ETLA
ELINKEINOELÄMÄN TUTKIMUSLAITOS
THE RESEARCH INSTITUTE OF THE FINNISH ECONOMY
Lönnrotinkatu 4 B 00120 Helsinki Finland Tel. 358-9-609 900
Telefax 358-9-601 753 World Wide Web: http://www.etla.fi/

Keskusteluaiheita – Discussion papers

No. 1201

Benedikt Gamharter* – Heli Koski**

– Tobias Kretschmer***

PRODUCT COMPLEXITY IN THE ADOPTION OF USER-CUSTOMIZED SYSTEMS

* Institute for Communication Economics, LMU Munich, and Centre for Economic Performance. Email: benediktg@gmx.net.

** The Research Institute of the Finnish Economy, ETLA. E-mail: heli.koski@etla.fi.
Heli Koski gratefully acknowledges financial support from the joint Networks, Services and Global Competition Research Program of ETLA (Research Institute of the Finnish Economy) and BRIE (Berkeley Roundtable on the International Economy). Financial support from Tekes (National Funding Agency for Technology and Innovation), the Federation of Finnish Technology Industries, and Nokia Corporation are gratefully acknowledged.

*** Institute for Communication Economics, LMU Munich, and Centre for Economic Performance. Email: t.kretschmer@lmu.de.

ISSN 0781-6847 14.10.2009

GAMHARTER, Benedikt **– KOSKI**, Heli – **KRETSCHMER**, Tobias, **PRODUCT COMPLEXITY IN THE ADOPTION OF USER-CUSTOMIZED SYSTEMS.** ETLA, The Research Institute of the Finnish Economy, Elinkeinoelämän Tutkimuslaitos, 2009, 24 p. (Discussion Papers, Keskusteluaiheita, ISSN 0781-6847; No 1201).

ABSTRACT: We study firm-level adoption of packaged software products of almost 4,000 UK sites between 2000 and 2003. We consider all software used in a firm as its software product and categorize it into related, but distinct subsystems: the core subsystem (Operating Systems), and Desktop and Enterprise Applications, which we define as peripheral subsystems. Adoption is studied across those subsystems. We find that larger firms tend to adopt and switch subsystems more often. We further consider three factors that may affect the adoption decision: product complexity, architectural innovation and competency scale. The complexity of existing peripheral sub-systems hinders switching with the exception of the UNIX operating system that becomes a more likely choice for firms with complex IT systems. UNIX OS is also adopted more often by firms with higher IT competency. Further, earlier generational expansion of peripheral subsystems generally decreases the likelihood of further adoptions. Architectural changes hinders adoption of some, but not all software applications.

JEL-Codes: L15, L86, O33.

Keywords: Software, diffusion, innovation, system products.

GAMHARTER, Benedikt **– KOSKI**, Heli – **KRETSCHMER**, Tobias, **PRODUCT COMPLEXITY IN THE ADOPTION OF USER-CUSTOMIZED SYSTEMS.** ETLA, The Research Institute of the Finnish Economy, Elinkeinoelämän Tutkimuslaitos, 2009, 24 s. (Discussion Papers, Keskusteluaiheita, ISSN 0781-6847; No 1200).

TIIVISTELMÄ: Tutkimme ohjelmistotuotteiden käyttöönottoa liki 4,000 isobritannialaisessa toimipaikassa vuosina 2000-2003. Yrityksen kaikki käytössä olevat ohjelmistot käsitetään ohjelmistotuotteena, joka jaetaan toisiinsa liittyviin, mutta erillisiin alijärjestelmiin: ydin(ali)järjestelmään (käyttöjärjestelmä) sekä työpöytäsovelluksiin ja yrityssovelluksiin, jotka määrittelemme oheis(ali)järjestelmiksi. Käyttöönottoa tutkitaan kyseisten alijärjestelmien tasolla. Aineistomme osoittaa, että suuremmat yritykset ovat taipuvaisia ottamaan käyttöön ja siirtymään uusiin alisysteemeihin muita useammin. Käsittelemme kolmea mahdollisesti käyttöönottopäätökseen vaikuttavaa tekijää: tuotteen monimutkaisuus, järjestelmän rakenteeseen liittyvät innovaatiot (architectural innovation) ja käyttöönottajan kompetenssi. Käytössä olevien oheisjärjestelmien monimutkaisuus hidastaa uusien alijärjestelmien käyttöönottamista. Poikkeuksena on UNIX-käyttöjärjestelmä, jonka valinta on todennäköisempi yrityksissä, joissa on käytössä monimutkainen tietotekniikkajärjestelmä. UNIX-käyttöjärjestelmä otetaan käyttöön useammin myös sellaisissa yrityksissä, joiden tietotekniikkakompetenssi on korkea. Tietotekniikkajärjestelmän rakenteeseen liittyvät muutokset hidastavat joidenkin, mutteivät kaikkien ohjelmistosovellusten käyttöönottoa.

JEL-koodit: L15, L86, O33.

Asiasanat: Ohjelmistotuotteet, diffuusio, innovaatio, järjestelmätuotteet.

1 INTRODUCTION

This paper studies the adoption of software in the United Kingdom (UK) from 2000-2003. We extend the literature on technology diffusion by developing an adoption framework based on firm-technology characteristics. By focusing on the software system currently used in the firm, we address the following central research question: *To what extent is a firm's adoption of new software affected by the existing bundle of software, and which characteristics of the current software bundle are particularly important?* Our key findings are that the complexity of the current software bundle and prior generational expansion of the peripheral subsystems play an important role in the adoption behaviour of firms. Specifically, switching to most software (i.e. Windows operating system, desktop software and enterprise applications) is less likely in firms with highly complex software systems already in place. However, firms with more complex software systems and higher IT competency as measured by IT intensity are more like to adopt the UNIX operating system. Our study is one of the first to explicitly recognize the fact that complex products are to a large part user-customized, which in turn may have important implications for the adoption of new system components.

The structure of the paper is as follows. We provide an overview of the relevant literature and link key insights from the diffusion and the innovation literature in Section 2. In Section 3, we give a brief account of the computer software market. We pay particular attention to the notion that software products cannot be viewed in isolation, but rather as related, but distinct elements of a firm's software landscape. In Section 4, we introduce our conceptual framework of software adoption and describe our data. Further details about the method, our results and their interpretation are provided in Section 5. Section 6 concludes with some final remarks and suggestions for further research.

2 EXISTING LITERATURE

This section introduces the key ideas of the paper by classifying existing studies in technological diffusion and introducing concepts initially developed in the innovation literature. We then develop a framework to study the diffusion of complex, user-customized systems.

Broadly speaking, the literature on the diffusion of new goods and services can be classified into two groups: single-technology and system-technology studies.[1] While the s-shaped diffusion curve is universally accepted in the literature, there is no clear consensus on the process that generates it. Geroski (2000), Stoneman (2002), Nelson et al. (2004) and Rogers (1995) review the different models of technology diffusion. Karshenas and Stoneman (1993), Zettelmeyer and Stoneman (1993) and Grajek and Kretschmer (2009a) provide comparative analyses of the different models. In our empirical implementation, we review the variables that have been found to affect diffusion speed.

