Product, Process, and Service:
A New Industry Lifecycle Model

Michael Cusumano
MIT Sloan School of Management
50 Memorial Drive, E52-538
Cambridge, MA 02142-1347 USA
617-253-2574
cusumano@mit.edu

Steve Kahl
MIT Sloan School of Management
50 Memorial Drive, E52-511
Cambridge, MA 02142-1347 USA
617-253-6680
skahl@mit.edu

Fernando F. Suarez
Boston University School of Management
595 Commonwealth Ave., Room 546-F
Boston, Massachusetts 02215, USA
617-358-3572
suarezf@bu.edu

June 7, 2006

Abstract

Existing models of industry lifecycle evolution tend to focus on changes in the products and processes and largely overlook the dynamics of services. Increasingly, however, the revenues of many firms are becoming dominated by sales of services rather than products, or products sold with services to gain competitive differentiation in markets marked by increasing product commoditization. In this paper, we try to explain the role of services in firm and industry evolution. To validate our framework, we analyze the revenue mix and contribution of services at several hundred publicly listed firms in the pre-packaged software products industry. We also offer comparisons to firms in other industries.

Keywords: industry life cycles, services, computer software industry

1. Introduction

One of the main tenets of how firms and industries evolve is that, as some businesses mature, the basis of competition shifts from product innovation to process innovation (Utterback and Abernathy 1975; Utterback and Suarez 1993; Utterback 1994; Klepper 1996, 1997; Adner and Levinthal 2001). For example, the model initially proposed by Utterback and Abernathy in 1975 holds that, early after the birth of a new industry, firms compete based upon product differentiation, investing heavily in developing new product features and determining what consumers want. But, as the market matures and customer needs become more defined, firms shift their focus to competing on cost and economies of scale, investing more heavily in manufacturing and other processes in order to make production operations more specialized and efficient.

This product-process lifecycle model does not hold for all industries; it seems to apply more to manufacturing settings where "dominant designs" or product standards emerge, and where competition then shifts to price (Utterback 1994). Sometimes a technological discontinuity interrupts this maturation process and the cycle starts over again (Tushman and Anderson 1986). Nonetheless, this basic lifecycle model has become an important framework in management literature to help us think about what strategies and investments companies should emphasize at different periods in their evolution and in different competitive environments (Porter 1980, Oster 1994).

We argue in this paper that the product-process lifecycle model is incomplete in that at least some firms in some industries seem to move on to an additional phase: a period where the emphasis of a firm, as indicated in its major source of revenues or profits or both, shifts to services (Figure 1). Several authors have noted the increasing importance of the service sector in most industrialized nations (e.g. Quinn, 1992), while others have looked at the differences between service firms and manufacturing firms (e.g. Heskett et al, 1997). However, little research has been done on the transition towards services within the firm, particularly from an industry life cycle perspective. The data we have collected so far strongly suggests that, in manufacturing and even in technology-based industries that normally are not considered as services businesses, many companies encounter a gradual decline in their basic products business and increases in revenues from sales of services. For firms, shifting their strategic focus due to changes in the environment often proves to be a major challenge. For instance, the shift from a focus on product innovation to one on process has been shown to affect firm survival (Suarez & Utterback, 1995). A successful transition from product to process requires firms to change their organizational structure and acquire new capabilities (Utterback and Abernathy, 1978). Likewise, the additional shift towards services implies that, as an industry matures, firms cannot simply focus on cost-based competition as prevailing life-cycle theories suggest. Rather, firms should also develop additional capabilities in services which often differ from traditional production activities.

In this paper, we test the validity of our “product, process, and service” lifecycle model for the pre-packaged software products industry, characterized by companies such as Microsoft, SAP, and Oracle. In the remaining sections of this paper, first, we provide some background on the motivations for the study and the basic questions we are asking. Second, we discuss how we distinguish products and services in general and in the context of computer software, and highlight the economic relationship between the two. Next, we examine how services fit into the software business or industry lifecycle. Finally, we provide a descriptive and quantitative investigation of the pre-packaged software industry and analyze how common it is for software product companies to develop a services business, as well as why and when this transition might occur.

Research Background and Basic Questions

This paper is part of a larger multi-year research project examining the interrelationship of products and services in computer software, with comparisons to other industries.[1] Cusumano (2004) posed the initial research questions after studying the financial records of several major software companies. Some firms, such as Microsoft and Adobe, have reported nearly 100 percent software product revenues since their founding and represent almost “pure” product companies. However, at other firms also normally thought of as products companies, such as SAP, Oracle, PeopleSoft, and Seibel, there occurred a major shift toward services during the mid-1990s, with services accounting for 70 percent or more of total revenues by the year 2000. These are mainly enterprise applications vendors that offer themselves or rely on third-party firms to provide customization of their products as well as long-term maintenance contracts, technical support, training, strategic advice, and other services.

Based on this small sample, Cusumano proposed a simple model suggesting that at least some products companies, as they mature, become service (and maintenance) companies over time (Figure 2). This model suggests that firms may start out focusing on products and intending to remain as products companies. Eventually, however, services revenues may increase to the point where, for some possibly interim time period, they are really “hybrid” providers of products and services. At some later point, the revenue streams “crisscross” and services become larger than product revenues.

We do not know from the small sample of companies how common it is for software product companies to experience this “crisscross” and become dominated by services and maintenance business, how much time this transition usually takes, which types of software products companies do not undergo this transition, or what the specific impact is of the transition on measures of performance, such as firm survival, profitability, sales growth rates, and market value. With regard to causality, a number of factors might converge to transform a software products company into a services company. For example, as an industry matures and markets become saturated, companies may find it difficult to locate new customers for their products. In this case, they might begin to emphasize services, such as product customization or strategic consulting. Or they may come to rely primarily on long-term maintenance contracts for product upgrades and ongoing technical support rather than new product sales.

Competition may also increase in the industry, turning some products into commodities; this may drive down product prices more than services prices because services may be generated through long-term contracts and hourly or daily billable hours. Prices of software products or other high-technology products may fall quickly or, as in the case of free and open software, fall to a price of zero. In addition, there are times when new “disruptive” technologies appear such as open-source products like the Linux operating system or the Apache web server, or different products based on Internet technology rather than traditional mainframe computers or client-server systems. In these cases of transition to a new industry lifecycle, software product companies as well as other firms such as IBM and dedicated IT services providers, can generate significant revenues by offering strategic and integration services to help customers understand and adopt the new technologies (Figure 3).

Even reviewing a small sample of other technology-based manufacturing firms suggests there is a shift going on toward services in businesses where intense global competition or low-cost overseas production from Chinese firms and factories already have made it difficult for firms to make a profit on sophisticated but now commonly available industrial products. IBM, for example, sells a variety of computer products as well as packaged software products and IT services. In a recent fiscal year, 48 percent of IBM's $89 billion in revenues came from services such as management consulting and IT outsourcing. This figure compares to only 18 percent of revenues coming from services in 1995 (Table 1). Other diversified computer and systems companies show similar trends. At Hewlett-Packard, according to its annual report, 19 percent of its $80 billion in revenues came from services such as technical support, IT consulting, integration work with various computer systems, business portals, and web services. This compared to only 14 percent of revenues in 1995. Sun Microsystems, a leading provider of computer workstations as well as storage systems and Unix software, reported no services revenues in 1995 but 34 percent service revenues in 2004. During the same time period, Cisco, the world's larger provider of networking systems, reported no service revenues in 1995 but 22 percent in 2004. At EMC, the world's largest storage-devices maker, 22 percent of its $6 billion in revenues came from services such as solutions consulting, interoperability testing for various storage devices, training, and technical support.

At General Electric, perhaps the world's largest diversified technology company, its annual report notes that 30 percent of its non-financial services revenues in 2003-2004 came from services; including financial services, the total is 62 percent of revenues. In contrast, non-financial services revenues for GE in 1995 accounted for 14 percent of revenues. GE's service offerings covered a broad range: product support and maintenance; electrical apparatus installation, engineering, repair and rebuilding services; computer-related information services; network television, cable, Internet and multimedia programming and distribution services (through its NBC subsidiary). It's General Electric Capital Services subsidiary offered a broad array of financial and other services including consumer financing, commercial and industrial financing, real estate financing, asset management and leasing, mortgage services, consumer savings and insurance services, and specialty insurance and reinsurance. As shown in Table 1, GE's non-financial services business also had superior gross margins to its products business (37 versus 26 percent).

