The Disruptive Nature of Information Technology Innovations: The Case of Internet Computing in Systems Development Organizations

Author(s): Kalle Lyytinen and Gregory M. Rose

Source: MIS Quarterly, Vol. 27, No. 4 (Dec., 2003), pp. 557-596

Published by: Management Information Systems Research Center, University of Minnesota

Stable URL: http://www.jstor.org/stable/30036549

Accessed: 16/11/2013 09:49
Abstract

Information technology (IT) innovation can be defined as the creation and new organizational application of digital computer and communication technologies. The paper suggests that IT innovation theory needs to be expanded to analyze IT innovations in kind that exhibit atypical discontinuities in IT innovation behaviors by studying two questions. First, can a model of disruptive IT innovations be created to understand qualitative changes in IT development processes and their outcomes so that they can be related to architectural discontinuities in computing capability? Second, to what extent can the observed turmoil among systems development organizations that has been spawned by Internet computing be understood as a disruptive IT innovation?

To address the first question, a model of disruptive IT innovation is developed. The model defines a disruptive IT innovation as an architectural innovation originating in the information technology base that has subsequent pervasive and radical impacts on development processes and their outcomes. These base innovations establish necessary but not sufficient conditions for subsequent innovation behaviors. To address the second question, the impact of Internet com-
putting on eight leading-edge systems development organizations in the United States and Finland is investigated. The study shows that the adoption of Internet computing in these firms has radically impacted their IT innovation both in development processes and services.

Keywords: Internet computing, innovation theory, disruptive IT innovation, IT innovation cores, system development, software management, IT applications

Introduction

In the mid-1990s, computing entered a state of flux that was triggered by the Internet. The flux was accompanied by the creation of tools, techniques, and practices called Internet computing. Internet computing as used here denotes a broad and evolving set of models of distributed computing and related solutions that rely on open, heterogeneous, ubiquitous network services, and associated protocols (March et al. 2000). Many scholars and practitioners have argued that Internet computing represents a new way of conceiving and developing information systems and constitutes a break from earlier approaches to organizational computing organized around mainframes, personal computers, or client-server models (Alter et al. 2001; Baskerville et al. 2001; Lyytinen et al. 1998). Recent research provides ample anecdotal and speculative evidence that Internet computing has spawned a wave of innovations in system development and services

3Fundamental technological principles underlying Internet computing were invented in the 1960s and their first successful implementations developed during the 1970s (Berners-Lee and Fischetti 1999). The successful adoption of technologies started in the 1980s and they became the dominating computing paradigm during the mid-1990s after the invention of the http protocol and hypertext concepts (HTML) along with changes in the regulation of Internet traffic that led to explosive growth in the user base. Here we refer to Internet computing as a holistic concept that draws upon all protocols that enable computing in an open, distributed, and heterogeneous environment running on top of Internet-based transmission protocols.

These studies, although illuminating, do not formulate a theoretical model of how changes in information systems (IS) development and services depend on antecedent changes in technological capability, and what types of changes are necessary to establish a disruptive information technology (IT) innovation. In short, they identify neither necessary nor sufficient conditions for radical and widespread change in IS development and its outcomes. They also fail to provide empirical validation of the extent to which organizations deploying Internet technologies have experienced significant and lasting changes.

One reason why studies of Internet computing have not relied on prior IS innovation research is because questions about kinds of changes in system development and its outcomes—like the kinds identified in Bijker's (1987) discussion of technological frames—have not entered into the mainstream of IS innovation research (e.g., Fichman 1992; Prescott and Conger 1995; Swanson 1994). In contrast, past IS innovation research has primarily described and explained changes in the volume of technological and organizational change associated with IT innovation. What little work has been done in understanding differences in kind has been narrow in scope and limited to descriptions of changes as either fashions (Newell et al. 2000) or imitations (Loh and Venkatraman 1990).

To address the observed gaps in Internet computing studies and IT innovation research, this paper engages in a cycle of theory generation and validation that seeks to advance our understanding of the dynamics of IT innovation and specific IT innovation types. We use the generated theory to explore the extent to which Internet computing as a new paradigm has led to disruptive changes (introduced here as being both
radical and pervasive) in the services that are built, and in the ways those services are built. In brief, the research questions posed are:

1. How can we model disruptive IT innovations to help us identify and analyze qualitative changes in IS development and its outcomes due to architectural discontinuities in computing capability?

2. To what extent can the observed turmoil among systems development organizations spawned by Internet computing be understood as disruptive IT innovations?

These questions were addressed in two ways. First, we engaged in theory generation and developed a model of disruptive IT innovation by drawing on theories of industrial innovation (Christensen 1992a, 1992b; Teece 1986), Swanson’s (1994) theory of IS innovation, and Zaltman et al.’s (1977) concept of radical innovations. Second, we use this model to formulate two conjectures that instantiate the disruptive IT innovation model in the context of Internet computing and help clarify the disruptive nature of Internet computing. These conjectures were derived by a careful analysis of the extant literature on Internet computing in light of the disruptive IT innovation model. We conducted a multisite case study by drawing upon the disruptive IT innovation model to validate the extent to which development processes and services had changed radically and pervasively in eight system development organizations that adopted Internet computing. The study lends support to widely published claims that Internet computing can result in radical and pervasive changes in development processes, along with their outcomes. It also shows the value of the disruptive IT innovation model in analyzing complex IT innovations like Internet computing as innovations in type.

The paper concludes with a discussion of the implications for IS innovation research and practice, and an epilogue that discusses these implications in light of the end of the dot-com boom.