2.1 Single-technology studies

Most technology diffusion studies consider single, mostly manufactured products or easily identifiable services.[2] The main thrust of these studies is to identify key adopter or firm characteristics that would have an effect on the likelihood and/or timing of adoption and therefore the speed of the diffusion process. More recent studies explicitly include the effect of an installed base has on diffusion speed, especially in the context of technologies with network effects. Lack of a sufficient installed base can delay adoption and, therefore, diffusion (Farrell and Saloner, 1986), although once critical mass is reached, diffusion can be rapid (Cabral, 1990, Grajek and Kretschmer, 2009b). Studies looking at the diffusion of single technologies are appropriate in contexts where adoption also takes place in isolation, i.e. adoption of a specific technology has no bearing on the propensity of adopting any other technology. Similarly, when network effects are predominantly direct, i.e. they originate from the direct interaction among users of a technology, so that the need to consider related technologies is limited. However, new technologies frequently do not appear in isolation, but as systems of complementary products, which we will discuss in the next section.

2.2 Adoption of system technologies

A system of complementary products frequently (or exclusively) used in conjunction with each other can have different implications for technology adoption. In the simplest case where both complementary products are used in a constant ratio (e.g. using *n* razor blades per razor per month), treating the

[1] We use the term “technology” as a catch-all for all goods and services.

[2] The range of products studied is enormous and we will not attempt a review of the existing studies. For a partial list of studies, see Geroski (2000), Stoneman (2002), or Rogers (1995).

system as a single technology will not bias results. However, if the ratio is not fixed (e.g. if Gillette has a significantly higher blades/razor ratio than Wilkinson), or if the adoption of one technology depends on the availability of the complementary product, measuring adoption of both components is necessary to uncover the true mechanisms behind the system's diffusion. This is especially relevant for technologies with indirect network effects, where interdependencies between hard- and software play an important role. Gandal et al. (2000), Nair et al. (2004), Gupta et al. (1999) and Claussen et al. (2009) are examples of studies where the cross- (or indirect) network effects are estimated. The key research questions in this strand of research are the strength and direction of indirect network effects between complementary goods. The products that these papers study, however, are still relatively simple since the technology and the nature of complementarities is determined by the producer: CD players require CDs, PDAs require software and Digital Television needs programming to be useful. That is, adopters still purchase a well-defined product. For many technologies used by firms or professionals, however, there is a significant degree of user-driven customization of the product or the bundle of products. The existing literature on technology diffusion has not adequately addressed this aspect of complex network technologies. In the following section, we introduce a framework that is helpful in organizing some of the effects on diffusion of such user-customized systems.

2.3 Adoption of user-customized systems

The IT hardware infrastructure used by a graphic design firm will most likely be very different from that of an international bank or a biotech company, and the software used by the firms will differ even more dramatically. Although most companies use IT, there are significant differences between users which may in turn affect the likelihood of upgrading to new generations of technology. However, it is the user who defines this technology or system. There is a complex relationship governing the diffusion of user-customized systems, as in the case of system technologies, but there are also significant adjustment costs arising from introducing new components. To our knowledge, there are no studies that explicitly look at the adoption of user-customized systems, although such a study will of course have much to borrow from the literature on mix-and-match products (Matutes and Regibeau, 1988; Matutes and Regibeau, 1992) and modular systems (Schilling, 2000, Henderson and Clark, 1990).

The main intuition from this literature is that the usefulness of a product depends on what other technologies are currently used in the firm. Kretschmer (2005) terms this phenomenon *internal complement effects* (ICE). The idea of ICE stresses the notion that it is not the general availability of complementary products (as the literature on indirect network effects typically assumes), but the actual usage of technologies in the firm that affects the adoption of a modular good. The dearth of current literature on the *diffusion* of such systems has led us to borrow from the literature on the *generation* of such modular products, i.e. the innovation of complex technologies.

The literature on innovation has long recognized that new products are flexible in their specifications of components and their linkages.[3] As an IT product is developed for mass production and marketing, individual components are changed and communication interfaces are defined and redefined. Henderson and Clark (1990) and Schilling's (2000) work on innovation in modular systems explicitly studies the *system architecture*, i.e. the relationship of different subsystems to each other and to the overall product. While it is intuitively appealing that subsystems are linked and that the nature of these linkages will determine the end product, operationalization of these concepts has often proved rather difficult, both on a theoretical and an empirical level. We therefore borrow from Gatignon et al's (2002) approach to describe and analyze complex innovations and adapt it to the study of user-customized system diffusion.

2.4 A classification framework for technological systems and their components

Gatignon et al.'s (2002) structural approach to evaluate innovation states that:

> […] innovation can be comprehensively described by distinguishing between product complexity (the number of its subsystems), the locus of the innovation in a product's hierarchy (core/peripheral), different types of innovation (generational and architectural), and the innovation's characteristics (incremental/radical, competence-enhancing, and competence-destroying) (Gatignon et. al., 2002, pp. 1104).

We follow Gatignon et al's approach and extend it by including product complexity as a fourth innovation characteristic (see Table 2-1). In their study, however, this framework is used to predict the origins of new product innovations, i.e. *supply* of a new product, rather than the *demand* for the entire product and its components. That is, we ask how likely a firm with a given system architecture (described by the characteristics in Table 2-1) is to adopt a particular new technology.

---

[3] For a detailed review of the earlier literature on innovation see for instance Utterback (1994) and more recently Christensen and Raynor (2003), Markides and Geroski (2005) and Moore (2005).

Table 2-1
Classification Framework

Characteristic		Definition
Locus of innovation (v)		'Core subsystems are those that are tightly coupled to other subsystems. In contrast, peripheral subsystems are weakly coupled to other subsystems (Gatignon et. al., 2002, p. 1106).'
Product complexity (i)		The product complexity increases in the number of sub-systems the product is made up of.
Innovation type	Generational consolida-tion/expansion (ii)	'Generational innovation involves changes in subsystems linked together with existing linking mechanisms (Ga-tignon et. al., 2002, p. 1106).'
	Architectural innovation (iii)	'Architectural innovation involves changes in linkages between existing subsystems (Gatignon et. al., 2002, p. 1106).'
Innovation characteristics	Competency destroy-ing/competenc y enhancing (iv)	'Competence-enhancing innovation builds upon and re-inforces existing competencies, skills, and know-how. Competence-destroying innovation obsolesces and over-turns existing competencies, skills, and know-how (Ga-tignon et. al., 2002, p. 1107).'
	Radicalness	'Incremental innovations are those that improve price/performance advance at a rate consistent with the existing technical trajectory. Radical innovations ad-vance the price/performance frontier by much more than the existing rate of progress (Gatignon et. al., 2002, p. 1107).'

Gatignon et. al. (2002) construct comprehensive scales to assess the characteristics set out in Table 2-1. In this paper, we do not follow their approach and instead use measures given in our data to proxy for these characteristics.

3 INDUSTRY REVIEW & PRODUCT FRAMEWORK[4]

The period 2000-2003 has been an exceptional time for the global computer software market. After the boom years of the late 1990s, following the emergence of the Internet and the commoditization of computer hardware, the market stalled at the turn of the millennium. Unsure about the effect of the Y2K switch, firms invested heavily into projects ensuring a smooth transition in the phase leading up to the millennium, with little investment otherwise. Market expectations were that the growth of the computer software industry would return to normal in 2000. In fact, the industry hoped for a revival of

[4] The primary sources for this section are the semi-annual publications of Standard & Poor's on the Software Market (Rudy, 2000-2004 and Bokhari, 2006).