At some of the world's largest automobile manufacturers, the prominence of services is even more striking. In recent years, General Motors has made little or no money from its products business. It has made most of its profits from financial services – selling loans or leases to customers. Half of GM's financial services revenues and profits also come from selling home mortgages.

In short, services seem to be increasing in a number of companies in different industries, and increasing not only in terms of revenues but also in terms of profit contribution. In this paper, we measure this transition to services in the pre-packaged software industry, although at this point we need to refine our definition of what is a product and what is a service – in general and in the specific case of software.

Distinctions and Complementarities between Products and Services

In general, most government classification schemes as well as the management and economics literatures consider services as anything that is not a tangible, manufactured good and is consumed as it is produced and sold.[2] For example, the industry that we consider in this paper, the software products business, is actually categorized as a service business by the U.S. government's standard industrial classification scheme (SIC code 7372).

We prefer a more specific definition to distinguish products and services. In our view, a "product" is the core offering of a firm. Services, when offered by a products company, are usually complementary activities meant to assist in promoting adoption of the core product or to enhance the core product. In our study, when software products companies report revenues from services, they are generally referring to activities such as product customization to create features or user interfaces tailored to a customer's special needs, or consulting to plan and implement these software solutions, or integration work to make different software products or databases operate together. They also may be referring to training on how to use new software. In addition, perhaps half or more of services for many software product firms selling to large organizations consist of revenues from maintenance contracts that cover technical support to rectify problems and provide upgrades of new versions of the software at long as usage terms (such as the number of users and the specific functional modules being licensed) do not change.

Software companies and other firms that offer both products and services are supposed to account for these different types of revenues in different ways, based on generally accepted accounting rules. For example, companies can recognize product revenues immediately when they ship a software product, whereas services are delivered over longer-term periods and can be recognized as revenues only as the company delivers or performs the services.

Moreover, many managers of software products companies and their investors, at least in past years, have preferred the products business to the services business because of much higher potential margins: Replicating a software product, which is a digital good, is a trivial expense. Gross margins on the products business (that is, sales minus direct expenses for producing and delivering the product) can approach 99 percent. At the same time, we must note that large R&D, sales, and marketing expenses may erode much of these potential profits. In contrast, delivering services is usually much more labor-intensive and costly, and gross margins can vary widely, often ranging between 30 and 60 percent (Cusumano 2004).

It is also useful to think of many services as tied to products, which can make them especially difficult to distinguish. A customer might buy an enterprise application from SAP. But it is extremely unlikely that the customer can install this complex software without spending a large amount of money on services to modify, install, and integrate the software with other systems. SAP or its partners may provide these additional services and change the basic product to the extent that it becomes murky where the standardized product ends and the customization services begin.

Conceptually and strategically, however, we can think of this close relationship between products and services as another form of complementary assets (Teece 1986). Two activities are complementary when an increase in one raises the marginal return of the other (Milgrom and Roberts 1990, Brandenburger and Nalebuff, 1996). The activities are synergistic in the sense that one benefits from the other. Note that the synergies can run in both directions. In a variety of industries, the additional sale of a product can generate service opportunities in the form or warranty agreements, maintenance, training, implementation, and after-sale technical support. In turn, the additional sale of services may generate product opportunities as the firm gets exposed to and learns a particular customer's requirements and how the customer uses the product (Rosenberg 1983; Von Hippel and Tyre 1995).

Our definition of products and services as core and complementary does not require that products must exist before services. In fact, in software and other industries, at least some firms may offer services first to learn user requirements before developing a standardized product. Generally speaking, however, the existence of a product tends to create and precondition the need for services. There is no need for financing an automobile if the customer does not buy an automobile. There is no need to buy maintenance for an aircraft or a locomotive if there is no customer that already has purchased the products. Financial institutions tend not to provide investment advice unless the customer has an account. Post-implementation support of software cannot happen if the customer does not purchase or acquire the software products.

Over time, services might also become somewhat independent of the products business. Sometimes, firms can generate additional services from the installed base without selling additional products. In General Motor's case, the financing arm may assist with purchases other than automobiles, such as homes. In the software case, employee turnover at the customer can generate additional training opportunities or a customer may want to change how it uses the software and require new customization services. Given the structure of maintenance contracts, even when a customer upgrades to a new version of the product, this may generate more service revenue than product revenue. From a product sales perspective, as long as the customer does not add more users or expand its functional use of the product, the upgrade should be covered by the maintenance agreement, which is usually fixed as an annual payment based on a percentage (usually between 15 and 25 percent) of the initial product purchase or licensing price. However, new versions of software often have technical changes or different procedures which may require additional service support in the form of consulting and training. Lastly, as products mature and become commoditized, and price competition intensifies, firms in software and other industries might focus on services as another means of differentiating their products (Quinn 1992; Heskett et al. 1997; Oliva and Kallenberg 2003).

The complementary relationship between products and services is, therefore, important economically and strategically. Economically, services have been linked to longer term and more stable sources of revenue, which may protect firms during economic downturns (Quinn 1992) or periods of commoditization (Heskett et al., 1997). Strategically, since services often require different labor-intensive capabilities, integrating products and services can make imitation by competitors more difficult (Heskett et al. 1997). As we argue below, how services impact industry economics and firm competitiveness – other things being equal – can change as the industry evolves.

It is also not necessarily the case that the products business has higher margins than a services business. This may often be true (Andersen et al. 1997). However, in industries already suffering from price competition, services may offer higher profit margins than the products business and become a major and more stable source of after-sale revenues and profits (Knecht et al. 1993, Gadiesh and Gilbert 1998, Quinn 1992).

Products, Processes, and Services in the Software Business

Technically, software programs consist of code that provides instructions to perform a certain task, whether it is to manage a business process, interface with hardware systems, manage documents, or display a video game. In order to be operable, software programs require storage and computational capabilities of a hardware system. Originally, software ran on mainframes, minicomputers, and microcomputers, but more recently software has been embedded in a wide variety of "hardware" systems, including cell phones, automobiles, machine tools, and iPods. We can think of different software products or systems as a stack of layers sitting on top of hardware (Figure 4).

The notion that software technology consists of layers or a "stack" is important to our purposes as it helps define the various sub-markets and illustrates the requirement for particular types of services, such as for backward integration. In the 1950s and 1960s, the software stack was completely integrated with the hardware and different layers or functional subsystems were contained in the same software program. Initially, hardware vendors bundled all these software layers together, offering it as part of the hardware package. In the mid-1950s, due to anti-trust pressure, IBM released maintenance information on its systems, creating a nascent business for independent software services companies. But most of the new sub-markets for products, customization, and maintenance services emerged after IBM's unbundling decision in the late 1960's and early 1970's and as new software product firms were able to target different levels of the software stack. By the mid to late 1980s, the pre-packaged software market was vertically disintegrated from the technological perspective.

The history of the software business reflects how the technology has evolved and unbundled over time, and clearly illustrates the products-processes-services lifecycle model we propose. In the early years, where particular technology platforms (such as the mainframe computer) dominated, university researchers and company engineers both paid more attention to product or system design issues than to process issues, such as *how* to construct a particular software program to run on a particular hardware platform or operating system. The initial concerns were thus more about *what* systems programs or applications to construct, or perhaps what tools to develop to help build the programs.

Software also represents an industry where there have been multiple periods of major technological change, usually driven from underlying hardware platforms, such as from mainframe computer platforms to distributed computing using workstations and personal computers, but more recently driven by changes in software-based platforms, such as when Microsoft introduced an inexpensive enterprise version of Windows for client-server computing (Windows NT, first released in 1994) or inexpensive Internet technology suitable for individuals and businesses (first made possible with Netscape's Navigator browsers, released in 1994-1995) as well as more recent wireless and web services. As these technological advances have appeared, researchers and companies alike have often shifted their concerns back to *what* systems programs and applications to design before taking up the challenge again of *how* to design the new software more efficiently. This is consistent with Anderson and Tushman's (1990) idea that technological discontinuities restart the industry life cycle.