A Model of Disruptive IT Innovation

Innovation Defined

A widely accepted definition of organizational innovation is that it involves adoption of an idea, material artifact, or behavior that is new to the organization adopting it (Daft 1978; Rogers and Shoemaker 1971). Not all ideas, material artifacts, or behaviors adopted are innovations; an innovation must be accompanied with newness or novelty as a key distinguishing feature. Because defining newness is often difficult, it is argued that it is the perception of newness that counts, rather than whether the idea or artifact is new to the world. Following Zaltman et al. (1977), innovations are always defined in terms of a specific individual, organization, or community.

The innovation literature is voluminous and diverse, but two important streams can be distinguished: (1) theories of industrial innovation and (2) the diffusion of innovation literature. The first stream deals with types of innovations: artifacts and ideas that are new to a community or industry. Industrial innovation research has examined structural characteristics of an industry, product (architecture), market, or organization, and has asked why and how ideas, behaviors, or artifacts that are novel for the industry or community emerge, and what their impacts are (Abernathy and Clark 1985; Christensen 1992a, 1992b). The diffusion of innovation stream has focused on innovations in scope: the adoption of artifacts and ideas that are new to the would-be adopter once they have been discovered. This research stream has focused on the innovation-demand side, and primarily applied diffusion of innovation theory to discern patterns of diffusion (Rogers 1990). Researchers’ primary interests have been in discerning factors and processes

4For simplicity, “services” as used herein represent both systems (that cover hardware, software, people, and other artifacts and their relationships that result from systems development), and services (which cover exploitation of those systems in time and space to carry out some organizational tasks or processes).
that can explain adoption outcomes and the infusion of innovations over user populations (Fichman 1992; Kwon and Zmud 1987; Lyttinen and Damsgaard 2001; Prescott and Conger 1995; Rogers 1990).

**Disruptive Innovation**

Theories of industrial innovation seek to understand what drives innovation in the context of specific industries and products, technology, and market environments (Christensen 1992a, 1992b; Chistensen and Bower 1996; Dosi 1982; Foster 1986; Teece 1986; Utterback 1996). Often these theories relate structural features in products, services, and industries or interactive processes (Slappendale 1996) to specific innovation outcomes. Normally, these models also seek to explain necessary transformations in the industrial organization and/or markets as conditions for, or outcomes of, such innovations.

During the last decade, several seminal studies advanced disruptive architectural innovations as the main engines of industrial transformation (Christensen and Bower 1996; Henderson and Clark 1990; Teece 1986; Utterback 1996). Disruptive innovations are often the outcome of unleashing new product architectures that deviate radically from existing product lines by incorporating novel and unprecedented architectural principles like changing telecommunication service from circuit switching to packet switching, or transforming imaging from an analog to a digital process (Christensen and Bower 1996; Henderson and Clark 1990; Teece 1986; Utterback 1996). Architectural innovations stand out as creative acts of adapting and applying latent technologies or potential to previously unarticulated user needs (Abernathy and Clark 1985). They radically deviate from an established trajectory of performance improvement, or redefine what performance means in a given industry (Chistensen and Bower 1996). They are radical (Zaltman et al. 1977) in that they significantly depart from existing alternatives and are shaped by novel, cognitive frames that need to be deployed to make sense of the innovation (Bijker 1987). Consequently, disruptive innovations are truly transformative (Abernathy and Clark 1985). To become widely adopted, disruptive architectural innovations demand provisioning of complementary assets in the form of additional innovations that make the original innovation useful over its diffusion trajectory (Abernathy and Clark 1985; Teece 1986). By doing so, disruptive innovations destroy existing competencies (Schumpeter 1934) and break down existing rules of competition.

**Information Technology Innovation**

An information technology innovation can be defined as an innovation in digital and communications technologies and their applications (Swanson 1994). IT innovations result from exponential improvements in computing speed (e.g., Moore’s law) and data storage functions that have over time led to radically enhanced functionality in processing, storage, transfer, and display of information. Hence, most IT innovations are inherently linked to a continuously improving computing capability-cost ratio. In their simplest form, IT innovations involve only a technological component—changes in hardware and software that are new to an industry or adopters—but they are often augmented with complementary organizational innovations including new forms of cognition, meaning, work process, business process, or organizational structure. Specific IT innovations involve these elements in different proportions, and consequently affect the content, scope, and organization of IT innovation processes within an organization or the industry (Swanson 1994). As a result, IT innovations normally penetrate organizations through integrating a complex network of interrelated innova-

---

5 Similar research has been carried out in organization theory. Hage (1980), for example, distinguished between radical organizational innovations and incremental innovations in organizational change.

6 In the remainder of this paper, we refer to these functionalities as computing capability.
tions covering discovery of new computing capability, establishment of new development capabilities, and formulation of new services. In addition, other forms of innovation normally help deliver and manage those innovations including virtual teams, total quality principles, and the like.

**IT Innovation Types**

We distinguish between three types of IT innovations and their interactions in a model hereafter referred to as the *three-set model of IT innovation* (see Figure 1). The first innovation set (what Swanson calls Type I) involves changes in *system development processes*, such as new development tools or programming teams. The second innovation set consists of outcomes of development processes (i.e., *services*). This set involves uses of IT to support the administrative core of the organization (what Swanson calls Type II) such as accounting. It also deals with those innovations in which the uses of IT affect either business functions or core business processes of the organization (what Swanson calls Type III). For many of these IT innovations, changes in computing capability are often a necessary (but not sufficient) precursor.7 Hence, the third set, noted here as IT base innovations, change available computing capability. They establish a necessary antecedent

---

7Swanson (1994) recognizes the limited applicability of his model to all types of IT innovations when he discusses how IS administrative process and technological process innovations are preceded by innovations in antecedent technological systems. For example, the innovation of creating a Database Administrator function is preceded by the invention of database management systems. This antecedent innovation layer is neither codified nor explained as his study focuses on internal IS innovation. Interestingly, Grover et al. (1997) later suggested that his model has to be extended to include innovations in the basic IT (e.g., TCP/IP based networks and IT tools).
Table 1. Subcategories of IT Innovation