IT spending to upgrade systems and infrastructure as corporate budgets were freed from the Y2K burden. As expected, the market did initially pick up in 2000 just before collapsing as the Internet bubble burst. Market conditions were difficult in 2001-2002 before improving in 2003.

3.1 Product Overview

We follow the definitions from International Data Group (IDC) and distinguish between the following product categories: Application Software, Application development & deployment Tools and System Infrastructure.[5] In 2002 the worldwide packaged Application market was US$83 billion, the Application development & deployment market was US$31 billion and the System infrastructure software market was US$47.7 billion (Rudy, 2000-2004).

Figure 3-1
Overview of the Packaged Software Market[6]

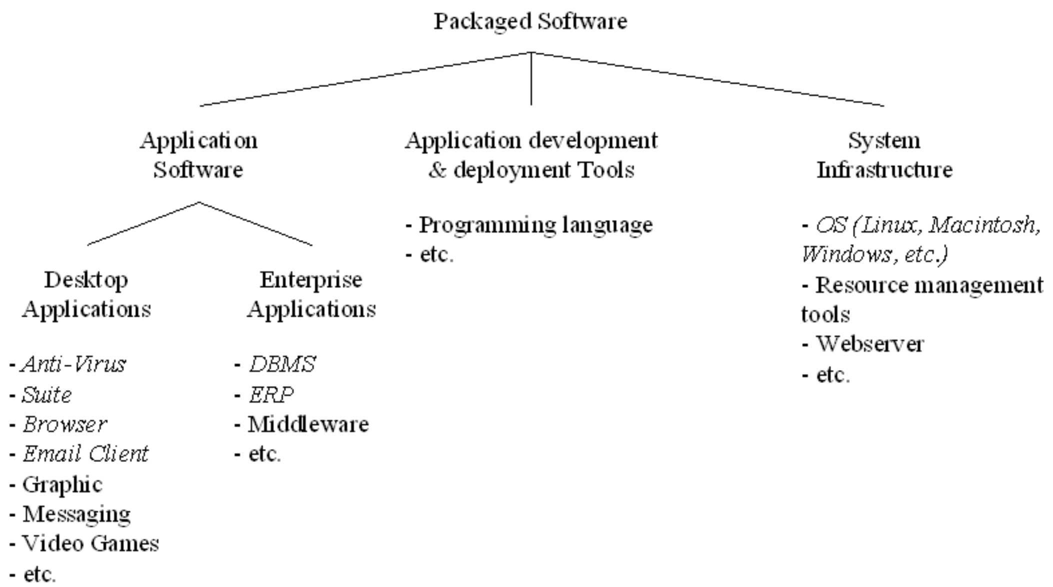


Figure 3-1 gives an overview of the product categories.[7] These distinctions, however, are not always clear-cut. For example, distinguishing different software product is made difficult by the fact that a) all

[5] IDC is one of the largest market intelligence firms covering the world-wide high-tech market (http://www.idc.com).
[6] The product groups used in this study are in italics.
[7] Note that we are ignoring custom-made software in this Figure. In general, however, any piece of software can be custom-made.

software firms integrate features of other products and actively push product consolidation;[8] b) advances of network technologies and integration protocols, standards or mechanisms make inter-product interaction increasingly seamless (e.g. .Net standard, XML or SOAP); and c) there is a long-term push to provide classical software products (which used to be sold via a licence or maintenance contract to users) as (Internet) services.[9]

We focus on professional software and especially look at Application Software and System Infrastructure. In particular, we focus on **Operating Systems (OS)** within the system infrastructure segment. In the application software market, we distinguish between **Desktop Applications (DA)** and **Enterprise Applications (EA)**. The products we study are highlighted in italics in Figure 3-1.

3.2 Product Framework

Given the strong (direct and indirect) network effects in the software industry, firms face a multi-technology adoption decision. Hence, firms need to consider not only the features of the product. Their adoption decision is also influenced by standards, compatibility concerns, availability of complementary goods and/or the current installed base, to list only a number of decision criteria that are not part of a single product (see Katz and Shapiro, 1986, and Gandal, 1995).[10]

A firm will purchase soft- and hardware to improve productivity and meet its business objective (Bresnahan and Greenstein, 1999; Brynjolfsson and Kremerer, 1996). We treat a firm as being composed of different tasks or functions (Kretschmer, 2004) and assume that software is chosen to match the respective task requirements. Assuming that there is a central decision maker in the firm or the site, e.g. a Chief Information Officer (CIO), decisions are made at the site level to optimize site performance and *do not view software applications in isolation*, but rather what we term a firm's *software architecture*. We argue that this architecture is made up of subsystems, i.e. the individual software products used by the firm. Hence, a firm faces a user-customized adoption decision.

---

[8] An extreme example of this is Oracle's acquisition of PeopleSoft.

[9] The trend of software as a service is currently accelerated by Google, which is targeting the corporate software market by positioning its consumer Internet services (e.g. GMAIL) to target corporate users (http://www.ft.com/cms/s/cb9bc46a-35e8-11db-b249-0000779e2340.html). This idea is not new, however. In 2000-2003, the industry partly already relied on application service providers (ASPs) to lease, host and manage software.

[10] All these features depend on a product *in relation* to others in the economy or the firm in question.

Figure 3-2
Software Product, Subsystem and Hierarchy

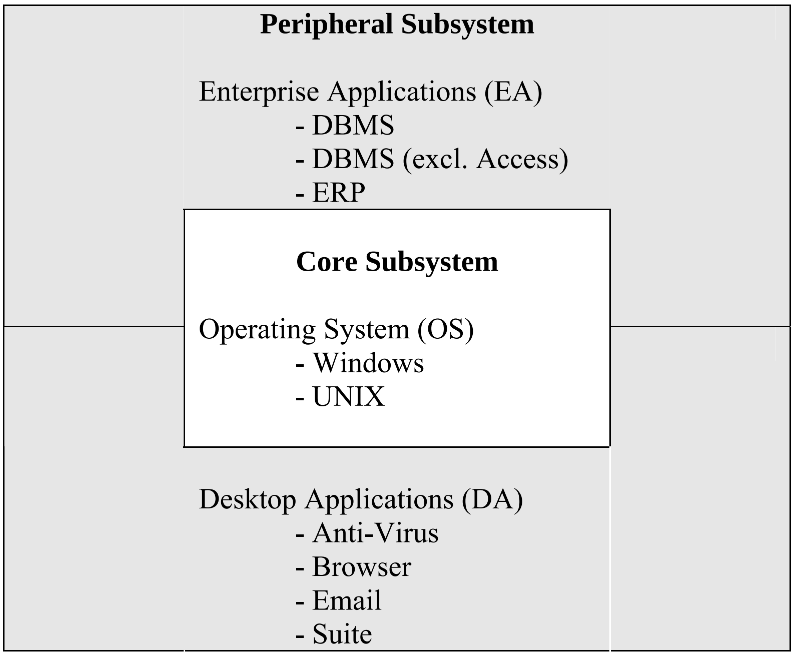


Figure 3-2 summarizes how we classify a firm's software product and also lists the individual products in our data. At the heart of the product are the core subsystems, which are characterized by multiple connections with other subsystems and their importance to the overall system. Specifically, a firm's OS connects to all subsystems and replacing a firm's OS has a cascading effect. Further, the software product is made up by of peripheral subsystems which can be added more freely to the system. We define two different types of peripheral subsystems: DA and EA. DA are generic and typically not customized, and used by various functions across the firm. EA are also peripheral subsystems but we classify them as more complex subsystems within the product hierarchy as they are mostly customized and affect numerous related systems and processes.

