that multiple users could not access the same application server simultaneously. To overcome this problem, SAP replaced the conventional two-tier (client-server) system with a three-tier architecture that featured a middle tier of application servers between the central database server and the decentralized clients. Although novel at the time, the three-tier architecture introduced by SAP became standard in corporate IT systems (Lehrer et al., 2016; Rashid et al., 2002).

Large multinational firms discovered ERP software as a technology to integrate operations across their globally dispersed subsidiaries at just the time that breakthroughs in transmission and routing technologies and improvements in computer chips were allowing the performance of client-server systems to equal and surpass that of mainframe systems. In virtually all industries and even in the public sector, ERP software became the instrument of choice for coordinating and re-engineering operations. ERP was adopted "multiplicatively" in the sense that multinational corporations began installing a unified ERP system on a global basis across all of their national subsidiaries, supplanting country-by-country IT systems in different subsidiaries.

6. Discussion

In the Abernathy-Utterback model, a phase of product innovation precedes a phase of process innovation. Not unlike the innovation patterns in services studied by Barras (1986), the order appears – at least at first glance – to be largely the reverse for process-based technologies. The foregoing case studies suggest a common stage model consisting of

three basic stages: 1) process refinement, 2) temporal scaling, and 3) multiplicative scaling. The following stage-by-stage remarks underline the way technology providers compressed the time needed to perform a complex technological process, in part by encapsulating this process in innovative technology products.

6.1. Process refinement

For the process-based technologies considered here, the term "process refinement" helps avoid confusion with the concept of "process innovation" concerning novel methods in how a given product is manufactured. Process refinement in PCR DNA amplification involved determining the optimal set of steps, ingredients and equipment for performing the DNA amplification procedure and codifying the way alternating heating and cooling steps were to be carried out in replicating target strands of DNA. In ERP software it involved (secretly) reutilizing software code from one customer's premises to the next instead of writing customized software from scratch. In both cases, an initial focus of innovation was on ensuring greater reliability.

6.2. Temporal scaling

The next focus of innovation was on temporal scaling, that is, on performing the process faster. The focus on temporal scaling culminated in automation. The automation of a standardized complex process denotes a stage in technology development in which the complex process is performed less as a separate, "offline" activity. Automation of PCR – incorporating the product innovations of a special polymerase and thermal cyclers – greatly accelerated and simplified performance of the PCR process. The development of a real-time (time-sharing) version of ERP software enabled users to input data and print out reports at any time instantly: "automation" refers to the automatic updating of firm information in real time. In both cases, automation constituted a milestone enabling time compression in the user's deployment of the technology. One way technology developers communicated the achievement of this milestone was to use the slogan of "real time" in their products.

An interesting common feature of PCR and ERP development, albeit one whose generalizability is hard to gauge, is that temporal scaling went hand in hand with the phenomenon of "productization." Productization designates the emergence of market products out of technologies that began essentially as processes, i.e. as non-products (Jaakkola, 2011; Sainio and Marjakoski, 2009). Efforts to automate piggybacked on efforts to embody the process-based technologies of PCR and ERP in marketable products.

In the PCR case, products like the Taq polymerase and thermal cyclers were a direct outgrowth of efforts to accelerate and ultimately automate use of the PCR process. In the ERP case, automation came about as the result of complementary hardware technology, namely time-sharing computer systems. The development of SAP's breakthrough product R/2 targeted time-sharing users and involved standardization of the software code to the extent it became possible to offer an ERP product that was separable from the coding service. Productization is a common feature of the software industry, and the term "productization" is in fact borrowed from the software sector (Jaakkola, 2011; Sainio and Marjakoski, 2009).

While significantly accelerating the use of PCR and ERP, automation was not the final stage in development of these technologies. There still remained constraints on the volume of work that could be performed at a given time: automating PCR still allowed only one sample to be processed at a time, while real-time ERP systems were still constrained by the capacity of the mainframe computer.

6.3. Multiplicative scaling

The advent of multiplicative scaling enabled an order-of-magnitude improvement in utilization capacity and throughput. Multiplex PCR allowed multiple samples to be run at once, whereas ERP systems on client/server architecture became "scalable," i.e. expandable by adding further servers and clients (Doganata et al., 2010: 288). Multiplex PCR and client/server ERP utilized design breakthroughs somewhat analogous to the principles of scaling up in product design in which higher efficiency results from economies of scale in product size (Narasimalu and Funk, 2011; Sahal, 1985).