<table>
<thead>
<tr>
<th>IT Innovation Sets</th>
<th>Description</th>
<th>Examples</th>
<th>Swanson’s IT Innovation Subcategories</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT Base (Base)</td>
<td>Base Technology Innovation (Base1)</td>
<td>DBMS, Client/Server Computing; OODB;</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>Base Development Capability Innovation (Base2)</td>
<td>Software Patterns; Software Component Brokering, Quality Assurance</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>Base Service Capability Innovation (Base3)</td>
<td>Point-and-Click GUI, Multimedia, QoS</td>
<td>n/a</td>
</tr>
<tr>
<td>System Development (SD)</td>
<td>Administrative Process Innovation (SD1)</td>
<td>Maintenance departmentalization, CIO, open source development</td>
<td>Type I-a</td>
</tr>
<tr>
<td></td>
<td>Technological Process Innovation (SD2)</td>
<td>Systems programming, data administration, prototyping,</td>
<td>Type I-b</td>
</tr>
<tr>
<td>Services (S)</td>
<td>Administrative Process Innovation (S1)</td>
<td>Accounting systems, EIS</td>
<td>Type II</td>
</tr>
<tr>
<td></td>
<td>Technological Process Innovation (S2)</td>
<td>MRP, computer integrated manufacturing</td>
<td>Type III-a</td>
</tr>
<tr>
<td></td>
<td>Technological Service Innovation (S3)</td>
<td>Remote customer order entry and follow-on customer service systems</td>
<td>Type III-b</td>
</tr>
<tr>
<td></td>
<td>Technological Integration Innovation (S4)</td>
<td>Interorganizational information systems, EDI</td>
<td>Type III-c</td>
</tr>
</tbody>
</table>

and an element of many forms of other IT innovations (Friedman and Comnford 1989; Galliers and Somogyi 1987; Sauer 1999; Tsichritzis 1997). IT base innovations include, among others, new software and hardware architectures and services, and new telecommunication capability. The three sets of innovations are mutually dependent in that an innovation in one type may spawn innovations in others. This codependency is created because IT innovation processes are simultaneously driven by both push and pull forces (Zmud 1982, 1984) resulting in what Swanson calls order effects (shown in Figure 1 by double headed arrows).

Subcategories of IT Innovation Types

The three innovation sets in Figure 1 can be divided into subgroups, depending on the nature and content of the innovation in each innovation set (Swanson 1994). These subgroups are illustrated in more detail in Table 1 and are used to identify key areas of IT innovation in our analysis of Internet computing.

We divide base IT innovations into three subcategories by using the impact of change as a basis for classification. The three subcategories are (1) changes in the base technology as defined by functionality, speed, reliability, architectural principle, or other features (Base 1); (2) changes in IS development as defined by modeling and design principles or by coordination of related processes (Base 2); and (3) changes in services as defined by changes in general service features (Base 3). Subcategories in system development...
innovations affect either technical (System Development 1) or administrative development activities (System Development 2). The subcategories in service innovations are created by organizational boundaries and task types within the adopting firm (Swanson 1994). Based on Swanson, we distinguish between four types of service innovations: (1) services that support the administrative core (Service 1); (2) services that support functional processes (Service 2); (3) services that expand and support customer interfacing processes (Service 3); and (4) services that support interorganizational processes and operations (Service 4).

A Model of Disruptive IT Innovation

Some IT innovations may become disruptive in that they call for radical shifts in IT focus, capability, and solutions that demand significant investments in complementary assets (Friedman and Comford 1989). Yet, the IS research community currently lacks adequate models to identify such innovations in kind and thereby help distinguish between disruptive and incremental IT innovations. To deal with this omission in the literature, we propose a model of disruptive IT innovation that is intended to examine the dynamics in kinds of IT innovations. Such a model would recognize those changes within computing capability that would constitute a significant transformation in the architectural principles governing services and development processes. The innovations in the IT base would thus be necessary (but insufficient) to create subsequent changes in development processes and services.

We define a disruptive IT innovation as a necessary but not sufficient architectural innovation originating in the IT base that radically and perversely impacts systems development processes and services. To avoid technological determinism we use the terms necessary and not sufficient in the definition to clarify the conditions under which specific changes in the technology base can become disruptive. In line with process theories (Markus and Robey 1988) of technological impact, architectural changes in the IT base have to be present in order for a disruptive IT innovation to happen. Hence, an architectural change per se does not create disruptive change in other IT innovations, but requires other (sufficient) conditions to be met as defined by the concept of complementary assets (Abernathy and Clark 1985; Teece 1986). In a disruptive IT innovation, the computing capability undergoes a qualitative change in kind through a set of novel IT base innovations and their application outcomes that radically alter the subsequent trajectory of IT innovation. A disruptive IT innovation knits together a set of interrelated technological and organizational advances involving qualitative IT base innovations and related IT innovations in development processes and services that further the exploitation of the base technology innovation by providing co-specialized assets (Teece 1986).

Significant qualitative changes in computing capability are outcomes of a unique constellation of non-linear growth in underlying computing power. This new qualitative capability eventually overwhelms the current computing paradigm and its associated design space created by available technological frames (Bijker 1987). For example, the http protocol and the concept of a browser within Internet computing (Berners-Lee and Fischetti 1999) drew upon exponentially growing capability in computing speed, data storage, and transfer. At the same time, a qualitative change occurred in the way we think about computing capability. This capability was expanded successively through new models and principles of distributed computing that refined and improved the architecture, solution patterns, and services of Internet computing—thereby accumulating co-specialized assets associated with the new paradigm. Hence, unique moments of discovery in computing capability can introduce a possibility for waves of other IT innovations that push the

---

5 This is in contrast to variance theories, which seek to identify both necessary and sufficient conditions to establish a strong form of causality.
new computing capability into the mainstream and over time integrate co-specialized assets (innovations) into the dominant computing platform like design methods, performance standards, or alternative ways of organizing the development activity (Christensen and Bowers 1996). In short, as a result of a novel base-computing discovery, IT innovators face a revolution and undergo a paradigm shift (Kuhn 1996).