The first electronic computers, introduced for military applications during the latter stages of World War II, had no software as we know it today. The machines required hard-wired instructions and data that programmers inputted through setting switches representing machine language (zeros and ones). Computers that stored programming instructions as well as data within the computer's memory first appeared in the early 1950s for commercial applications, although most companies sold "systems" that bundled hardware and software, as well as programming and support services. In fact, the term "software" did not come into general use until around 1958-1960.[3]

Several independent software companies appeared during the 1950s, especially in the United States, to build one-of-a-kind defense systems and large-scale commercial applications, such as the SABRE airline reservation system, designed by IBM and its subcontractors between 1954 and 1964. As computer hardware evolved during the 1960s, it became possible to create more elaborate software programs. By the later 1960s, demand for computers had outstripped the ability of user IT departments as well as IBM and other computer hardware manufacturers to provide all the needed software, which was largely custom-built for each user. The standardized software products business emerged slowly during the 1960s as IBM and other companies were able to identify common user features for widely used applications and package them for multiple customers, such as in the insurance industry.

Another key event for the products business came after 1964, when IBM began introducing the System/360 family of compatible mainframe computers. This architecture became the industry standard and inspired IBM (which had about 80 percent market share) and other firms to write new operating systems and applications software. But an independent software products business did not really take off until IBM, under anti-trust pressure, decided during 1968-1970 to unbundle hardware from software and services, at least to some extent. IBM's unbundling mostly applied to *applications* software, and this is where most new opportunities arose for new software products (such as for databases).

During the 1960s, we can also see a rising concern with the *process* of programming. Once IBM technology became a standard for mainframe computing, IBM and other companies as well as university researchers began to pay more attention to how to make computer programs and manage projects more efficiently and predictably. In particular, IBM's difficulties in writing the new operating system for System 360 mainframes led to a broad recognition that writing software and managing large projects required special efforts and knowledge. Programmers and managers at IBM as well as at other companies and universities soon created a new field called "software engineering" in the late 1960s and early 1970s. The goal was to bring some industrial discipline, methods, and computer-aided tools to software development.

These efforts in software engineering resulted in many process innovations, especially during the 1970s. We can see the emergence of techniques such as structured programming (top-down hierarchical design, progressively hiding lower-level details from higher levels), formal methods (a math-based approach to defining and testing system requirements or specifications), and the life-cycle model for project and product management (incremental releases and periodic maintenance enhancements). Other process innovations included techniques ranging from design and code reviews (sometimes called "inspections") to collecting statistical data on projects to improve schedule estimates and quality analysis. These and other approaches would later become more refined in Japanese software factories operated at Hitachi, NEC, Fujitsu, and Toshiba, largely during the 1970s and early 1980s for building mainframe applications and systems software. We can also see this process focus culminating in a set of best-practice recommendations from the Software Engineering Institute (SEI), established by the U.S. Department of Defense in the mid-1980s and based at Carnegie-Mellon University as well as in the emergence of large-scale factory-like facilities for custom programming in India (Cusumano, 1991, 2004).

Most of the efforts in software process targeted mainframe computer systems and applications. The personal computer, first introduced as a kit in 1975, began a new era of mass-market products, coupled with cheap hardware, especially after Apple and IBM introduced personal computers in the late 1970s and early 1980s. Again, with the new technology, we see a shift back to *what* new products to design. In the first couple years of the PC software business, the most prominent companies identified new horizontal markets – operating systems and programming tools – so that buyers of the PC kits could make the machines do something useful. As in the mainframe era, technical improvements in PC hardware, and new enabling software technologies, led to new hardware products and then new software applications.

In fact, PC software resembled many new mass markets – ranging from automobiles in the early 1900s to mainframes in the 1950s. There was a flurry of firms in the first few years until a standard platform or "dominant design" emerged. Microsoft defined the key systems software standards while Intel defined the key hardware platform standards. IBM, which introduced its own PC in 1981, initially acted as the system architect, until being replaced in this leadership role by Intel, Microsoft, and a combination of other firms during the later 1980s. PC technology also underwent major shifts in design technology. One was the diffusion of graphical programming from the Apple MacIntosh during the later 1980s and then Microsoft Windows during the 1990s. Another major transition that software providers and users are still adapting to is the Internet.

The Internet again generated demand for new types of software products – a variety of systems, communications, and server software, as well as new applications and content management tools. Netscape and a host of other new companies, as well as Microsoft and IBM, developed a variety of products to create Web sites and distribute content, email, and applications over the Internet. Sun Microsystems developed Java as a programming language especially well-suited for building applications based on servers and accessed through Internet browsers. In the Internet frenzy, there was often little concern for *how* to build the new software and a lot of concern with how to get products out to market fast – "on Internet time" (Cusumano and Yoffie 1998).

In recent years, however, Microsoft and other firms as well as researchers in the industry have again turned from a focus on *what* new products to design to *how* to design the new generations of software more effectively. There has been a rising interest in "iterative design" and project management approaches to allow software developers to handle rapid change more easily (Cusumano et al., 2003). In addition, the "software factory" idea has again come into prominence, this time displaying more concern with how to construct more easily reusable modules and design patterns, as well as how to generate better code automatically.[4]

In short, this brief history of the software industry shows that each major technological change prompted the emergence of an initial period when firms focused on product innovation, followed by a period when firms focused on process (software engineering). That is, the software industry apparently conforms to the traditional lifecycle theory. But the shift from product to process is not the whole story. As we see below, services also play an important role in the later stages of the industry lifecycle.

Software Technology and Software Services

The layered structure of software systems, especially for enterprise users, as well as the changes in technology platforms over time, has made the software industry especially ripe for services. Different layers in the stack represent subsystems that can change at different rates. But all the layers must work together in order to function. As a result, if a particular market advances at a faster rate than other levels in the stack, it must still integrate with the older versions to work. Over time, standard interfaces and integrating technologies such as XML have reduced this burden, but the market structure and technical reality of software systems helps sustain this need. In addition, customers tend to slowly replace technology to the extent that critical applications use the old technology (Bresnahan and Saloner 1997), which generates additional need for integration services. Services have also become more important over time as a source of revenues and profits as hardware prices and then software product prices declined with the advent of cheap personal computers and open-source programs.

For instance, in 1986 Oracle, a leading producer of database products, organized its own consulting group and began to hire executives with consulting backgrounds. Third party consultants began to offer system integration services in the mid-1980s to help customers integrate and manage their increasingly heterogeneous technology infrastructure (Kahn, et. al, 1989). In the early 1990s, consultants connected this client/server technology with the principles of business process re-engineering and started leveraging this new technology to radically reconfigure how manufacturers organized their business processes. In the case of the Internet, software firms began to offer new services that helped measure the value from implementing a system and got involved in establishing new Internet marketplaces, such as Covisint, to exchange goods within an industry. They also started experimenting with offering their products as a service, in which the application is hosted by a third party accessed by the customer.

Looking at the revenues of major enterprise application providers illustrates this steady shift toward service revenues during the late 1990s and early 2000s. In fact, many enterprise software companies *doubled or tripled* their revenues over time through accumulation of contracts for services, including maintenance, even if new product sales lagged behind. Software companies selling to large corporate customers can generally get up to a dollar in service revenues for every new dollar in software product revenues (called “license fees” or “subscription fees”) for the first year or so of a new sales contract. This is in addition to the 15 to 25 cents in annual maintenance fees for every dollar of license fees for the period that the customer uses the product (Cusumano 2004).

Many enterprise software companies also offer *perpetual* licenses to their customers – that is, the customer, as long as it pays the agreed-upon annual maintenance fee, has the right to upgrade to new versions of the software, with an occasional special charge, for example, to accommodate hardware platform changes or expansion of usage rights. Therefore, over the lifetime of many enterprise software products, 70 percent or more of the total cost to a customer may come from services and other maintenance fees and only 30 percent from the original product sale (Hoch et al. 1999). The percentage from maintenance can become especially high for products in usage for a decade or more. If a software company sells a lot of products early on but then fails to keep up a high rate of new sales to new customers, it will inevitably become a hybrid provider as indicated in Figure 2 and eventually see the majority of its revenues shifting toward services and maintenance coming from the installed base.

We can see some of these trends in the financial data reported by leading enterprise software companies. Figure 5 compares the products and services (including maintenance) revenues of Oracle and Siebel, the leading producer of customer relationship management software acquired by Oracle in 2005. The top graph for each company shows the revenues in absolute dollars and the bottom graph compares the percentages of revenues coming from products versus services. Services at Oracle were approximately 40 percent of total revenues in 1992 and only 5 percent of Siebel revenues in 1995. After the “crisscross” occurred in 1997 for Oracle, services rose to approximately 65 percent of total revenues by 2002. At Siebel, product revenues collapsed in 2002 and fell below services revenues for the first time. By 2002, services and maintenance had risen to nearly 60 percent of revenues.