4 DATA AND CONCEPTUAL FRAMEWORK

4.1 Data

The data we use is from a dataset gathered by Harte-Hanks Inc. (HH). HH is an international direct and targeted marketing company that helps IT companies improve their marketing efforts. To service its clients, HH collects annual information on the IT stock of companies and government institutions across the UK.[11] On average, the firm collects information on roughly 16,000 sites.

This paper follows the data cleaning and panel data construction process as set out by Kretschmer (2005). See Appendix A and B for details on variable definitions and construction.[12] We eliminated observations with data inconsistencies (e.g. observations which do not report key variables (such as Operating Systems) or report implausible results). This cleaning process left 23,639 observations. We then balanced the panel with a total of 15,564 observations from 3,891 firms. Descriptive statistics and an overview of the data are presented in the next Section.

4.2 Software and Site Characteristics

We construct the following product hierarchy: three generic product categories (**PC_***) which are made up by product families (**PF_***). Product families are defined as groups of product substitutes. Therefore, product families are made up by individual products (**PD_***). Figure 4-1 summarizes the variables.

All product usage variables are dummy variables set to '1' if the particular site uses the particular software during any given year and '0' otherwise. This detailed data setup allows us to study the dynamics of product change (see Appendix A and Appendix B for details about the variables).

Figure 4-1
Product Variables

PC	Operating System (OS)	Desktop Applications (DA)	Enterprise Applications (EA)
PF	- Windows - UNIX	- Browser - AntiVirus - Email - Suite	- DBMS - ERP
PD	- 9 Operating Systems (Windows XP, etc.; see Appendix A for full list)	- 2 Browser - 5 Anti-Virus - 4 Email - 7 Suite (Internet Explorer, Outlook, Office 97, etc.; see Appendix A for full list)	- 7 DBMS - 5 ERP (Oracle, SQL Server, etc.; see Appendix A for full list)

[11] In this paper, we do not distinguish between firms and government institutions. The terms "firm" and "site" are used interchangeably and address the entire sample.

[12] The variables names follow a consistent logic. For instance, 'PF_ERP' refers to 'Product Family ERP'. When we want to refer to all variables of a category we can do so; for instance, 'PF_*' refers to all Product Families or 'IND_*' to all Industry Groups.

We constructed variables on our site characteristics along the following dimensions:

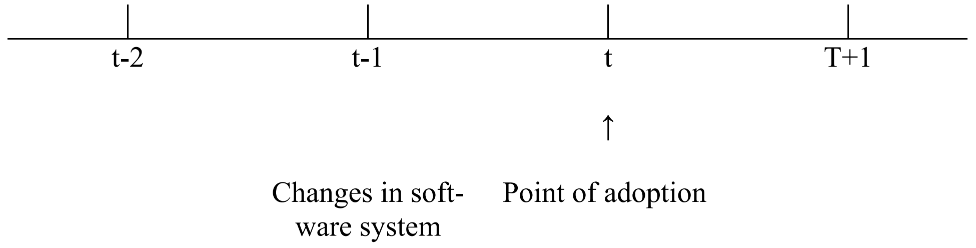
- **Industry groups**: The sites are classified based on their primary activity into three industry groups a) Service Industry (**IND_G_Service**; e.g. communication, transport, education), b) Manufacturing Industry (**IND_G_Manuf**; e.g. agriculture, mining, manufacturing) and c) Other (**IND_G_Other**; e.g. government). Furthermore, each site's primary activity is classified as belonging to one of the 24 UK Standard Industrial Classifications (SIC) of Economic Activity as defined by the UK Office of National Statistics;[13]
- **Firm size**: We include a variable (**LOG_Emp**) describing the size of the firm;[14]

4.3 Adoption Characteristics

In our adoption framework we bring together the innovation and product framework. We are interested in studying drivers of software diffusion and therefore understanding if current firm characteristics help estimate the probability of future adoption decisions, i.e. when studying sites adopting a new technology at time *t* we need to consider the characteristics of their software system at time *t-1*. The timeline in Figure 4-2 visualises our approach.

In addition to the three firm characteristics introduced above, we introduce Structural Change variables (**SC_***). The variables compare the year-on-year change of certain firm's software system characteristics. Hence, the t-1 variables are calculated based on the comparative results between *t-2* and *t-1*. Effectively, we study if past changes in the firm's software system have an impact on the likelihood of further changes.

Figure 4-2
Timeline



[13] Source: http://www.statistics.gov.uk/methods_quality/sic/default.asp and http://www.statistics.gov.uk/methods_quality/sic/methodological_guide.pdf.

[14] As discussed earlier, we do not want to explore/include the complexities of organisational design into our analysis. Hence, for the sake of simplicity, we do not differentiate between potential site types (e.g. headquarters, subsidiaries) and assume that all sites can influence their software adoption decision. We plan to address functional differences between sites in future work.

The change variables are calculated as set out in (4-1).

$$\text{(4-1)} \qquad SC\_*_{\Delta} = \log\left(\sum \frac{SC\_*_{T}}{SC\_*_{t-1}}\right)$$

Further, we introduce Product complexity variables (**PC_***) which are level variables. Table 4-1 summarizes how we construct the **SC_*** and **PC_*** variables for our regression analysis.

Table 4-1
Adoption Framework

<table>
<tr><th colspan="2">Characteristic</th><th>Measure (Variable)</th></tr>
<tr><td colspan="2">Dependent variables:
Locus of innovation</td><td>First, we study the effect of core subsystem replacement or enhancement. The paper distinguishes between Microsoft and UNIX products (SC_Locus_Core_MS and SC_Locus_Core_UNIX). Second, we study the effect if a new peripheral subsystem is introduced (SC_Locus_Peri_*) and study the various product families.
The dummy variables get value 1 when site adopts a new technology, and are 0 otherwise.</td></tr>
<tr><td colspan="2">Explanatory variables:</td><td></td></tr>
<tr><td colspan="2">Product complexity</td><td>We measure product complexity by a simple (log) count of subsystems the product is made up of. We calculate three separate complexity variables: one describing the complexity of the core subsystems (PC_Complex_Core) and two the peripheral subsystems (PC_Complex_Peri) at time t-1.[15]</td></tr>
<tr><td rowspan="2">Innovation type</td><td>Generational consolidation/ expansion</td><td>Generational consolidation (expansion) dummy that gets value 1 if there is a decrease (increase) in the number of subsystems at time t-1 (SC_peri_consolidation, SC_peri_expansion).</td></tr>
<tr><td>Architectural innovation</td><td>We construct a variable which captures the change of linkage between subsystems and how subsystems interact by the variable that gets value 1 if the number of servers used at the site has changed at time t-1 (SC_Architecture).</td></tr>
<tr><td rowspan="2">Innovation characteristics</td><td>Competency enhancing (/destroying)</td><td>We can only measure whether adoption of technology is competence-enhancing, i.e. whether it builds upon existing competencies, skills, and know-how. We calculate the variable IT_intensity based on the IT intensity ratio at time t (i.e. IT staff to overall staff ratio).</td></tr>
<tr><td>Radicalness</td><td>Not included.</td></tr>
</table>

The estimation procedure is discussed in Section 5.3.