Our definition identifies two conditions that together define a disruptive IT innovation; specifically, changes in IT innovations that follow a base IT innovation must be both pervasive and radical. The dimension of pervasiveness follows from the concept of disruptive architectural innovations, which states that these innovations create qualitative changes—a kind of systemic shock throughout computing—after which the technology and the existing rules of the game are changed. Accordingly, a disruptive IT innovation is pervasive in that it simultaneously and necessarily spans new services, and new types of development processes, thereby covering all of the IT innovation subcategories listed in Table 1.

The ability to meet the requirement for pervasiveness for sets of IT innovations does not itself result in a disruptive IT innovation. For example, a minor upgrade in one base technology area—for example, a change in the color capability of standard Wintel software that would count as a service capability innovation—is subsequently felt throughout all IT innovation types. However, the overall impact of such an innovation would be insignificant, because this type of a change in computing capability would not substantially affect the configuration of other capabilities in the technology base, and therefore would not qualify as an architectural innovation. A disruptive IT innovation embraces a radical shift in how adopting organizations must view, operate, and utilize IT so that their subsequent use of computing capability will be different after adopting the IT base innovation. Such a change is radical when technology adopters’ behaviors depart significantly from existing alternatives (livari 1986; Zaltman et al. 1977) so that the services, as well as the development processes, deviate signifi-

[10]This implies that IS developers must unlearn and drop their cognitive schemata. They must experiment, engage in bricolage (Ciborra and Lanzara 1994), and negotiate what the technology signifies, and how it can be exploited.

Internet Computing as a Disruptive IT Innovation

Definition and Origins of Internet Computing

Internet computing has been seen by many to be a major revolutionary change in computing (Alter et al. 2001; Baskerville et al. 2001; Carstensen and Vogelsang 2001; Glass 2001; Isakowitz et al. 1998; Lytinen et al. 1998; Lytinen and Yoo 2002; Pressman 1998; Turoff and Hiltz 1998). It departs in multiple ways from earlier computing concepts both in what design elements can be manipulated, and in how an IT service is developed and assembled (Lytinen et al. 1998). Internet computing forms a creative synthesis and expansion of the long and relatively slow-paced evolution of packet-based data transmission principles, which have been associated with open data transmission protocols (TCP/IP) (Tuomi 2002). In the mid-1990s, these protocols were integrated with the functionality of client-server computing and hypertext capability through the invention of a browser and the associated principle of tagging texts and other data that guides their representation while they are transferred over the Internet (i.e., the invention of World Wide Web at the European Organization for Nuclear Research (CERN) by Berners-Lee in 1992). By pairing HTML language (in its variants) with the http protocol embedded in the browser (MOSAIC), an event-artifact pair was created that
crystallized a new revolutionary architectural concept (Berners-Lee and Fischetti 1999; Tuomi 2002). At the time, this was revolutionary because tagging with HTML and the use of a thin (browser) client operating a simple request-response protocol (http) masked all underlying design complexity of service interactions in an open environment. Internet computing, therefore, offered a new mechanism that unified interactions over open networks and made it possible to execute distributed transactions and to build radically new types of services. These unforeseen services could be further vertically integrated with legacy platforms that resulted in the additional integrative capabilities of Internet computing.

**Internet Computing as an Architectural Base Innovation**

It is not easy to pin down all of the features that characterize the new capabilities associated with Internet computing. To clarify the essential architectural features of Internet computing we adopt Lyytinen et al.’s (1998) discussion as a starting point. They enlist nine characteristics, called technology characteristics of Internet computing. These characteristics can be regarded in our classification as IT base innovations (Base1 per Table 1). Out of those nine, we recognize three characteristics—thin, universal clients, middleware, and the n-tier architecture—that constitute the key architectural principles of Internet computing. We regard these three as key architectural characteristics of Internet computing because they formed critical novel elements in the invention of the original http protocol, thin clients (browsers) and tagging with HTML (Berners-Lee and Fischetti 1999). These three characteristics together also define what makes Internet computing distinctively different from earlier computing paradigms in that they integrate multiple latent computing capabilities into a new computing architecture, which addresses an obvious but largely unexplored user need (Abernathy and Clark 1985; see also Berners-Lee and Fischetti 1999): how to represent, organize, and distribute information uniformly, and in an interoperable manner that supports transactions within an open network.\(^{11}\) Hence, the minimum criteria for engaging in Internet computing can be defined through the use of the following: (1) a three or n-tier architecture; (2) a middleware layer; and (3) a thin client like a browser, or a WAP-phone that operates over TCP/IP network using http or other higher level protocols.

Several latent computing capabilities have been enabled and mobilized by adopting these three architectural characteristics.\(^{12}\) Not all Internet computing services draw upon each of these capabilities, but they as a whole offer necessary flexibility in adapting Internet computing to multiple service needs and types and to development contexts. They also define increasing levels of complexity that specific Internet computing services may have.

The list of latent innovations integrated into Internet computing cover the following:

1. Separation of the representation layer from the business logic layer as adopted from database research of the late 1970s. This provides a basis for provisioning a uniform user interface layer (van Griethuysen 1982).

---

\(^{11}\) Naturally these features are dependent upon other technology features that are critical for Internet operations including data transmission protocols, file transfer protocols, addressing protocols, etc.