Quantitative Analysis

To explore how well the conceptual model described in Figure 1 applies to a broader set of companies, we analyzed the role of services in the pre-packaged software products industry. Pre-packaged software (SIC code 7372, NAICS 51121) includes firms that sell discrete programs consisting of software code that when executed on a hardware platform performs a certain task, such as to automate a business process or display a video game. Software is a particularly good industry to study the role of services because of the aforementioned synergies between product and service revenue. Specifically, we collected a sample of publicly traded software firms over the time period of 1990-2004 and used a fixed effect time-series analysis to test the changes in service revenue contribution relative to firm, industry, and exogenous environmental factors derived from the various theoretical explanations. Using service contribution to total revenue as the dependent variable, the model form is as follows, where i denotes a firm and t denotes the time period in years.

$$Service\,Contribution_{it} = \alpha + \beta_1 Firm\ Effects + \beta_2 Industry\ Effects + \beta_3 Environment\ Effects + \varepsilon_{it}$$

Sample and Data Collection

Compustat is a widely used data source for academic analysis. However, even though Compustat provides business segment information, software firms often do not report product and service revenues as separate business segments. This means that Compustat often does not capture this distinction in revenue. Software firms do, however, typically break this information out in their 10-K annual reports to the U.S. Securities and Exchange Commission, which the Mergent Database captures. Therefore, we used the Mergent database for financial information.

We identified pre-packaged software firms as those listed under the pre-packaged software SIC code 7372 in 2002, when data collection began. This list includes firms who were acquired and went out of business prior to 2002 but existed in 1997. To correspond with the technology lifecycle of the software industry we focused our data collection efforts for the years 1990-2004. To supplement this list, we identified public pre-packaged software firms that were included in the Software 500 list during the years 2000-2003. Since Mergent only provides the previous 15 years, we conducted additional 10-K analysis to fill out as many years as possible for the firms in the 1990-2004 time frame. This resulted in 464 firms identified. One issue with Mergent is that, if a firm existed in 1990 but went out of business before 1997, the firm was not captured in this database. For example, IBM acquired Lotus in 1995 so Lotus was not included in the Mergent list. To correct this problem, we identified those firms in Compustat under SIC code 7372 whose last entry was between 1990 and 1997. This identified an additional 51 firms. Since data for these firms are not captured electronically, we collected 10-K information for as many firms as possible using microfilm records. This increased the total sample to 485 firms.

However, not all firms break out products and service revenues. This may be because they only provide products as in the case of Microsoft and Adobe for most of their histories or video games producers, or because the firms simply do not break out the revenue. For each of these companies, we carefully studied their business description in the recent 10-Ks to determine if there was a meaningful breakout of revenues, and assigned the revenues accordingly. For 89 companies, however, we were unable to determine what the proper breakout should be and excluded these firms from the final sample. Figure 6 represents a histogram of the percentage of revenues coming from products. Note that some firms in some years reported 100 percent product revenues or 100 percent services revenues. The median for the sample is approximately 60 percent product revenues and 40 percent services revenues (also see Table 3, below).

In addition, since the financial information is only available for when a firm was public and possibly a few years before the firm went public, it is possible that we have some censored data points. For example, if a company was founded in 1988 and when it went public in 1994 only reported the two prior years, the database would not include data for the years 1990 and 1991. To avoid this problem, we eliminated firms that went public in the period of 1990-1995, but we are missing data during that time period. The exception to this rule is cases where firms were established during this time period but we are only missing the first year of data. In many cases, firms do not post operating results in their first year because it was an incomplete year, and the first year of data, which the database includes, represents the first full year of operation. This reduces the sample to 215 firms. Eliminating over half of the firms introduces a different concern about bias if these excluded firms behave systematically differently than the sample. However, we tested for differences in summary statistics of key explanatory variables and did not see any significant difference.

Our data set does not represent the entire software industry as it focuses on publicly traded firms on a US stock exchange. This list excludes private firms and most software firms who only trade on international local exchanges. The main concern here is that omitting these firms could introduce bias into our results. However, in related research on the history of one large sub-segment of the market – manufacturing software systems – Kahl (2005) has not observed significant difference between private and public firms with respect to product and service revenue mix.

Another important issue with our data relates to the way we coded product versus service revenue. Software firms vary in how they describe and allocate revenue between products and services. We coded product revenues as those revenues which relate to the licensing or sale of a software product. The main criterion is whether the customer was buying a product and not the mode of how they were paying for it. For example, some firms sell their software as a service, meaning that firms pay a monthly fee based on some metric such as number of users or transactions, as opposed to paying for the product upfront in a lump-sum license fee. We would code this as product revenue. We coded service revenues as those revenues associated with maintenance agreements, training, implementation, custom development, and post-implementation technical support. Salesforce.com, for example, sells basic customer relationship management (CRM) systems as a monthly service, and also follows this methodology in its financial reports. It separates out monthly subscription revenue from professional services, although it does include basic technical support in the monthly subscription fees. All other revenue at the firms in our sample, such as hardware sales, if any, we coded as other revenue.

Measures

Dependent Variable. The main dependent variable of interest is service revenue contribution, defined as the percentage of total revenues coming from services. Service revenue includes all maintenance, training, support, and consulting services that software firms provide to customers. For service revenue contribution, we focused on the mix between product and service revenues, excluding other revenue such as hardware sales. Increasing service revenue contribution within a firm indicates greater emphasis of services relative to product sales. Thus, a firm may grow its service revenue contribution without growing overall services as product revenue declines. We categorized our explanatory variables into the different firm, industry, and environmental effects found in the existing literature.

Firm Effects. Firm level operating characteristics may offer alternative explanations in the growth and migration toward service revenues. First, firms may migrate toward service revenues not by choice but because their product business begins to deteriorate. Service revenue and contribution may grow not because of overall growth but because other parts of the business do not perform. To capture this effect, we measured product revenue growth in the prior period. Second, to capture the effect of the firm maturing, we measure the firm age. Firm age has been associated to differences in organizational structure and activities (Delacroix and Swaminathan, 1991), and the type and level of organizational competences (Levinthal, 1991). Lastly, we also control for the size of the firm in the form of the natural log of sales in the prior period. Firm size has long been associated with a firm’s ability to respond to changes in the environment (Meyer and Zucker, 1989, Levinthal and March, 1993).

Industry Effects. Differences in industry characteristics can result in differences in firms’s offerings and the resources needed to compete (Porter 1980). The pre-packaged software industry, SIC 7372, comprises several different product categories. For example, some firms such as SAP and i2 provide software to help firms automate business processes; others, such as BMC or WebMethods, provide infrastructure software that help optimize hardware platforms and support higher level applications. In essence, separate businesses have formed at each level of the software stack described in Figure 4. We categorized SIC 7372 firms into one of 8 broad product categories described in Table 2. These categorizations were based upon product descriptions, competitive analysis in S-1s and 10-Ks, as well as industry analyst categorizations. Some firms, such as Microsoft and Oracle, sell products into multiple segments; in these cases, we used the product category in which the firm derives most of its revenue for its industry classification. Based on this criterion, no firm switched industries. As noted earlier, those firms in the games industry and those firms that did not have a clear industry category were not included in the sample. For each industry, we identified the founding event or firm as well as the date of introduction of client/server and Internet products.

We used this information to calculate industry measures. The introduction of a new technology within an industry is a significant event that can change the industry's competitive dynamics (Christensen, 1992) and may create service opportunities. As noted in the brief history of the industry, during the time period that we collected data, the Internet architecture began to replace client/server architecture. For each industry, a dummy variable captures the pre- and post-internet periods. Competition and industry maturity may also influence service revenue contribution. We used the ecological measure of density or the total number of firms at a given year (Hannan and Freeman, 1977). To calculate this number, we used firms from the entire sample, including those 30 firms collected from Compustat that we were unable to collect detailed financial information.

Finally, product pricing pressure due to commoditization may also influence service revenue contribution. From the U.S. Labor of Statistics, we collected the producer price index for the NAIC code that corresponds to SIC 7372. Unfortunately, these data only covers the period 1997-2004, but we were still able to use it in some of the analyses..

Environmental Controls. As noted, exogenous shocks such an economic recession may influence service and product revenue differently. During the period of observation, economic concerns certainly contributed to a slow down in spending during 2001-2003. In addition, there was the Y2K effect which positively impacted sales in 1999 as firms bought software to avoid conversion issues before 2000. To control for these exogenous shocks, we created several dichotomous variables, one for the recession in the years 2001-2003 and another Y2K for the year 1999.