The concept of "throughput," which surfaced in our interviews, helps explain the how speed and volume, temporal and multiplicative scaling, are related. For process-based technologies like PCR and ERP, reducing the time required to perform the core process constitutes a major dimension of performance improvement. Accelerating the speed and increasing the volume with which the process is performed are two different avenues for increasing throughput, that is, increasing the number of times the process can be performed within a given time period.

6.4. Reconciling the stage model with traditional product-process lifecycle models

With the foregoing design phases in mind, it may be helpful to compare our stage model with the famous product-process lifecycle model of Abernathy and Utterback (1978). While different in detail, the overall evolution of the underlying economic logic motivating technology developers is actually rather similar. In their model, a new technology product market evolves from a focus on product innovation to a focus on process innovation, with the shift triggered by a so-called dominant design. In the stage model outlined above, the phenomena of both process refinement and productization correspond more or less to the Abernathy/Utterback period of product innovation in which variety reduction and reliability are paramount. In contrast, the phenomena of both automation and multiplicative scaling correspond to the Abernathy/Utterback phase of process innovation to the extent that in both cases efficiency concerns predominate. As mentioned, economies of scale in ERP and PCR resulted precisely from the scaling up of products (Narasimalu and Funk, 2011; Sahal, 1985) rather from the kind of process innovation phase noted by Abernathy and Utterback. One can furthermore postulate that "real-time" embodiments of the technologies played something of the role of a dominant design. These approximate parallels are summarized in Table 2.

In other words, our stage model enables us to place the classic product-process lifecycle models (Abernathy and Utterback, 1978; Murmann and Frenken, 2006) in a larger context. One can logically posit that design efforts in the early phase of the technology lifecycle will go into improvement of the technology to ensure functional reliability and that these efforts will show up as "product innovation" when the technology in question is product-based and as "process refinement" when the technology in question is process-based. Once a first major dominant design has penetrated and defined the market, subsequent design efforts will aim at efficiency improvements. Again, the nature of these design efforts will depend on the kind of technology

Table 2
Comparison of stage model with Abernathy/Utterback model.

Abernathy/Utterback phases	Comparable stage model phases	Economic focus in both cases
Product innovation phase	Process refinement, productization	Variety reduction, reliability
"Real-Time" dominant design	Temporal scaling, automation	Facilitating product take-off by creating a recognized leap in cost effectiveness
Process innovation phase for greater efficiency and economies of scale	Multiplicative scaling	Economies of scale through scaled-up design of products or systems

involved. Whereas product-based technologies may give eventual rise to process innovations in an economic sense (OECD, 1997), our research suggests that process-based technologies will stimulate efforts at temporal and multiplicative scaling. To reiterate, both temporal and multiplicative scaling enhance throughput.

6.5. Demand-side factors

The foregoing remarks pertain mainly to the supply side. Of course, it is also the nature of demand that determines the extent to which efficiency-oriented design innovations – temporal and multiplicative scaling, in the cases examined here – will actually be relevant to the marketplace. The success of "real-time" versions of PCR and ERP products reflects market interest in offerings that made use of temporal scaling. Complex gene-related diseases and the Human Genome project were drivers of demand for multiplicative scaling in PCR, whereas in ERP it was the conjunction of client-server architectures and the requirements of multinational corporations that drove demand for a "multiplicative" ERP product. In both cases, users learned about new ways to use these technologies on a vastly greater scale (e.g., biomarkers and the Human Genome project, global client-server architectures in multinational corporations), thus intensifying demand and use of these process-based technologies. In the absence of such demand, the stages of temporal or multiplicative scaling may not emerge for lack of market interest.

Viewed from a broader perspective, the appetite of PCR and ERP users for ever greater scale confirms the relevance of demand-side approaches to technology strategy (Adner, 2004; Priem et al., 2012). The ability of technology providers to sell products with attributes of higher performance, including scale, is constrained as much by demand conditions as by conditions of technology supply (Adner and Levinthal, 2001). The graphically informative juxtaposition of technology S-curves and demand S-curves (Adner, 2004: 30) can be fruitfully applied to summarize the basic findings of the PCR and ERP case studies.

Following the heuristic depiction offered by Adner (2004), Fig. 2 represents the technology S-curve against the left vertical axis of performance in terms of throughput. This S-curve exhibits two vertical inflections corresponding to the introduction of temporal and multiplicative scaling, respectively, on the supply side. The demand S-curve, mapped against the willingness to pay (WTP) for apparatus as depicted in the right vertical axis, exhibits the characteristic flattening of demand curves later in time (i.e., towards the right). However, the demand S-curve remains above the technology S-curve in the time frame considered (roughly 1975–2010). Adner (2004: 29) identified photolithographic printing equipment for semiconductors as an analogous technology in which demand perennially exceeded supply in technology performance. In such situations, technology improvement does not outstrip what the market requires, thus avoiding the syndrome underlying the famous Innovator's Dilemma (Christensen, 1997).