\(^{12}\) These capabilities are: (1) services have a uniform look and feel interface for each platform and offer the same functionality as dedicated application based interfaces; (2) services are based on scripting and metadata (HTML/XML code) that complement traditional compiled code, thus enabling new levels of interoperability; (3) services depend on effective and flexible open telecommunication services, which also include mobile services making any time/any place of service possible; (4) services utilize component-based software capability that makes it possible to use granular, configurable, and market driven software components across multiple platforms; (5) code and data are dynamic and can reside anywhere on the network and be posted dynamically through an appropriate interface; (6) services integrate dynamically both structured and unstructured data at the interface and database levels.
2. Generalized tagging as adopted from SGML standardization that drives the execution at user clients (see, for example, http://xml.coverpages.org/sgml.html).

3. Hypertext (see, for example, http://www.w3.org/MarkUp/) and multimedia (see, for example, http://www.w3.org/AudioVideo/) capability as adopted from stand-alone multimedia systems from the 1980s (Berners-Lee and Fischetti 1999). This constituted one of the key features of HTML tagging together with the use of a Universal Resource Locator (URL).

4. Object-orientation as adopted from object-oriented programming principles developed in the late 1960s as a basis to offer flexible and interoperable services (Krämer et al. 1998).

5. Reusability as adopted from programming methods developed in the early and mid-1970s to offer easy assembly of services (Kim and Stohr 1998).

6. The idea of interoperable services across standardized interfaces executing several high-level protocol stacks as developed earlier by the ISO/OSI model (Krämer et al. 1998).

Overall, Internet computing enables unforeseen flexibility in the design, implementation, distribution, and delivery of IT services that can satisfy previously unexplored user needs such as viewing IT services as an instance of media. To achieve this, Internet computing has integrated innovations in the three base innovation subcategories identified previously in Table 1. These “Internet Computing” innovations include those in base technologies (Lyytinen et al. 1998) (ICBase1), in development capability (ICBase2), and in service capability (ICBase3), as shown in Table 2. By offering these new capabilities, Internet computing has transformed the development and delivery of IT services and their performance characteristics (Christensen and Bower 1996).

### Internet Computing as a Disruptive IT Innovation

Having examined novel architectural characteristics of Internet computing, we can now ask whether resulting IT innovations in services and development processes meet the specific criteria of disruptive IT innovation. The following two research conjectures formulate the criteria under which Internet computing can be regarded as a disruptive IT innovation. These conjectures follow the logic (Markus and Robey 1988) of process theories as suggested by the definition of disruptive IT innovation. These conjectures seek to detect necessary but not sufficient conditions for specific adoption outcomes. We propose conjectures in this form to avoid the trap of technological determinism in explaining the impact of Internet computing. This is at odds with the concept of complementary assets (Abernathy and Clark 1985; Teece 1986).

**Research Conjecture A: Radical IT innovation in Internet computing related development processes:** Internet computing will be a necessary (but not sufficient) factor radically affecting system development processes in adopting systems development organizations.

**Research Conjecture B: Radical IT innovation in Internet computing related services:** Internet computing will be a necessary (but not sufficient) factor radically impacting services developed by adopting systems development organizations.

These conjectures convey the necessary (but not sufficient) relationships between the main constructs of the disruptive IT innovation model and their application to Internet computing. The

---

13 It can be speculated that some radical innovations in development processes and services are caused indirectly from radical IT base innovations via the other sets. However, this interaction would not be necessary in order to have a disruptive innovation. As a result, its existence is omitted from the formal conjecture.
### Table 2. Internet Computing as Architectural Base Innovation

<table>
<thead>
<tr>
<th>Architectural Features</th>
<th>Innovative Features of Internet Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Technology Innovation of Internet Computing (ICBase1)</td>
<td><strong>ICBase1a</strong>: Services use uniform clients (e.g. browser) with multimedia and hypertext capability and platform independence (Java, Java Scripts, Java Beans).</td>
</tr>
<tr>
<td></td>
<td><strong>ICBase1b</strong>: Services use middleware to &quot;glue&quot; transactions, external and unanticipated services, and legacy systems together. Service interfaces of the middleware allow single-user, workflow, and group-level services to be configured into the same user interface.</td>
</tr>
<tr>
<td></td>
<td><strong>ICBase1c</strong>: Services separate user interface, application logic and application data, which introduces new levels of flexibility in offering services and developing them.</td>
</tr>
<tr>
<td>Base Development Innovation of Internet Computing (ICBase2)</td>
<td><strong>ICBase2a</strong>: Mainstreaming and runtime access mechanisms help exploit software components through component markets.</td>
</tr>
<tr>
<td></td>
<td><strong>ICBase2b</strong>: Middleware frameworks and solutions emerge as technological mechanisms for coordinating and managing development complexity.</td>
</tr>
<tr>
<td></td>
<td><strong>ICBase2c</strong>: Services are designed from a media perspective.</td>
</tr>
<tr>
<td>Base Service Innovation of Internet Computing (ICBase3)</td>
<td><strong>ICBase3a</strong>: Services can be made available anytime and anywhere.</td>
</tr>
<tr>
<td></td>
<td><strong>ICBase3b</strong>: Services have multi- and hypermedia-features.</td>
</tr>
</tbody>
</table>

*The literature is based on an analysis of the following references: Alter et al. 2001; Baskerville et al. 2001; Carstensen and Vogelsang 2001; Glass 2001; Isakowitz et al. 1998; Kammersgaard 1988; Lyytinen et al. 1998; Lyytinen and Yoo 2002; Turoff and Hiltz 1998.*

*Notations for Internet computing innovations are to be read as follows: in ICBase1a: "IC" denotes Internet Computing, "Base" refers to its nature as Base Innovation, "1" refers to its type as a base technology innovation, and "a" refers to the specific item in that innovation set (first item).*

Investigation of these constructs and their relationships can draw upon objective indicators such as the emergence, or the existence, of specific technology artifacts including services and tools, and the birth of new processes, goals, or organizational principles. In addition, the evaluations may seek evidence of whether these changes have been radical in the sense that the developer's cognitive realm, behaviors, and expectations have significantly changed. The evaluation of radicalness thus relies on perceptual measures and often demands a post hoc assessment by the adopters regarding the extent to which their new alternatives depart significantly from the earlier technological trajectory.