Results

The total sample of public software product firms exhibits the crisscross effect and gradual transition to services as the dominant source of revenues (Figure 7). Table 3 provides summary statistics and correlation table of the variables. There is only low to mild correlation among the explanatory variables, which suggests no multicollinearity issues.

Table 4 provides the results of the regressions. Model 1 includes only the industry level effects and the environmental controls. The signs on the industry coefficients are consistent with the dominant design thesis. The negative coefficient on the density measure indicates that, as the number of firms decreases within the industry, the service contribution increases. According to the dominant design literature, after a dominant design emerges firms shift their competitive focus to cost-based competition and a "shake-out" begins (Utterback and Suarez, 1993), which decreases the number of competitors (density). Firms focus less on product development and more on processes and on finding mechanisms to "lock-in" their customer base. Adner and Levinthal (2001) suggest that during this period firms engage in non-price differentiation. It is in this phase of decreasing industry density that firms start to emphasize services as a source of revenue.

The positive coefficient on the internet discontinuity measure suggests that the service revenue contribution increases after the discontinuity sets in. This is consistent with our argument that discontinuities increase the opportunities for firms to offer new services to help customers manage through the transition from one technology to another. However, as seen in Table 4, the discontinuity measure is not significant in the other regression models, suggesting that the density measure has more longer-term significance relating to rising services revenues.

Model 2 supports the claim that firm level factors also influence the transition to services. The positive coefficient on age indicates that as firms mature they transition to services. In addition, as firms increase in size – a potential indication of an increasing installed base – service revenue contribution increases. Lastly, the negative coefficient on product sales growth indicates that as firms grow their product revenue, service revenue contribution decreases. This is not surprising given how we calculate service revenue percentage. The relationship between product revenue growth and service contribution is complex. On the one hand, product sales should lead to additional service revenues but, on the other hand, many firms whose product revenues are decreasing begin to focus on service revenues, which would suggest the negative coefficient. We tried several other measures of product growth, including expanding the average period from a one year lag to two years with similar results. Our results here suggest that the product growth measure reflects more about firm performance than the relationship between product growth and service revenue growth.

Model 3 combines both the firm level effects and the industry effects while controlling for environmental changes. The firm effects coefficients retain their direction and significance, while the measure of the technological discontinuity no longer is significant. The density measure also remains significant. These results suggest that firm maturity nd prior performance in conjunction with changes in the level of competition best explain the increase in service revenue contribution. In model 4, we added the producer price index for the time period of 1996-2004. Except for past product performance, this additional measure did not significantly change the results.

Conclusions and Further Research

We have demonstrated that, in the traditional software products industry, there clearly is a lifecycle for many firms that extends from product to processes and then to services, as indicated conceptually in Figure 2. As a whole, the focus of the industry has shifted over time from what products to design, to which processes to use to design those products more effectively, and finally to which services to provide in order to complement product sales (or even substitute them in some cases). In addition, as indicated conceptually in Figure 5, our results suggest that the transition from one technology to another, such as from client-server computing to internet-based computing, generates large amounts of service revenues for product firms as they help their customers move to the new technology or as product revenues from the older technology decline.

These results have implications for the strategic behavior of firms as an industry evolves. Prevailing lifecycle theories argue that firms should shift focus from product differentiation to cost-based competition as the industry matures. In these models, successful firms are those that can transition from a focus on R&D and product innovation to a focus on operational efficiency and processes. Our research suggests that, as the industry matures, there is also a shift from products revenue to service revenue. A change of strategic focus toward services implies an important change in organizational capabilities. This change is far from trivial, as suggested by the difficulties experienced by Sun Microsystems and HP in trying to imitate IBM's successful move to services. Growing the service part of the business requires careful attention to efficiency and process because services tend to be intensive in skilled labor and hard to scale. Emphasis on efficiency is paramount to having a profitable service business. On the other hand, and unlike what happens in the process stage of the industry life cycle, the service stage requires significant flexibility and customization on the part of the firm. Many services require the integration of new systems with a customer's legacy systems, or training customer personnel with different skill levels. These activities cannot be totally standardized. What makes this third stage challenging for firms is that it combines some of the characteristics of the first and second stages, namely flexibility and efficiency. Successful firms in stage 3 of the industry life cycle not only benefit from economies of scale, but also benefit from expanding their scope and offerings to customers.

In recent years, we have seen that public software companies may begin by emphasizing products but most will end up with more revenues from services, including maintenance, than from new product sales, for a variety of reasons. Most importantly, services appear to be a major source of revenues and profits as firms mature. It follows that firms should want to manage the transitions in this lifecycle as strategically as possible and pursue new service opportunities, rather than simply allowing product revenues to deteriorate as market competition increases, free or cheap software products appear, or economic conditions worsen and customers simply stop buying new products.

Altough we have tried to be as comprehensive as possible in building our databases, the available data still have some limitations. In our analysis, for example, service revenues mix together annual “maintenance” payments, which provide the customer with interim quality releases as well as version upgrades and some technical support, with more labor-intensive and lower-margin services such as customization, installation, consulting, and training. Maintenance revenue more closely resembles recurring product subscription revenues, except for bundled technical support, but companies can recognize these revenues only as they deliver the upgrades, patch releases or support. Furthermore, maintenance revenue is directly tied with the sale of a product and may behave differently from the other services. Ideally, we would like to separate out maintenance from the other services, but software product firms do not consistently break out this information. Despite this limitation, in our sample, several firms do break maintenance revenue out over a consistent period of time, resulting in over 500 data points. Our preliminary analysis on this suggests that the results in this paper will hold even after considering this further breakdown of the data.

Another avenue for improvement is a more detailed study of the impact of technological transitions on the generation of services or new services firms as opposed to product revenues and product firms. In our results, the Internet discontinuity measure is associated with an increase in service revenues; however, it was not significant in the model which combined both firm level and industry level effects. Services do seem to play a role in technological transition, especially for incumbents who try to assist their customers make the technological transition. However, it is not entirely clear that incumbent product firms may be able to appropriate service revenues during the transition as other industry participants such as third party service firms will compete for these service dollars. The technology strategy literature also suggests that significant technological transitions like the Internet attract new entrants who may focus on growing product revenues in order to establish market share. An apparent tension between focusing on service and product revenues arises in many technological transitions which warrants further systematic research.

Further research could also shed light on the dynamic relationship between products and services in software and other industries. Ideally, one could define business models that combine products and services in the software products industry more precisely in order to examine the impact of different models on firm survival and performance. It will be also interesting to extend the analysis to include other firms in the software industry, not only pre-packaged software. Another obvious extension is to consider a much broader sample of firms than discussed in this paper in order to study if this shift to services and away from traditional products appears in other industrial sectors as well. Finally, to help product firms manage the transition to services, further research could look at the processes by which firms acquire the new required capabilities, specifically, how existing services firms noted for innovation and efficiency manage the process of service R&D and new service creation.

References

Adner, R. and D. Levinthal 2001. Demand heterogeneity and technology evolution: implications for product and process innovation. Management Science **47**(5): 611-628.

Andersen, Eugene W., Claes Fornell, and Roland T. Rust 1997. Customer satisfaction, productivity, and profitability: differences between goods and services. Marketing Science, vol. 16, no. 2, pp. 129-145.

Anderson, P. and M. Tushman 1990. Technological discontinuities and dominant designs: a cyclical model of technological change. Administrative Science Quarterly **35**(4): 604-633.

Brandenburger A. and B. Nalebuff 1996. Co-Opetition. Doubleday: New York.

Bresnahan, T. and G. Saloner 1997. Larger firms' demand for computer products and services: competing market models, inertia, and enabling strategic change. Competing in the Age of Digital Convergence. D. Yoffie. Harvard Business School Press**:** Boston, 265-300.

Campbell-Kelly, M and William Aspray 1996. Computer: A History of the Information Machine. Basic Books:New York.

Christensen, C. M. 1992. Exploring the limits of the technology S-curve, Part 1. Production and Operations Management Journal **1**: 334-357.

Cusumano, M. A. 1991. Japan's Software Factories. Oxford University Press: New York.

Cusumano, M. A. 2004. The Business of Software. Free Press: New York.

Cusumano, M. A. et al. 2003. Software development worldwide: the state of the practice. IEEE Software, November-December 2003, vol. 20, no. 6, pp. 28-34.