While Fig. 2 involves simplifications and glosses over differences between the technologies, as a heuristic it communicates two generic features of the phenomenon in which "the sky is the limit on scale" in technology development. The first, of course, is the appetite of PCR and ERP users for greater scale on the demand side. This appetite was not unbounded in advance, however, but rather was expanded by user learning about new applications of these technologies involving higher throughput and scale. The case studies document amply how complementary technologies in the case of PCR and lead users in the case of ERP stimulated industry-wide demand for higher-scale applications. The second generic feature applies to vertical inflections of the technology supply curve in tandem with innovations in scaling. The advent of temporal and multiplicative scaling produced steep discrete improvements in throughput performance. Although these improvements were first visible on the supply side, in the case of both PCR and ERP they ultimately spilled over into demand conditions as well. User learning about new high-throughput applications of PCR and ERP constituted the nexus between scaling up on the supply side and scaling up on the demand side.

7. Conclusion

The attempted reconciliation in Table 2 of our stage model with the classic product-process lifecycle models (Abernathy and Utterback, 1978) helps resolves an obvious tension in the way the concept of scale has been used. The notion of economies of scale co-existed uneasily with concept of the scaling-up (or scaling-down) of products and systems. Yet as Narasimalu and Funk (2011) point out with regard to Danish wind turbines, both types of scale ultimately serve the purpose of generating higher cost efficiency.

Although case studies like ours authorize only limited conclusions about how to organize and manage firms, they do provide a framework for anticipating possible future developments and inflection points of the technology. Like other technology life cycle models, our derived stage model can inform future planning at the firm and inter-firm level. For example, when firms ponder the longer-term ramifications of temporal scaling in process-based technologies, they need to look

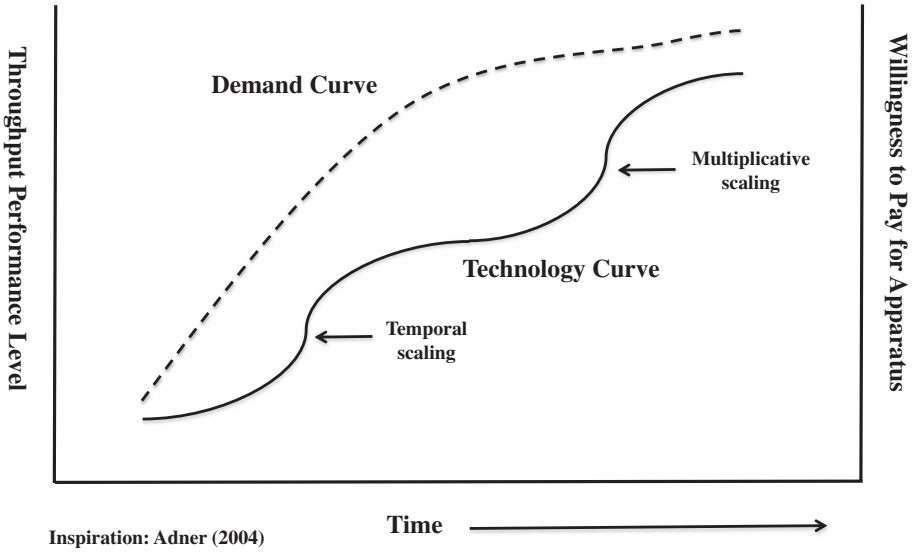

Fig. 2. Technology and demand S-curves.

beyond just acceleration of the basic process. One way of achieving the equivalent of temporal scaling is to use multiplicative scaling – an "industrial" innovation, one might say, rather than a technical improvement of the core technology. For many technologies, process-based or not, the longer-term economic benefits stem from "industrial" improvements made after the basic technology has been established (Sherif, 2006: 184), just as the classic Abernathy/Utterback model implies. An interesting feature of the process-based technologies examined here is that the "industrial" scaling up later in the life cycle evidently stemmed from yet further *product* innovation, that is, new dominant designs in products. Evidently, there are different possible routes to industrial scaling up that technology providers need to anticipate: in some cases, phases of intense product innovation give way to a greater emphasis on process innovation, whereas in other cases, as here and in Barras (1986), the lifecycle may actually begin with process-based innovations that later give way to multiple stages of product innovation, embracing such phenomena as productization, automation and multiplicative scaling.

References

Abernathy, W.J., Utterback, J.M., 1978. Patterns of industrial innovation. Technol. Rev. 80 (7), 40–47.