Evidence for both conjectures can be sought in multiple ways. A first, indirect way is to explore professional texts and literature for evidence of how system development has changed since adopting Internet computing, and trace this change to the adoption of Internet computing. A second, more direct way is to investigate empirically (either through qualitative or quanti-
Innovations in Systems Development and Services Created by Internet Computing

The first step taken to validate Conjectures A and B was to analyze textbooks and the professional literature published between 1998 and 2002 to find out how characteristics of system development and services have been observed to change as a result of adopting Internet computing. Tables 3 and 4 summarize our main findings with references to the pertinent literature about services and their development and suggest a set of sub-conjectures that can be validated through empirical fieldwork. The deliberations of the impacts of Internet computing cover all IT system development (noted by SD) and service (noted by S) innovation subcategories as outlined in Table 1. This suggests the pervasiveness of the impact of Internet computing (noted by IC in Tables 3 and 4). Moreover, we detected a significant number of unforeseen and novel services in the literature that suggest that Internet computing has had a radical impact on new services. Likewise, a large set of new development features and processes including telecommunication design, artistic development, and increased diversity of development tasks indicate that radical changes in system development have taken place since the adoption of Internet computing. The simultaneous impact of Internet computing on system development processes and on services can be inferred from the fact that the demands for new services have consequently created the increased diversity and new tasks within development processes. Overall the analysis of the extant system development literature argues for both the radical and the pervasive impact of Internet computing on services and system development processes as reflected by the sum of items in Tables 3 and 4.

A Multisite Case Study of Internet Computing as a Disruptive IT Innovation

Research Methodology and Sampling

To further validate the two conjectures proposing the disruptive nature of Internet computing, we carried out a theory-driven, rigorous, and confirmatory multisite case study (Yin 1994) in eight system development organizations. Four

---

14 We carried out a relatively systematic search of all textbooks that discussed e-business or Internet development. In addition, we searched conference proceedings dealing with systems development and software engineering (e.g., Information Systems Development, European Conference on Information Systems) and analyzed published articles that discuss system development changes during Internet computing. Thanks go to two anonymous reviewers and the associate editor for pointing out additional sources.

15 The authors approached the companies with a request of having access to their system development documentation and technology strategies as well as a willingness to share their experiences of using Internet computing. We promised them a summary report of our findings, and a list of “best practices” in these companies. Under these conditions, all contacted companies were willing to share their experiences, and discuss and reveal their technology and software development strategies, as long as their anonymity was honored.
Table 3. Internet Computing as a Disruptive IT Innovation in System Development

<table>
<thead>
<tr>
<th>System Development Innovation Types</th>
<th>Pervasiveness of Innovation</th>
<th>Radical Nature of Innovation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrative Process Innovation of Internet Computing (ICSD1)</td>
<td>ICSD1a: User interface design involves new artistic and representational tasks; and this has increased the division of labor.</td>
<td>The organizational <em>modus operandi</em>, and the technological basis of system development will radically change as a result of system development innovations.</td>
</tr>
<tr>
<td></td>
<td>ICSD1b: System development has become mission critical for organizations and therefore an integrated element of organizational design, which necessitates the use of change management tactics in the ISD process.</td>
<td>Literature:* Alter et al. 2001; Baskerville et al. 2001; Braa et al. 2000; Carstensen and Vogelsang 2001; Lyytinen et al. 1998; March et al. 2001.</td>
</tr>
<tr>
<td></td>
<td>ICSD1c: Due to the unprecedented heterogeneity and diversity of development tasks, new managerial and organizational strategies are needed to manage and coordinate these tasks.</td>
<td></td>
</tr>
<tr>
<td>Technological Process Innovation of Internet Computing (ICSD2)</td>
<td>ICSD2a: Telecommunication services have become a critical and integrated part of the solution and will increase the division of labor.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ICSD2b: Increased technical and business complexity of services demands new process and coordination technologies.</td>
<td></td>
</tr>
</tbody>
</table>

*The references deal primarily with the radical nature of change in different innovation sets outlined in Table 1. The pervasiveness is observed by noting that the literature suggests that all innovation sets will be radically affected.
Table 4. Internet Computing as a Disruptive IT Innovation in Services

<table>
<thead>
<tr>
<th>Service Innovation Types</th>
<th>Pervasiveness of Innovation</th>
<th>Radical Nature of Innovation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrative Process Innovation of Internet Computing (ICS1)</td>
<td>ICS1a: Unforeseen services for administrative core covering non-strategic intranets, routine data and information delivery, document management and sharing using Internet computing capability have been developed.</td>
<td>Services become an integral part of the business Web and inseparable from the organization itself and its environment. The services and users extend beyond the enterprise boundaries, and are not easily identified.</td>
</tr>
<tr>
<td>Technological Process Innovation of Internet Computing (ICS2)</td>
<td>ICS2a: Unforeseen services for functional integration covering strategic intranets, research and development-related knowledge management applications and, business intelligence applications are using Internet computing has been developed.</td>
<td></td>
</tr>
<tr>
<td>Technological Service Innovation of Internet Computing (ICS3)</td>
<td>ICS3a: Unforeseen services for process integration related to customer related order entry and customer relationship management that utilize one-to-one marketing ideas (rule based, collaborative filtering) have been developed.</td>
<td></td>
</tr>
<tr>
<td>Technological Integration Innovation of Internet Computing (ICS4)</td>
<td>ICS4a: Unforeseen services for inter-organizational integration including collaborative extranets, electronic market places, electronic auctions, supply-chain management, and logistic management systems have been developed.</td>
<td></td>
</tr>
</tbody>
</table>

*The references dealt primarily with the radical nature of change in different innovation sets outlined in Table 1. The pervasiveness is observed by noting that the literature suggests that all innovation sets will be radically affected.*

U.S. and four Finnish firms were chosen to minimize cultural or regional bias. The firms selected could be regarded as leading edge exploiters of Internet computing in the early adoption cohort (Rogers 1990).\(^{16}\) They were all heavily involved in systems development that drew upon Internet computing, which formed a key strategic direction of their business.