Cusumano, M. A. and D. B. Yoffie 1998. Competing on Internet Time: Lessons from Netscape and the Battle with Microsoft. Free Press: New York.

Delacroix, J. and A. Swaminathan. 1991. Cosmetic, speculative and adaptive organizational change in the wine industry: a longitudinal study. Administrative Science Quarterly, 331-661.

Gadiesh, Orit, and James L. Gilbert 1998. Profit pools: a fresh look at strategy. Harvard Business Review, May-June, pp. 139-147.

Hannan, M. and J. Freeman 1977. The population ecology of organizations. American Journal of Sociology **82**:929-964.

Heskett, J. L., W. E. Sasser, et al. 1997. The Service Profit Chain. Free Press: New York.

Hoch, Detlev J. et al. 1999. Secrets of Software Success. Harvard Business School Press: Boston.

Kahl, S. 2005. Technological change, usage patterns, and industry evolution. Unpublished Working Paper, MIT Sloan School of Management: Cambridge, MA.

Kahn, M., Kugel, K., and Browne, A. 1989. Systems Integration: Tying all the Pieces Together, Part 2. International Data Corporation: Framingham, MA.

Knecht, Thomas, Ralf Leszinski, and Felix A. Weber 1993. Memo to a CEO: making profits *after* the sale. McKinsey Quarterly, Winter, no. 4, pp. 79-86.

Klepper, S. 1996. Entry, exit, growth, and innovation over the product life cycle. American Economic Review **86**: 562-583.

Klepper, S. 1997. Industry life cycles. Industrial and Corporate Change **6**(1): 145-181.

Levinthal, D.A. and , J. G. March. 1993. The myopia of learning. Strategic Management Journal **14** 95–112.

Levinthal, D.A. 1991. Organizational adaptation and environmental selection—interrelated processes of change. Organization Science **2** 140–145.

Mansharamani, Vikram 2005. Towards a theory of service innovation: an inductive case study approach to evaluating the uniqueness of services. Unpublished Master of Science thesis, MIT Sloan School of Management: Cambridge, MA.

Meyer, M., L. Zucker. 1989. Permanently Failing Organizations. Sage: Newbury Park, CA.

Milgrom, P. and J. Roberts 1990. The economics of modern manufacturing: technology, strategy, and organization. American Economic Review **80**(3): 511-528.

Oliva, R. and R. Kallenberg 2003. Managing the transition from products to services. International Journal of Service Industry Management **14**(2): 160-172.

Oster, Sharon M. 1994. Modern Competitive Analysis. Oxford University Press: New York.

Porter, Michael. 1980. Competitive Advantage. Free Press: New York.

Quinn, J. B. 1992. Intelligent Enterprise. Free Press: New York.

Rosenberg, N. 1983. Inside the Black Box: Technology and Economics, Cambridge University Press: Cambridge, UK.

Suarez, F. and J. Utterback 1995. Dominant designs and the survival of firms. Strategic Management Journal **16**(6): 415-430.

Teece, D. 1986. Profiting from technological innovation: implications for integration, collaboration, licensing and public policy." Research Policy **15**: 285-305.

Tushman, M. and P. Anderson 1986. Technological discontinuities and organizational environments. Administrative Sciences Quarterly **31**(3): 439-465.

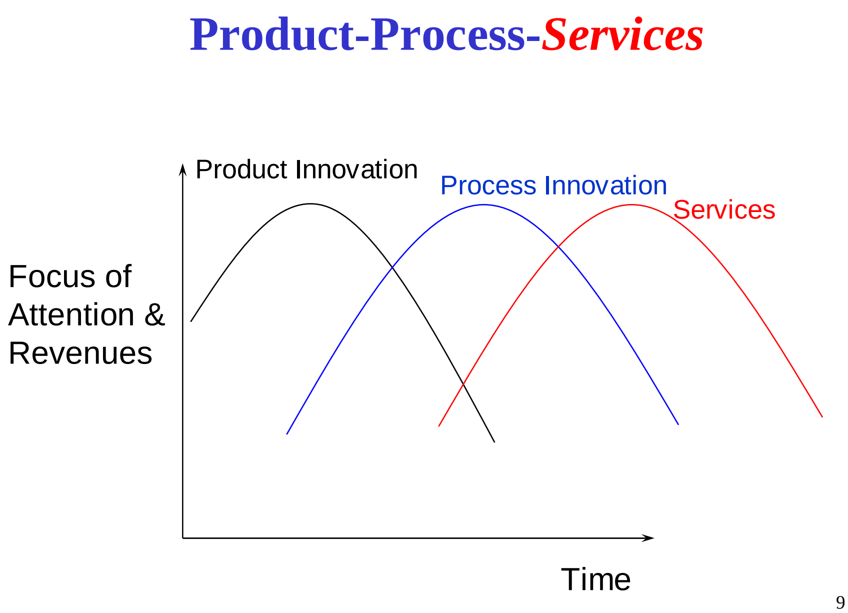
Utterback, J. 1994. Mastering the Dynamics of Innovation. Harvard Business School Press: Boston.

Utterback, J. and W. Abernathy 1975. A dynamic model of process and product innovation." Omega **3**(6): 639-656.

Utterback, J. and F. Suarez 1993. Innovation, competition, and industry structure. Research Policy **22**: 1-21.

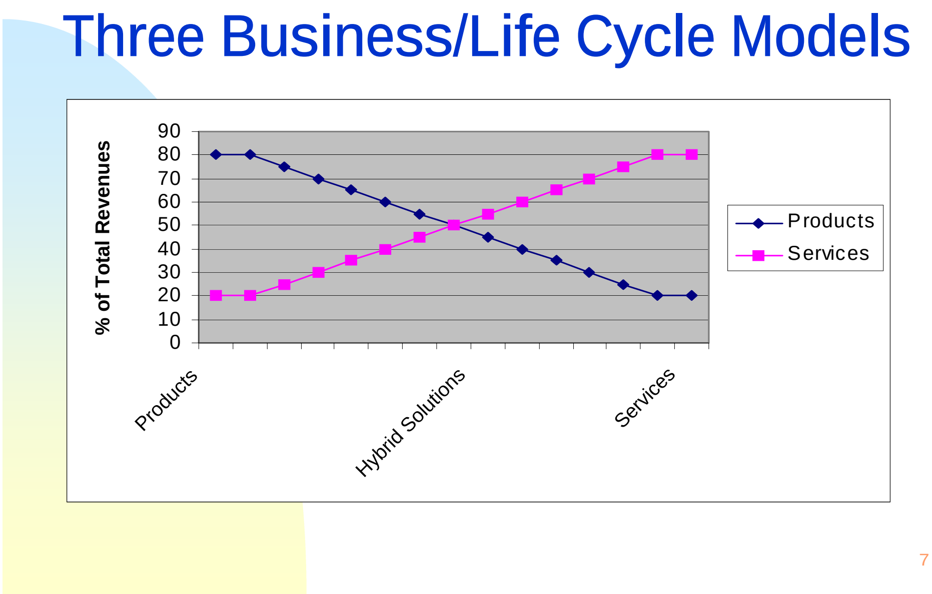
Von Hippel, E. and M. Tyre 1995. How learning by doing is done: problem identification in novel process equipment." Research Policy **24**(1): 1-12.

Figure 1: Enhanced Product-Process Industry Evolution Model



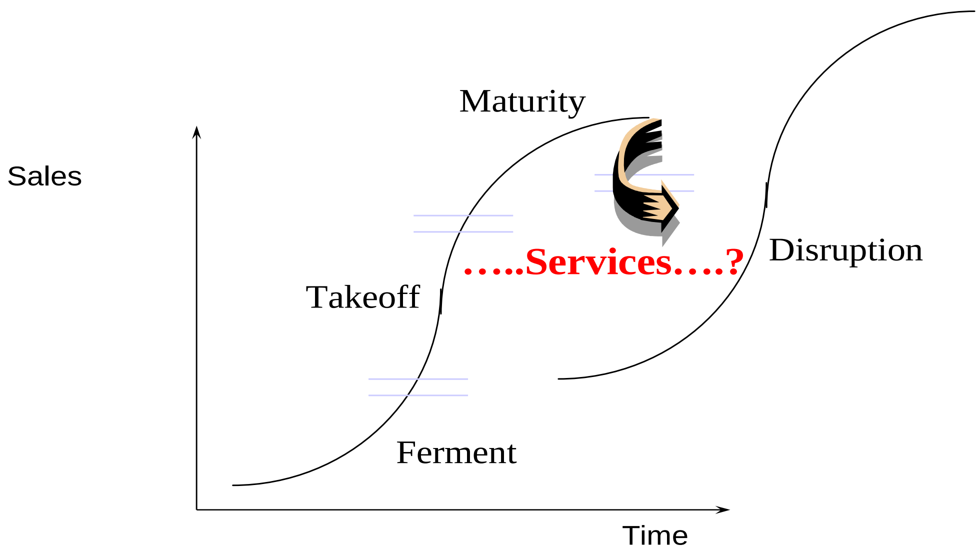
Source: Adapted from Abernathy and Utterback 1978.