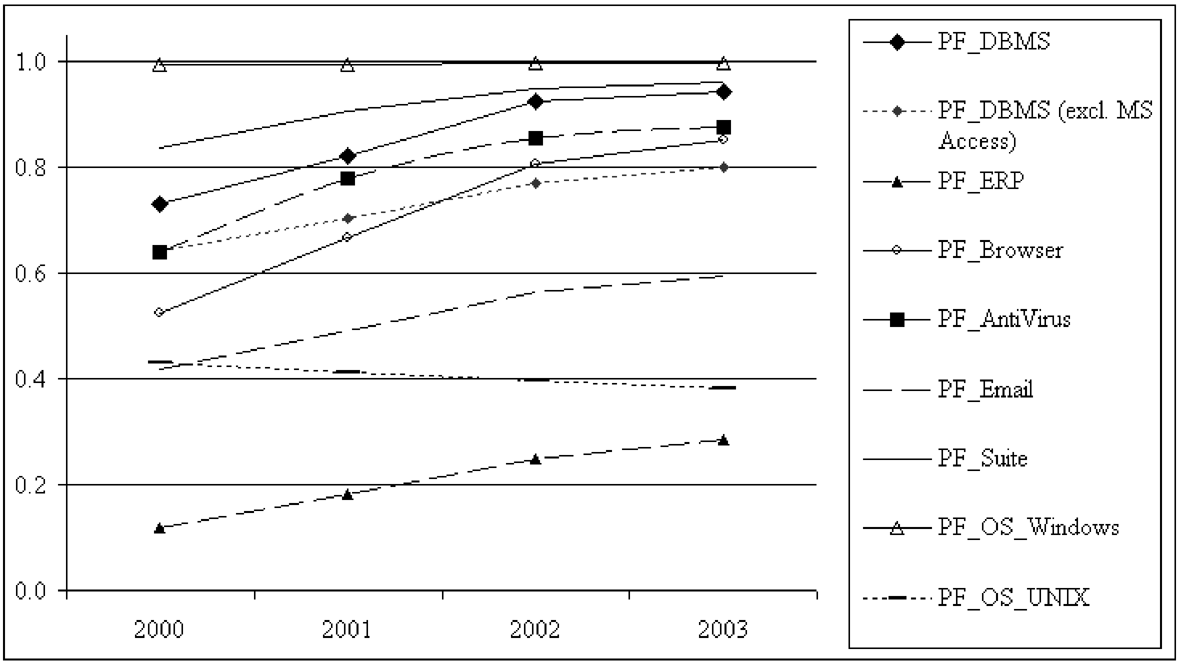
[15] Given the data we have, we can not match application software to OS and hence we do not know how the product complexity changes when the sites use multiple OS, i.e. to accommodate that application software, for instance an Anti-Virus product, might need to be multiple-configured in order for it to work across multiple OS

5 METHOD, RESULTS & INTERPRETATION

5.1 A brief look at the data

In Figure 5-1 we present the usage shares of the various product families (more detailed descriptive statistics are given in the Appendix). In general, it can be observed that software usage is increasing across the board (except for UNIX), and that general expectations about specific software packages are confirmed (for example, browsers diffuse rapidly, Windows and office suites are practically ubiquitous, and MS Access, a low-end database, drives most of the diffusion in database management systems). What is surprising is the fact that email usage lags far behind internet usage, which may be due to reporting peculiarities.

Figure 5-1
Usage shares of Product Families (Mean)



Our dataset covers multiple industries, locations, and sizes. Of the 3,891 companies we study, 49 per cent are in manufacturing, 42 per cent are service firms and 9 per cent are other sites. Further, 87 per cent of the sites are located in England, 7 per cent in Scotland, 3 per cent in Wales and the rest are firms in Northern Ireland or Other areas (e.g. Channel Islands). The majority of the sites in our sample employ between 50 to 499 employees – 35 per cent are large-sized sites (200-499) and 34 per cent are medium sized firms (50-199). 25 per cent of the sites are very large (500+ employees) and 6 per cent are small (1-49 employees). To illustrate the degree of interdependency between the technologies in our study, Appendix C gives a summary of the pairwise usage across all product families.

5.2 Adoption Regressions and results

We study the adoption of software product families across the three lenses: OS, DA and EA.[16] Our empirical approach does not differentiate if a site changes its product configuration or is a completely new adopter. The dependent variable in our models is a dummy variable and the regressors are both continuous (e.g. IT_Intensity) and discrete (e.g. industry dummies). To estimate the probability of adopting a certain technology we use a random effects probit model. As discussed previously, we calculate the innovation characteristics and identify sites which adopt a new (generation of) technology. Our econometric model explaining the firms' software adoption decisions by the characteristics of their software systems prior to adoption (i.e. previous year) also helps to overcome endogeneity problems that might arise from the use of the software system specific variables at the time of new software adoption.

Table 5-1
Adoption Regressions: Operating Systems

	PF_OS_Windows		**PF_OS_UNIX**	
Variable Name	Coeff.	t-value	Coeff.	t-value
PC_Complex_Core	-0.35	-11.06**	0.28	5.16**
PC_Complex_Peri	-0.07	-1.95*	0.11	1.93*
SC_Peri_Consolidation	-0.18	-5.18**	-0.17	-2.69**
SC_Peri_Expansion	-0.14	-2.20*	0.20	1.84
SC_Architecture	-0.07	-1.81	0.11	1.70
IT_Intensity	-0.02	-1.16	0.11	4.32**
LOG_Emp	0.07	4.47**	0.14	5.53**
Year_2003	-0.15	-4.96**	-1.04	-16.00**
IND_G_Service	-0.10	-1.80	-0.31	-3.70**
IND_G_Manuf	-0.14	-2.63**	-0.25	-3.12**
LOG_Emp	-0.11	-0.90	-1.86	-9.55**
Constant	-0.35	-11.06**	0.28	5.16**
Log-likelihood	-4892.79		-1435.94	
Observations	7,699		7,699	

Notes: ** indicates 1% significance, * indicates 5% significance

[16] Regression results for individual OS products (e.g. Windows 2000, Windows XP, etc. – see full list of Product in the Appendix) are available from the authors upon request.