A semi-structured interview questionnaire (Appendix A) was formulated on the basis of Tables 2, 3, and 4 to carefully validate both Conjectures A and B. Several interviews were conducted with en-

\(^{16}\) We knew from contacting some organizations that only a few were engaged in Internet computing at the time of the interview due to the learning barriers involved (Attewell 1992). Internet computing was too complex, too alien, and too fast-changing for traditional IS departments. This was mentioned many times in the interviews as a reason for these organizations to enter the market, and why they were able to work with so many clients.
sembles of highly technical and managerial-level employees within these organizations. The interviews were transcribed, coded, and validated across the research team and subsequently with the interviewees. Details of the methodology and the reliability and validity issues of the research results, firm characteristics, and interview participants are provided in Appendix B.

Each organization had been engaged in Internet computing for two to three years at the time of the interviews, and was experienced in assessing the critical features of the new architecture. To find the participants, only those firms meeting the following minimum criteria were pursued. First, firms had to build systems that were utilizing the base technology innovations of Internet computing (ICBase1a-c): (1) uniform clients (e.g., the use of the browser as the software client); (2) middleware (e.g., what is commonly referred to as Internet servers or Web servers to run their applications); and (3) an n-tier architecture (e.g., interface separate from application logic separate from data management function). Based on available public information such as brochures, phone discussions, e-mail, and personal meetings, several firms were identified. Further discussions eliminated firms that were building only trivial systems in the area, those that had only recently begun to build Internet systems, and those firms that had not declared that they had changed their primary focus to development for the Web. All participant firms had to indicate that they were developing systems compliant with one of the commonly recognized middleware architectures that embody most of the latent technology capabilities discussed above. Preliminary discussions also validated the awareness of these capabilities among participating firms. Any firm that was not aware of these characteristics and did not hold a sufficient working knowledge of how to exploit them was omitted from the study.

**Exploitation of Internet Computing among Studied Firms**

We naturally observed variation in the extent to which firms were exploiting all latent capabilities of Internet computing. Most notably, only one of the U.S. firms but all four of the Finnish firms were building applications that included client-side applications for mobile computing. Likewise, only two U.S. and two Finnish firms were targeting development of middleware services as a focus of their business strategy (and were thus more intensively exploiting this capability of Internet computing than the other four).

We also verified that each of the eight firms recognized, and in most cases exploited, innovations in Internet computing development capabilities as summarized through ICBase2a-c as well as its new general service capabilities ICBase3a-b. At a minimum, each firm had adopted an application framework (J2EE) architecture with Enterprise Java Beans (EJB) components that enabled the component market to mature (ICBase1a) (Narayanan and Liu 1999). Seven of the eight firms had adopted more than one of the common n-tier architectures (Sun’s J2EE, OMG’s CORBA, or Microsoft’s DNA.Net) for their Internet computing. All three architectures offer strong mechanisms to exploit components (ICBase2b), and therefore the assembly of components had become, or was becoming, a routine part of the development processes in these companies. Exploiting component markets in fact demanded the adoption of software integration mechanisms such as JavaBeans, Microsoft’s DCOM or .Net, and CORBA and associated middleware technologies (ICBase2c).17

Interviews with the firms also included reviews of their software portfolios. These reviews showed that organizations extensively exploited innovations in base service capability (ICBase3). For each firm, the majority of the new applications assumed ubiquity of service (ICBase3a) with 24-hour-a-day by 7-day-a-week availability, and a global reach via the Internet. Likewise, seven of the eight firms indicated that they were developing systems that required careful attention to media features (ICBase2c/ICBase3b) in a way that was a radical departure from traditional systems.

Internet Computing as a Disruptive Development Innovation

Data from the interviews support Conjecture A that organizations adopting Internet computing had to carry out significant development process innovations. Overall we found strong evidence for Conjecture A: system development trajectories had significantly departed from past practices as a result of adopting Internet computing. For each of the five features (ICSD1a-c, ICSD2a-b) presented in Table 3, we provide evidence below of how companies had experienced or carried out significant changes in development processes. These changes are summarized in detail in Appendix C.

Seven firms argued that the prevailing way to develop systems was being replaced by a very different idea of systems development as a novel administrative process innovation (ICSD1c). All but one (firm 4) firm of the older firms admitted that the new systems development environment challenged many of their widely held truths of system development. Those most affected were firms with rigid methodologies (firms 1, 3, 7, and 8). They found that they now either had to give up or radically alter their routines of how to plan, coordinate, and control development through methodologies. Many of the firms believed that having structured methodologies was desirable and saw the time before Internet computing as more conducive to methodologies. For example, a developer in firm 8 referred to this period as “you know, [the] good old days.” The following quote from the architect at firm 1 demonstrates the radical break in the effectiveness of the old practices:

I did out of shear panic. I spent two days [at a customer site] trying to get [deliverables from using the old methodology] that worked and failed miserably. It was embarrassing.

16Firm 4’s culture had been one of constantly changing development practices. In essence, their modus operandi was perpetual disruptive innovation. Since their processes had changed constantly before Internet computing, they saw no major differences, as they had to continue to adapt.