Figure 2: Lifecycle and Business Models for Software Product Companies



Source: Cusumano 2004, p. 26.

Figure 3: Enhanced S-Curve Model of Industry Lifecycles and Disruption

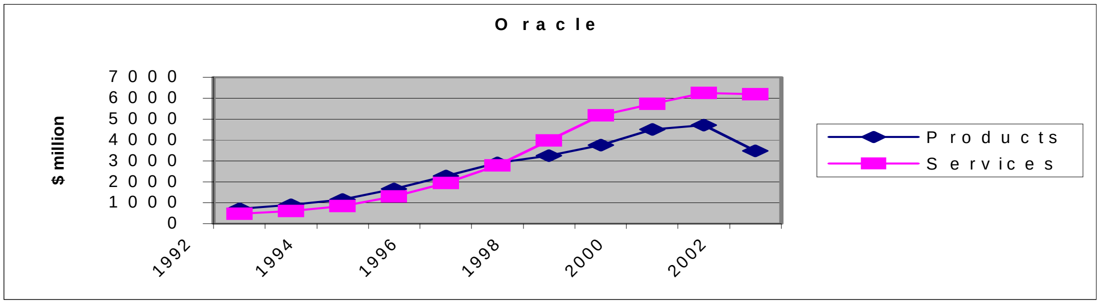


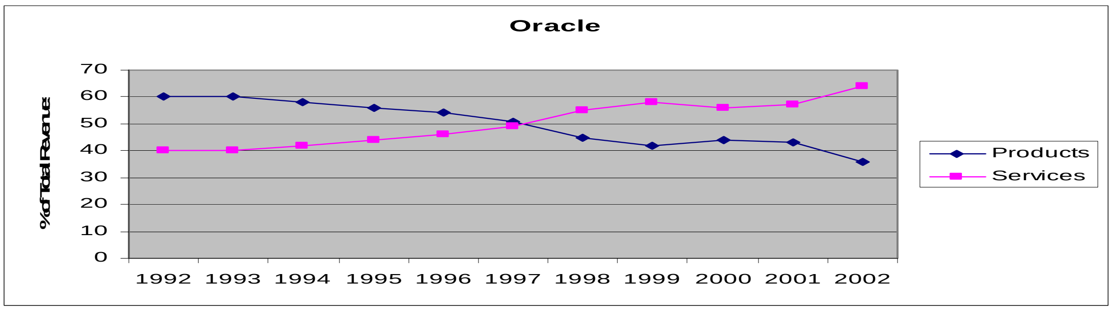
Source: Adapted from Christensen 1992.

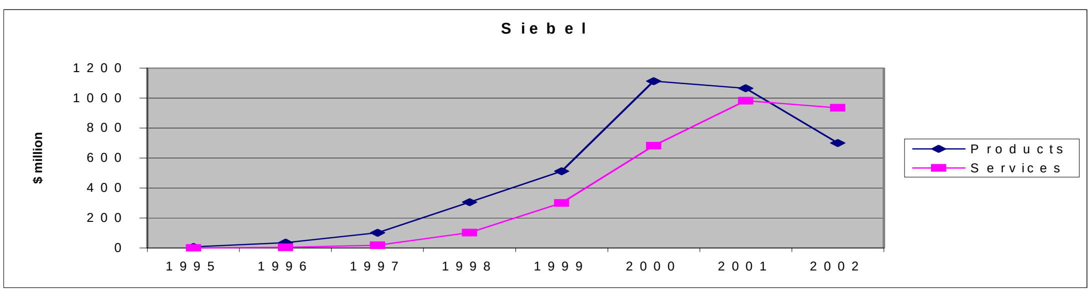
Figure 4: The Software Stack

	Application Technology Stack	Function	Vendor Examples
	Applications	Manage business processes	SAP, Microsoft, EA
	Development Tools	Language and methodologies to build applications	PowerSoft, Sybase, Microsoft, Sun
	Database	Manage data and optimize database	Oracle, IBM, Microsoft, Sybase
	Communications	Networking protocols, transfer data, transaction processing	BEA, Tibco, Novell
Software ↑	**Operating System**	Interface software with hardware	Microsoft, IBM, Unix, Linux
	Hardware	Execute software instruction	IBM, Sun, HP, DEC, RIM, Nokia

Figure 5 Products vs. Services and Maintenance Revenue Analysis
Source: Cusumano 2004.







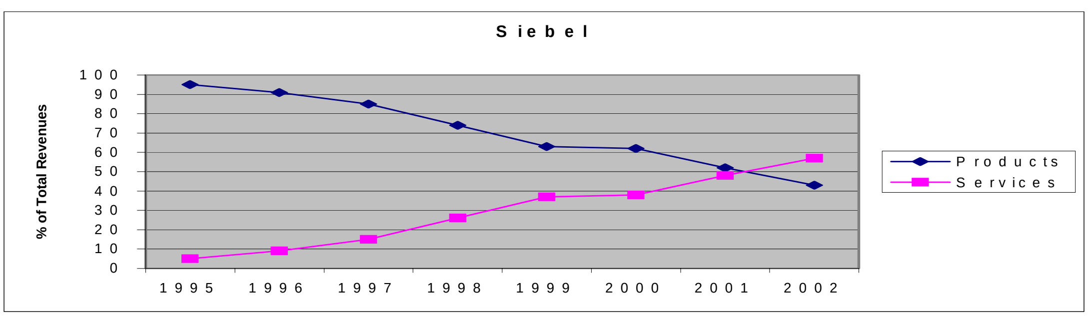


Figure 6 Histogram of the Sample

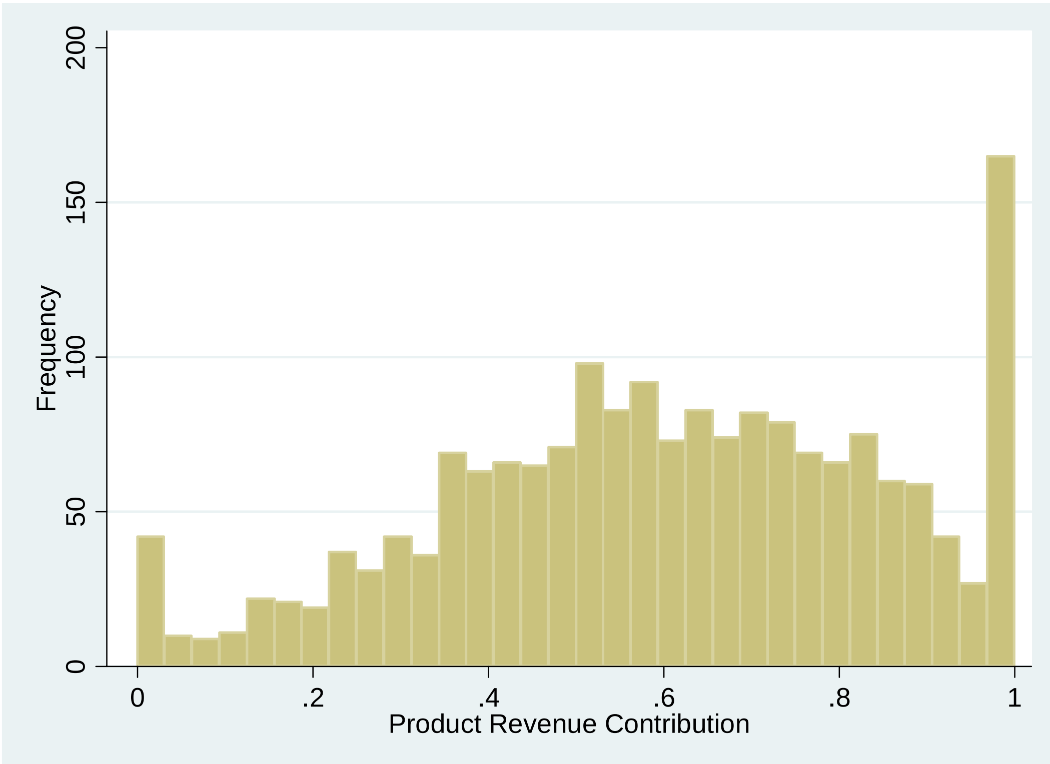
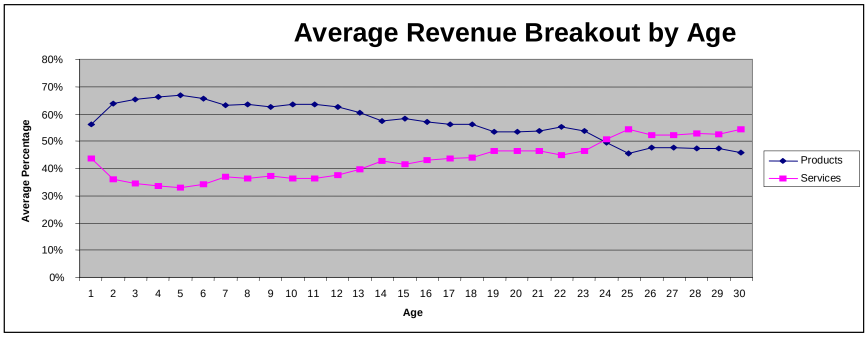