Adner, R., 2004. A demand-based perspective on technology life cycles. Adv. Strateg. Manag. 21, 25–43.

Adner, R., Levinthal, D., 2001. Demand heterogeneity and technology evolution: implications for product and process innovation. Manag. Sci. 47 (5), 611–628.

AMR Research, 2006. The ERP market sizing report, 2006–2011. Report Boston.

Anderson, P., Tushman, M.L., 1990. Technological discontinuities and dominant designs: a cyclical model of technological change. Adm. Sci. Q. 35, 604–633.

Barras, R., 1986. Towards a theory of innovation in services. Res. Policy 15 (4), 161–173.

Benner, M.J., Tripsas, M., 2012. The influence of prior industry affiliation on framing in nascent industries: the evolution of digital cameras. Strateg. Manag. J. 33 (3), 277–302.

Cabral, R., Leiblein, M.J., 2001. Adoption of a process innovation with learning-by-doing: evidence from the semiconductor industry. J. Ind. Econ. 49 (3), 269–280.

Campbell-Kelly, M., 2003. From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry. MIT Press, Cambridge, MA.

Chamberlain, J., Gibbs, R., et al., 1988. Detection screening of the Duchenne muscular dystrophy locus via multiplex DNA amplification. Nucleic Acids Res. 16, 11141–11156.

Christensen, C.M., 1997. The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail. Harvard Business School Press, Cambridge, MA.

Christensen, C.M., Suarez, F.F., Utterback, J.M., 1998. Strategies for survival in fast-changing industries. Manag. Sci. 44 (12), 207–220.

Davies, A., Brady, T., 2000. Organizational capabilities and learning in complex product systems: towards repeatable solutions. Res. Policy 29 (7), 931–953.

Dey, P.K., Clegg, B., Bennett, D., 2010. Managing implementation issues in enterprise resource planning projects. In: Sherif, M.H. (Ed.), Handbook of Enterprise Integration. CRC Press, Boca Raton, pp. 563–578.

Doganata, Y., Kozakov, L., Jahn, M., 2010. Software architecture for enterprise applications. In: Sherif, M.H. (Ed.), Handbook of Enterprise Integration. CRC Press, Boca Raton, pp. 271–306.

Florida, R., Kenney, M., 1990. High-technology restructuring in the USA and Japan. Environ. Plan. 22, 233–252.

Fore, J., Wiechers, I.R., Cooke-Deegan, R., 2006. The effects of business practices, licensing, and intellectual property on development and dissemination of the polymerase chain reaction: case study. J. Biomed. Discov. Collab. (www.j-biomed-discovery.com/content/1/1/7).

Frenken, K., Leydesdorff, L., 2000. Scaling trajectories in civil aircraft (1913–1997). Res. Policy 29 (3), 331–348.

Gambardella, A., Giarratana, M.S., 2013. General technologies, product market fragmentation, and markets for technology. Res. Policy 42 (2), 315–325.

Gambardella, A., McGahan, A.M., 2010. Business-model innovation: general purpose technologies and their implications for industry structure. Long Range Plan. 43 (2), 262–271.

Gasson, S., 2004. Rigor in grounded theory research: an interpretive perspective on generating theory from qualitative field studies. In: Whitman, M.E., Woszczynski, A.B. (Eds.), The Handbook of Information Systems Research. Idea Group, Hershey, PA, pp. 79–131.

Gerring, J., 2007. Case Study Research: Principles and Practices. Cambridge University Press, Cambridge.

Glaser, B.G., Strauss, A.L., 1967. The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine Publishing, Chicago.

Hammer, M., Champy, J.A., 1993. Reengineering the Corporation. HarperCollins, New York.

Hidding, G.J., Williams, J.R., Sviokla, J.J., 2011 January 25. The IT platform principle: the first shall not be first. Sloan Manag. J. Exec. Adv.

Jaakkola, E., 2011. Unraveling the practices of "productization" in professional service firms. Scand. J. Manag. 27 (2), 221–230.

Lehrer, M., Behnam, M., 2009. Modularity vs. programmability in design of international products: beyond the standardization-adaptation tradeoff? Eur. Manag. J. 27 (4), 281–292.

Lehrer, M., Banerjee, P.M., Wang, I.K., 2016. The improvement trajectory of PCR DNA replication and ERP software as general purpose technologies: an exploratory study of 'anchor technologies'. Tech. Anal. Strat. Manag. 28 (3), 290–304.

Leimbach, T., 2009. Die Geschichte der Softwarebranche in Deutschland. (PhD Dissertation). Ludwig-Maximilians-Universität München.