Table 5-2
Adoption Regressions: Desktop Software

	PF_Anti-Virus		PF_Browser		PF_Email		PF_Suites	
Variable Name	Coeff.	t-value	Coeff.	t-value	Coeff.	t-value	Coeff.	t-value
PC_Complex_Core	0.10	2.78**	0.08	1.61	0.07	1.52	-0.05	-1.39
PC_Complex_Peri	-0.45	-11.78**	-1.31	-27.85**	-0.89	-20.24**	-0.51	-13.23**
SC_Peri_Consolidation	-0.12	-2.92**	-0.03	-0.44	-0.13	-2.44**	-0.18	-4.56**
SC_Peri_Expansion	-0.09	-1.23	-0.10	-1.02	-0.25	-2.42**	-0.14	-1.89
SC_Architecture	0.03	0.79	-0.05	-0.88	-0.13	-2.18**	-0.12	-2.80**
IT_Intensity	0.02	1.00	0.06	2.64**	0.01	0.60	0.03	1.53
LOG_Emp	0.02	0.96	0.06	2.81**	0.03	1.18	0.06	3.30**
Year_2003	-0.98	-26.62**	-0.29	-5.94**	-0.26	-5.71**	-0.05	-1.51
IND_G_Service	0.08	1.22	-0.05	-0.62	0.03	0.40	0.07	1.06
IND_G_Manuf	0.01	0.14	-0.19	-2.26**	0.02	0.25	0.02	0.28
LOG_Emp	0.44	3.27**	1.23	7.28**	0.42	2.58**	0.04	0.30
Constant	0.10	2.78**	0.08	1.61	0.07	1.52	-0.05	-1.39
Log-likelihood	-3541.47		-1843.47		-2011.35		-3553.15	
Observations	7,699		7,699		7,699		7,699	

Notes: ** indicates 1% significance, * indicates 5% significance

Table 5-3
Adoption Regressions: Enterprise Applications

	PF_DBMS		PF_ERP	
Variable Name	Coeff.	t-value	Coeff.	t-value
PC_Complex_Core	-0.05	-1.51	0.05	1.11
PC_Complex_Peri	-0.58	-15.32**	-0.09	-1.72
SC_Peri_Consolidation	-0.04	-1.03	-0.07	-1.29
SC_Peri_Expansion	-0.02	-0.27	-0.07	-0.68
SC_Architecture	-0.10	-2.35**	-0.01	-0.25
IT_Intensity	0.03	1.81	-0.02	-0.84
LOG_Emp	0.05	2.73**	0.04	1.93*
Year_2003	-0.26	-7.45	-0.17	-3.64**
IND_G_Service	0.03	0.48	0.28	3.10**
IND_G_Manuf	-0.05	-0.88	-0.06	-0.67
LOG_Emp	0.42	3.17**	-1.71	-9.08**
Constant	-0.05	-1.51	0.05	1.11
Log-likelihood	-3706.46		-1799.01	
Observations	7,699		7,699	

Notes: ** indicates 1% significance, * indicates 5% significance

5.3 Discussion of our Results

Our regressions shed light on a number of the issues we discussed earlier. We summarize our key results in three groups: a) We report the effect of site and industry characteristics on switching behavior and pay particular attention to the role of our measures for IT competency. b) The effects of product complexity on firm propensity to switch are discussed. c) Finally, we discuss the consequences of architectural changes on subsequent switching decisions.

Firm competency and task variety

We generally find that larger firm size facilitates switching to new software. This result is robust and significant across most specifications and applies to both core and peripheral subsystems (with the exception of antivirus and e-mail programs). This confirms the results by Kretschmer (2004) suggesting that larger firms have higher task variety. The IT intensity, as measured by the share of IT staff of all employees on site, is positively and statistically significantly related to the adoption of new UNIX operating systems and web browsers. As the interlinkages between software products are high, the costs for switching to a single product are typically much higher than the simple purchasing and adjustment costs for the isolated application, so that an IT-savvy firm will have higher absorptive capacity with respect to adopting certain new software.

System complexity

We find a negative relationship between the complexity of the core subsystem (**SC_Complex_Core**) in case of Windows operating system: companies with more complex core IT system are less likely to adopt or change to new Windows operating system. Instead, more complex core system seems to facilitate UNIX OS adoption, as well as the adoption of new antivirus programs. We find similar results with the complexity of the peripheral subsystems (**SC_Complex_Peri**): more complex existing peripheral systems hinder the adoption of Windows OS, while it affects positively the adoption of UNIX OS. The prior peripheral complexity also clearly has negative influence of further adoptions of both desktop software and enterprise applications. Generally, it seems then than when the complexity of peripheral subsystem is greater is less likely that the core or peripheral components are changed, with the exception of UNIX OS. These findings probably reflect the greater flexibility and power of the (programmable) Unix operating system compared to Windows which makes it more likely choice for complex IT infrastructures and systems.

We also find that the prior generational expansion of the peripheral sub-systems is generally negatively and statistically significantly related both to the adoption of new operating systems and other software. This highlights the importance of recognizing the links between different software applications and the evolution of user-customized system. Relative changes towards more complex user-customized IT infrastructure within a firm hinder further changes in both IT core and peripheral subsystems.

Architectural change

If a system architecture changes, e.g. the number of servers of IT infrastructure changes (**SC_Architecture**), the firm may try and avoid further software changes in the near future. We can therefore only speculate that there is a certain time lag between architectural (hardware) change and software adoption. That is, a lumpy investment is preferred to multiple smaller ones demanding continuous adjustments. We find some support for these ideas in our data: architectural change is negatively related to further changes in case of some software applications (e-mail, suites, DBMS).

6 CONCLUDING REMARKS

We proposed a framework to study the adoption of user-customised systems more thoroughly. We believe that our adoption framework provides some useful insights to study the adoption of multiple related technologies. This paper highlights the crucial importance of defining what constitutes a product if users get to assemble their technologies from different components for any meaningful study about diffusion of modular technologies. The empirical innovation and diffusion literature has a strong focus on studying single, well-defined products from the manufacturing industry. However, this may not be appropriate for more complex technologies like software. Packaged software is increasingly a core business enabler for any firm, and there is very limited work on the diffusion of software products. In this paper, we highlighted that software products are much harder to describe than other component systems (particularly classic examples from the manufacturing space, such as cars, etc.), mainly because of their high degree of modularity and complexity.

Another important observation is that it is critical to understand if the demand or supply side defines the product.[17] The current literature mainly assumes (certainly in the mass product space) that the product design (or ‘assembly’ of subsystems) is completed by the supply side (which again holds for cars, for instance). However, as we showed in the case of software products, the adopting firm designs the product, which requires a very different approach to study adoption. We believe that the framework we introduced can be successfully used to refine the demand side of software adoption.

We tackled our research question by looking at potential adopters’ product complexity, generational expansion and consolidation, architectural linkage and competency scale. We find a positive relationship between firm size and the adoption decision. We also find that the complexity of core and peripheral systems generally hinders the further adoptions of new software, with the exception of UNIX operating system of which attractiveness seems to increase with the complexity of a firm’s IT system. The evolution of IT systems is thus also strongly characterized by path dependencies: the prior core and peripheral system choices affect the future adoption decisions regarding the IT system. We also study the effect of architectural change and find a negative relationship between architectural linkage and certain software product adoption. We point out a number of limitations of our study:

First, our empirical analysis would benefit from a longer panel. However, as the software industry is characterised by very short product lifecycles and as the early 2000s were a particularly interesting period for the software market, we feel that working with only four years of data is acceptable. Further, as we construct the **SC_*** variables by comparing t-2 with t-1 data, our results may be more robust if we could calculate those variables based on longer trends.