Figure 7: Criss-Cross Graph of Services Percentage by Year and Firm Age



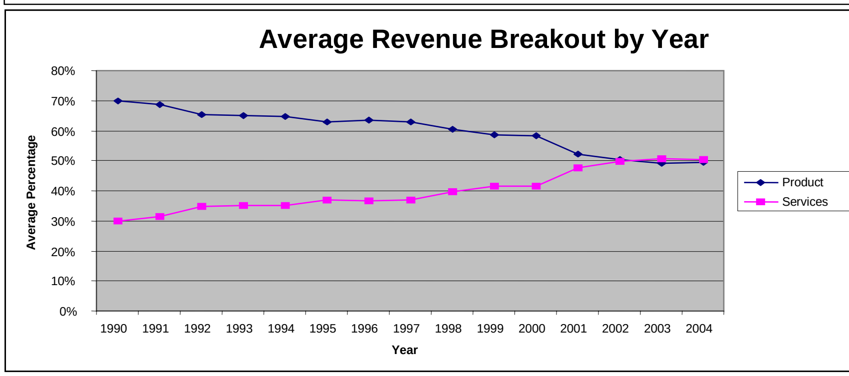


Table 1: Revenues and Margins for Selected Technology and Manufacturing Firms

Company	*2003/4 Revenues ($ billion)*	*Services Gross Margins(%)*	*Products Gross Margins (%)*	***2003/4 Services %***	***1995 Services %***
IBM	$ 89.1	25	28 (h/w)	**48**	**18**
Sun	11.2	38	42	**34**	**--**
H-P	79.9	23	25	**19**	**14**
Cisco	22.9	67	69	**16**	**--**
EMC	6.2	51(i.s.)	43 (i.s.)	**22**	**--**
GE	134.2	37 (-finan)	26	**30 (62)**	**14 (52)**
GM	185.5	9	0.4	**16**	**8(-EDS)**

Notes: EMC gross margin numbers refer to its information system products. GE gross margin for services does not include financial services. The percentage of GE services revenues for 2003-4 and 1995 in parentheses includes financial services. The 1995 services percentage figure for GM excluded Electronic Data Systems, an IT services firm that GM acquired and then divested.

Source: Compiled from annual 10-K reports.

Table 2 Industry Categories within SIC 7372, with significant product dates

	Industry Date	Reason	Client/ Server	Reason	Intern et	Reason	Examples
Bus. Apps	1968	MSA founded	1987	Client/Server business apps begin to emerge (e.g. PeopleSoft	1995	Internet-Arch. Firms begin to emerge (e.g. Vignette)	SAP, Siebel, PeopleSoft
Business Intelligence/ Content Mgmt	1970	IRI Releases OLAP Queries	1989	BRIO, Microstrategy Founded	1995	Vignette and other Internet Apps founded	Business Objects, Informatica, Documentum
Multimedia & CAD/CAM	1963	SKETCHPAD released	1981	PC Based CAD/CAM	1995	Internet release Products	PTC, Dassault
Database	1968	Cullinet Founded	1986	1st commercial release of distributed database management systems (IDC #3079)	1996	Oracle's release of web-enabled database (Oracle Website)	Oracle, Informix
Operating Systems	1970	Amdahl	1983	AT&T offers standardized Unix product (IDC 2974)	1995	Windows 95 Released	Microsoft
Networking	1966	IBM Releases IMS	1984	x/Open and TCP/IP standards war gets formalized (IDC #3533)	1994	Web Apache Server invented; Netscape founded	Webmethods
Games	NA		NA		NA		
Other	NA		NA		NA		

Table 3 Summary Statistics

Variable	Obs	Mean	Std. Dev.	Min	Max
Service %	1718	.4028428	.2510799	0	1
Age	2016	12.54911	8.590177	1	48
Prod Gwth Lag	1344	1.43123	9.012836	-.9578317	151.72
Ln Sales Lag	1822	10.58418	2.25975	.6931472	17.28748
Internet Prod	2016	.7589286	.4278397	0	1
Density	2016	79.36756	43.85064	11	151
Price Index	1117	99.40367	2.038753	94.3	101.4
Y2K	2016	.0793651	.2703748	0	1
Recession Dum	2016	.1795635	.3839187	0	1

Correlation Table:

	Serv %	Age	Prog G L	LnSale L	Internet	Density	Price In
Serv %	1.0000						
Age	0.2161*	1.0000					
Prod G L	-0.0914*	-0.1652*	1.0000				
Ln Sales L	0.0908*	0.3931*	-0.1572*	1.0000			
Internet	0.0883*	0.0277	-0.0120	0.0797*	1.0000		
Density	0.2465*	-0.0151	0.0679*	-0.2499*	-0.0602*	1.0000	
Price In	-0.1461*	-0.1409*	0.1087*	-0.1582*	.	0.2184*	1.0000
Y2K	0.0098	-0.0111	0.0329	-0.0313	0.1655*	0.0304	0.1503*
Rec	0.1712*	0.1311*	-0.0558*	0.1824*	0.2637*	-0.1577*	-0.1556*

	Y2K	Rec
Y2K	1.0000	
Rec	-0.1374*	1.0000

* = significant at .05 level

Table 4 Regression Results

	1	2	3	4
Dependent Variable	Service %	Service %	Service %	Service %
Firm Effects				
Age		.019 *** (.0013)	.017 *** (.0018)	.017 *** (.0048)
Product Growth Lag		-.001 *** (.0004)	-.001 ** (.0004)	-.000 (.0005)
LN Total Sales Lag		.009 ** (.0043)	.010 ** (.0043)	.013 ** (.0057)
Industry Effects				
Internet Products	.106 *** (.0103)		.012 (.0133)	Dropped
Density	-.003 *** (.0003)		-.001 *** (.0004)	-.002 *** (.0006)
Price Index				-.002 (.0030)
Environment Controls				
Y2K (1999)	.011 (.0128)	-.006 (.0110)	-.000 (.0111)	.009 (.0106)
Recession (2001-03)	.050 *** (.0110)	.022 ** (.0090)	.015 (.0093)	.018 * (.0103)
constant	.524 *** (.0280)	.042 (.0414)	.151 *** (.0543)	.366 (.3493)
Rho	.82	.84	.85	.88
F – Test	16.66 ***	24.92 ***	22.37 ***	19.34 ***
N	1718	1325	1325	831

Standard error in parentheses, *** p <.01, ** p< .05, * p < .10

Endnotes

---

[1] The authors wish to thank the Center for eBusiness at the MIT Sloan School for providing ongoing funding for these projects and Bill Crandall of Hewlett Packard for providing the initial seed funding.

[2] See Mansharamani (2005) for definitions and key references on the definition of services.

[3] This historical sketch is a summary of Cusumano 2004, Chapter 3, which is based on numerous sources, especially Campbell-Kelly and Aspray 1996.

[4] See information on Microsoft activities around software factories at http://msdn.microsoft.com/vstudio/teamsystem/workshop/sf/default.aspx.