Linton, J.D., Walsh, S.T., 2008. A theory of innovation for process-based innovations such as nanotechnology. Technol. Forecast. Soc. Chang. 75 (5), 583–594.

Maine, E., Garnsey, E., 2006. Commercializing generic technology: the case of advanced materials ventures. Res. Policy 35 (3), 375–393.

Maine, E., Lubik, S., Garnsey, E., 2012. Process-based vs. product-based innovation: value creation by nanotech ventures. Technovation 32 (3), 179–192.

Meissner, G., 1997. SAP - Die heimliche Software-Macht. Hoffmann und Campe, Hamburg.

Mullis, K.B., 1990. The unusual origin of the polymerase chain reaction. Sci. Am. 262 (4), 56–65.

Murmann, J.P., Frenken, K., 2006. Toward a systematic framework for research on dominant designs, technological innovations, and industrial change. Res. Policy 35 (7), 925–952.

Narasimalu, S., Funk, J.L., 2011. Geometric scaling, long term reductions in cost, and implications for public policy: The case of wind turbines. Paper Presented at DRUID 2011, Copenhagen.

OECD, 1997. Oslo Manual: The Measurement of Scientific and Technological Activities. OECD, Paris.

Pavitt, K., 2003. Specialization and systems integration: Where manufacturers and services still meet. In: Principe, A., Davies, A., Hobday, M. (Eds.), The Business of Systems Integration. Oxford University Press, Oxford, pp. 78–91.

Plattner, H., 2000. Dem Wandel Voraus. Galileo Press, Bonn.

Priem, R.L., Li, S., Carr, J.C., 2012. Insights and new directions from demand-side approaches to technology innovation, entrepreneurship, and strategic management research. J. Manag. 38 (1), 346–374.

Rabinow, P., 1996. Making PCR. University of Chicago Press, Chicago.

Rashid, M.A., Hossain, L., Patrick, J.D., 2002. The evolution of ERP systems: a historical perspective. In: Hossain, L., Patrick, J.D., Rashid, M.A. (Eds.), Enterprise Resource Planning: Global Opportunities and Challenges. Idea Group Publishing, Hershey, pp. 1–16.

Rosenberg, N., 1963. Technological change in the machine tool industry, 1840–1910. J. Econ. Hist. 23 (4), 414–443.

Rosenberg, N., 1976. Perspectives on Technology. Cambridge University Press, Cambridge.

Sahal, D., 1981. Patterns of Technological Innovation. Addison-Wesley, New York.

Sahal, D., 1985. Technological guideposts and innovation avenues. Res. Policy 14 (2), 61–82.

Sainio, L.-M., Marjakoski, E., 2009. The logic of revenue logic? Strategic and operational levels of pricing in the context of software business. Technovation 29 (5), 368–378.

Schmidt, T., 2014. Ko-spezialisierung in Strategischen Netzwerken: Eine Pfadtheoretische Untersuchung. Kölner Wissenschaftsverlag, Cologne.

Sherif, M.H., 2006. Standards for telecommunications services. In: Jakobs, K. (Ed.), Advanced Topics in Information Technology Standards and Standardization Research. Idea Group Publishing, Hershey, PA, pp. 183–205.

Slayton, R., Spinardi, G., 2016. Radical innovation in scaling up: Boeing's Dreamliner and the challenge of socio-technical transitions. Technovation 47, 47–58.

Suarez, F.F., Utterback, J.M., 1995. Dominant designs and the survival of firms. Strateg. Manag. J. 16 (6), 415–430.

Thoma, G., 2009. Striving for a large market: evidence from a general purpose technology in action. Ind. Corp. Chang. 18 (1), 107–138.

Tushman, M., Murmann, J.P., 2003. Dominant designs, technology cycles, and organizational outcomes. In: Garud, R., Kumaraswamy, A., Langlois, R. (Eds.), Managing in the Modular Age: Architectures, Networks, and Organizations. Blackwell, Malden, MA, pp. 316–348.

Youtie, J., Iacopetta, M., Graham, S., 2008. Assessing the nature of nanotechnology: can we uncover an emerging general purpose technology? J. Technol. Transfer. 33 (3), 315–329.

Mark Lehrer received his BS at MIT (1980) and his PhD at INSEAD (1997). He is now Professor of Strategy and International Business at the Sawyer Business School, Suffolk University, Boston, Massachusetts.

Preeta M. Banerjee was formerly Assistant Professor of Strategy at Brandeis University and now works for Deloitte Services LP in Waltham, Massachusetts.

I. Kim Wang received her PhD at the University of Illinois and is Assistant Professor of Strategy and International Business at the Sawyer Business School, Suffolk University, Boston, Massachusetts.