Another limitation is that we try to capture the technological expertise of a site by the variable **IT_Intensity**. This variable is constructed by looking at internal headcount data. This implies, however, that we do not measure the degree of outsourcing of IT operations by firms. With the current dataset, such firms are classified as with low IT expertise, which is not necessarily correct. Hence, the accuracy of our results would benefit from details on externally sourced IT skills.

[17] We mean ‘define’ literally in this context and do not refer to the marketing idea of ‘supply push’ and ‘demand pull’.

Another potential source of error is that our current dataset does not contain any financial information. Given that the benefit of adoption is such a key driver for diffusion, follow-up papers would profit greatly if any financial data (e.g. firm profitability, adoption costs, etc.) could be mapped to the dataset Finally, the computer software market evolves hand-in-hand with the hardware market. We can assume that there are strong complementary effects and it would be valuable to enhance our work to include more characteristics describing those dynamics.

The challenge for future work is twofold: First, to improve the adoption framework and address the limitations outlined above. Second, however, it is also important to better understand the managerial consequences that can be derived from this method. For example, it would be interesting to better understand the following questions: a) 'How can core subsystems be replaced? Are they always replaced by new core subsystems (radical innovation) or can they also be 'graded down' to become a peripheral subsystem?'; b) 'How can we better describe what core and what peripheral subsystem are?'; and, c) 'Can peripheral subsystems 'be positioned' to be 'more core' than others?'

We stress that our empirical results are preliminary. Nevertheless, with the help of the adoption framework we have identified a set of intriguing, new characteristics (to our knowledge) to study the adoption decisions of user-customized systems.

References

Bokhari, Z. (2006) *Industry Surveys – Computers: Software (April 27, 2006)*. Standard & Poor's, Vol. 174, No. 17, Sec. 1, McGraw Hill, New York.

Bresnahan, T. and S. Greenstein (1999) *Technological Competition and the Structure of the Computer Industry*. The Journal of Industrial Economics, Vol. 47, No. 1, pp. 1-40.

Brynjolfsson, E. and C. F. Kemerer (1996) *Network Externalities in Microcomputer Software: An Econometric Analysis of Spreadsheet Market*. Management Science, Vol. 42, No. 12, pp. 1627-1647.

Cabral, L. (1990) *On the adoption of innovations with network externalities*. Mathematical Social Sciences, Vol. 19, pp. 299–308.

Choi, J. (1997) *Herd behaviour, the penguin effect and the suppression of informational diffusion*. RAND Journal of Economics, Vol. 28, pp. 407–425.

Christensen, C. M. (1998) *The Innovator's Dilemma*. Harvard Business School Press, Boston, MA, United States.

Christensen, C. M. and M. E. Raynor (2003) *The Innovator's Solution: Creating and Sustaining Successful Growth*. Harvard Business School Press, Boston, MA, United States.

Claussen, J., T. Kretschmer, and T. Spengler (2009): Backward Compatibility as Entry Deterrence – Evidence from the US Handheld Video Game Industry. *Mimeo*, University of Munich.

Cusumano, A. C. and D. B. Yoffie (2000) *Competing on Internet Time: Lessons From Netscape and its Battle with Microsoft*. Touchstone, New York, NY, United States.

Davis, F. D. (1989) *Perceived usefulness, perceived ease of use, and user acceptance of information technology*. MIS Quarterly, Vol. 13, No. 3, pp. 319-340.

Davis, F. D. and R. P. Bagozzi, P. R. Warshaw (1989) *User acceptance of computer technology: A comparison of two theoretical models*. Management Science, Vol. 35, pp. 982-1003.

Farrell, J. and G. Saloner (1986) *Installed base and compatibility*. American Economic Review Vol. 76, pp. 940–955.

Gandal, N. (1995) *Competing Compatibility Standards and Network Externalities in the PC Software Market*. The Review of Economics and Statistics, Vol. 44, No. 4, pp. 599-608.

Gatignon, H. and M.L. Tushman, W. Smith, P. Anderson (2002) *A Structural Approach to Assessing Innovation: Construct Development of Innovation Locus, Type and Characteristics*. Management Science, Vol. 48, No. 9, pp. 1103-1122.

Geroski, P.A. (2000) *Models of technology diffusion*. Research Policy, Vol. 29, pp. 603-625.

Goodstadt, B. and S. Kessler (2000) *Industry Surveys – Computers: Software (March 2, 2000)*. Standard & Poor's, Vol. 168, No. 9, Sec. 1, McGraw Hill, New York.

Grajek, M., T. Kretschmer (2009a), "Usage and Diffusion of Cellular Telephony, 1998-2004," *International Journal of Industrial Organization* 27(2), 238-249.

Grajek, M., T. Kretschmer (2009b), "Estimating Critical Mass in the Global Cellular Telephony Market," *Mimeo, ESMT* and *University of Munich*.

Griliches, Z. (1957) *Hybrid Corn: An Exploration in the Economics of Technological Change*. Econometrica, 48, pp. 501-522.

Gupta, S., Jain, D., and M. Sawhney (1999). Modeling the Evolution of Markets with Indirect Network Externalities: An Application to Digital Television. *Marketing Science*, Vol. 18, pp. 396-416.

Henderson, R. M. and K. B. Clark (1990) *Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms*. Administration Science Quarterly, Vol. 35, No. 1, pp. 9-30.

Karshenas, M. and P. Stoneman (1993) *Rank, Stock, Order and Epidemic Effects in the Diffusion of New Process Technology*. Rand Journal of Economics, Vol. 24, No. 4, pp. 503-528.

Katz, M. and C. Shapiro (1986) *Technology Adoption in the Presence of Network Externalities*. The Journal of Political Economy, Vol. 94, No. 4, pp. 822-841.

Kretschmer, T. (2004) *Upgrading and niche usage of PC operating systems*. International Journal of Industrial Organization, Vol. 22, pp. 1155-1182.

Kretschmer, T. (2005) *Competing technologies in the database management systems market*. NET Institute, Working Paper #05-17.

Nair, H., Chintagunta, P. and J.-P. Dubé (2004). Empirical Analysis of Indirect Network Effects in the Market for Personal Digital Assistants. *Quantitative Marketing and Economics* 2, pp. 23-58.

Nelson, R., A. Peterhansl and B. Sampat (2004). Why and How Innovations Get Adopted: A Tale of Four Models. *Industrial and Corporate Change*, 13, pp. 679-699.

Nilakanta, S. and R. W. Scamell (1990) *The effect of Information Sources and Communication Channels on the Diffusion of Innovation in a Data Base Development Environment*. Management Science, Vol. 36, No. 1, pp. 24-40.

Markides, C. and P. Geroski (2005) *Fast Second: How Smart Companies Bypass Radical Innovation to Enter and Dominate New Markets*. Jossey-Bass, San Francisco, United States.

Markides, C. (2006) *Disruptive Innovation: In Need of Better Theory*. The Journal of Product Innovation Management, Vol. 23, pp. 19-25.

Matutes, C. and P. Regibeau (1988) *"Mix and Match": Product Compatibility without Network Externalities*. The RAND Journal of Economics, Vol. 19, No.2, pp. 221-234.

Matutes, C. and P. Regibeau (1992) *Compatibility and Bundling of Complementary Goods in a Duopoly*. The Journal of Industrial Economics, Vol. 40, No. 1, pp. 37-54.

Moore, G. A. (2005) *Dealing with Darwin: How Great Companies Innovate at Every Phase of Their Evolution*. Penguin Books Ltd, London, England.

Rogers, E. (1995) *Diffusion of Innovations*. 4th edn. The FreePress, New York, United States.

Rudy, J. (2000-2004) *Industry Surveys – Computers: Software*. Standard & Poor's. McGraw Hill, New York.

Dates of the publications: October 5, 2000, Vol. 168, No. 40, Sec. 2; November 8, 2001, Vol. 169, No. 45, Sec. 2; October 10, 2002, Vol. 170, No. 41, Sec. 2; October 9, 2003, Vol. 171, No. 41, Sec. 2; October 28, 2004, Vol. 172, No. 44, Sec. 1

Schilling, M. A. (2000) *Toward a General Modular Systems Theory and Its Application to Interfirm Product Modularity*. The Academy of Management Review, Vol. 25, No. 2., pp. 312-334.

Schumpeter, J.A. (1934) *The Theory of Economic Development*. Harvard Business Press, Cambridge, MA, United States.

Stoneman, P. (2002) *The economics of technological diffusion*. Blackwell Publishers Ltd., Oxford, United Kingdom.

Thomas, L. A. (1999) *Adoption order of new technologies in evolving markets*. Journal of Economic Behavior & Organisation, Vol. 38, pp. 453-482.

Tushman, M. and P. Anderson (1986) *Technological discontinuities and organizational environments*. Administrative Science Quarterly, Vol. 31, pp. 439-465.

Utterbeck, J. (1994) *Mastering the Dynamics of Innovation*. Harvard Business School Press, Cambridge MA, United States.

Zettelmeyer, F. and P. Stoneman (1993) *Testing Alternative Models of new Product Diffusion*. Economics of Innovation and New Technology, Vol. 2, pp. 283-308.

Appendix A. Variable Definitions

Definition	Variable(s) Name
Generic	
Unique site identifier (Integer number)	Siteid
Variable equals year when site data was collected; values: 2000-2003	Year
Site Product Usage	
Following Product Categories (PC) are defined: - Operating System - Desktop Application - Enterprise Application Dummy equals '1' if site uses Product Category; else '0'	PC_OS PC_DA PC_EA
Following Product Families (PF) are defined: - DBMS - ERP - Anti-Virus - Email - Browser - Suites - Operating Systems Dummy equals '1' if site uses Product Family; else '0'	PF_DBMS PF_DBMS (excl. Access) PF_ERP PF_AntiVirus PF_Browser PF_Email PF_Suite PF_OS_Windows PF_OS_Unix
Following Products (PD) are defined: - 7 DBMS - 5 ERP - 2 Browser - 5 Anti-Virus - 4 Email - 7 Office Suites - 9 Operating Systems (Count includes Applications and Application versions) Dummy equals '1' if site uses Product; else '0'	PD_DBMS_MS_Access PD_DBMS_MS_SQL PD_DBMS_MS_Other PD_DBMS_IBM_All PD_DBMS_Oracle_All PD_DBMS_Sybase_All PD_DBMS_All_Other PD_ERP_SAP_All PD_ERP_People_All PD_ERP_JDEdwards_All PD_ERP_Oracel_All PD_ERP_All_Other PD_Browser_MS PD_Browser_Netscape PD_Anti_McAfee PD_Anti_DrSolomon PD_Anti_Associates PD_Anti_Symantec PD_Anti_All_Other PD_Email_Lotus_CCMail PD_Email_MS_Outlook PD_Email_MS_Mail PD_Email_All_Other PD_Suite_Lotus_SmartSuite PD_Suite_MS_All_Other PD_Suite_MS_Office95 PD_Suite_MS_Office97 PD_Suite_MS_Office2000 PD_Suite_MS_OfficeXP PD_Suite_All_Other

Definition	Variable(s) Name
	PD_OS_MS_Win95 PD_OS_MS_Win98 PD_OS_MS_Win_Other PD_OS_MS_WinNT PD_OS_MS_Win2000 PD_OS_MS_WinXP PD_OS_HP_UNIX PD_OS_Sun_UNIX PD_OS_UNIX_All_Other
Site Industry	
Classification of site into 3 generic industry types: - Services - Manufacturing - Other Dummy equals 1 if firm belongs to industry type, 0 otherwise.	IND_G_Service IND_G_Manuf IND_G_Other
Site size	
(log) Total number of employees on sites	LOG_Emp
Site adoption framework (explanatory variables)	
Product complexity: Log (count of the core subsystems and the peripheral subsystems) Peripheral subsystem comprises desktop and enterprise applications	PC_Complex_Core PC_Complex_Peri
Generational consolidation (expansion) dummy that gets value 1 if there is a decrease (increase) in the number of subsystems at time t-1, and 0 otherwise.	SC_Peri_Consolidation SC_Peri_Expansion
Dummy variable that gets value 1 if the number of servers used at the site has changed at time t-1, and 0 otherwise.	SC_Architecture
Measure of competence: IT intensity ratio measured based on number of IT employees; defined: log (IT employees on site/Total number of site employees)	IT intensity

Appendix B. **Summary statistics of the explanatory variables (for the sample years 2002 and 2003)**

Variable	Obs	Mean	Std. Dev.	Min	Max
PC_Complex_Core	7782	0.75	0.49	0.00	2.08
PC_Complex_Peri	7699	2.07	0.43	0.00	3.04
SC_Peri_Consolidation	7782	0.37	0.48	0.00	1.00
SC_Peri_Expansion	7782	0.07	0.25	0.00	1.00
SC_Architecture	7782	0.26	0.44	0.00	1.00
IT_Intensity	7782	-3.41	1.16	-7.50	0.00
LOG_Emp	7782	5.48	1.11	1.39	9.62
IND_G_Service	7782	0.50	0.50	0.00	1.00
IND_G_Manuf	7782	0.50	0.50	0.00	1.00
LOG_Emp	7782	0.41	0.49	0.00	1.00

Appendix C. Descriptive Statistics

Pairwise Usage

Product Family		DBMS	ERP	Browser	Anti-Virus	Email	Suite	Windows	Unix
DBMS	**2000**	--	.128	.513	.600	.391	.667	.727	.395
	2003	--	.291	.844	.865	.594	.924	.945	.385
ERP	**2000**	--	--	.140	.148	.131	.125	.118	.100
	2003	--	--	.293	.291	.246	.289	.285	.196
Browser	**2000**	--	--	--	.679	.377	.531	.525	.306
	2003	--	--	--	.797	.567	.839	.852	.370
Anti-Virus	**2000**	--	--	--	--	.420	.647	.638	.339
	2003	--	--	--	--	.570	.862	.879	.372
Email	**2000**	--	--	--	--	--	.419	.418	.282
	2003	--	--	--	--	--	.592	.596	.319
Suite	**2000**	--	--	--	--	--	--	.837	.389
	2003	--	--	--	--	--	--	.962	.376
Windows	**2000**	--	--	--	--	--	--	--	.428
	2003	--	--	--	--	--	--	--	.380
Unix	**2000**	--	--	--	--	--	--	--	--
	2003	--	--	--	--	--	--	--	--