The firm abandoned this methodology, but did so reluctantly because they had invested in it. According to this architect,

We understand the importance of a methodology and repeating the process. As a matter of fact, at 2:00 today I’ve got the first of several sessions with some of our senior staff here on [identifying] what is that process and how does it work [and then] train some more people up on it. But it is a different process. We found that the processes we used in client/server are no longer applicable. In the client/server space, the project life cycle expectation is nearly six or nine months. In e-Business, it’s six to nine weeks. And that’s a huge back and forth change, and that’s a totally different process.

Similarly, developers in firm 3 recognized that methodologies that had worked successfully for their legacy systems could not be applied to Internet computing. As a result of failing to transfer existing methodologies to the Internet computing environment, firm 3 had begun to design new methodologies that would work with the new delivery speeds required in Internet computing (Greenbaum and Stuedahl 2000). They strongly believed that these methodologies would be useful based on their past success with methodologies to build legacy systems:

We’re going to be smart on our feet. I should say quick on our feet and be smart about our [yet to be developed] methodology, and that we will vary our methodology [from the legacy ones]. We have to [because our] environment change[d]. And that’s how successful companies deliver at Internet speed...[our currently developed Internet services] were developed using some disparate methodology resources and quite honestly we don’t have [methodologies] today, that’s one of the things we’re working on—a more well-defined set of methodology tools that will
enable us to develop and deliver at the speed required by the new economy. . . . So I suppose with all due respect to [the developers] and the other technology people in [the new company] today, they've just been kind of winging it and doing what they have to do using some basic methodology rules and controls to deliver a quality product and deliver a safe product that's reliable. The group knew a methodology...and wanted a methodology, but it was a case of we didn't have one really available to us, and I mean there was a large effort of putting [the old methodology] in place within [parent firm in the past; stressing they are important to them]. So like I said, we just kind of winged it.

Likewise, a representative from firm 8 observed that traditional methodologies were too heavy for their current development, which required a “very lightweight” process.

The remaining traditional development firm (firm 2) experienced the least amount of change in development processes. The lack of significantly altered processes could be attributed to the fact that they were traditionally only using abstract ideas and principles and did not historically have rigid methodologies:

I will not say [our traditional] methodology goes to a level as deep as an approved state-of-the-art software process would dictate. We are largely relying as well on existing known literature. We have a number of identifiable best practices books and literature on how to do stuff.

They did, however, recognize significant changes to their prototyping processes and tool use when developing services for Internet computing. So in these two areas where they had a rigid method in earlier development, disruptions were seen.

Beyond changes in development methodologies, most firms saw an unprecedented reliance on the purchase and assembly of software components as a critical part of their software development process (ICSD1c, ICBase2a). The overriding theme among the firms was that a combination of the mature components market (ICBase2a) along with drastically shortened timelines imposed by their customers created this change (or planned change) in how they built software. All firms found that the adoption of Internet computing correlated with expectations of a drastic shortening of software delivery times when compared with traditional system development. In many cases, timelines were shortened by a factor of four or more. In response, seven firms (all but firm 4) were currently buying, or were planning on buying, significantly more components from the external market. As an executive of firm 1 stated:

We always take the fast way out [now in Internet computing]. We have often been able to buy and plug in a component faster than we would have been able to just spec out the requirements alone, let alone build with quality assurance.

Similarly, the executive in firm 5 stated that if there is “a good component that some small company has, there’s no sense for us making [it] ourselves.”

By changing the scope, content, and complexity of firms’ development processes, Internet computing had also created the need for the firms to segment their resources into stronger specializations than was required in traditional system development. The nature of development was radically different as it relied upon these increased specializations, and consequently development complexity had increased in an unprecedented fashion. Firms had difficulties coordinating new skill sets and specialized tasks, and they were constantly crafting resources to manage and integrate the work in each specialized area (ICSD2b). Each firm had created dedicated managers in the form of project managers, technical managers, or systems architects, who were responsible for pooling diverse resources and skills together to develop complex Internet services and managing the acquisition of related necessary skills via learning and grafting (c.f. Appendix C).
In consequence, all firms reported that the sheer volume and scope of knowledge that they had to tap into to successfully carry out development had increased dramatically across various specializations (ICSD1a-c, ICSD2a). Organizations allocated significantly more resources to identify, garner, and absorb knowledge within each IT innovation set covering advances in IT base, software development, and service families and domains. Due to the volume of this change, they argued that their current resources were becoming more inadequate. Interviewees told that the proportion of relevant technical and business knowledge they were exploiting had actually decreased because the knowledge was so different and could not be integrated easily to existing knowledge bases. A manager from firm 2 stated:

Our firm traditionally had extremely good knowledge sharing practices in place... [now we] have just not been able to keep up with technology and the way that technology has been spreading rapidly and diversifying into different subgroups.

As a result firms had less specialized knowledge and skill redundancy for development. In some cases, a single individual held all critical skills. Firm 6, for example, noted that they had one expert for many critical technologies and worse still, they "don't have enough people yet to say that we have very good expertise on every technology we employ."

During the interviews, the firms were specifically asked whether they had garnered specialized resources for each of the four areas identified to be critical in Internet computing in Tables 2 and 3: telecommunications capabilities (ICSD2a), graphical and media skills (ICSD1a), business process change skills (ICSD1b), and new types of (traditional) software development skills (ICBase2b). As a result, a new critical role of an architect or director had emerged, as Lyttinen et al. (1998) proposed. This person's task was to orchestrate diverse resources and processes into a final service. This new role was seen as a necessary by-product of increased development complexity during Internet computing. Therefore, firms were asked about the existence of such organizational innovations. As noted in the table in Appendix C, seven of the eight firms had to garner and orchestrate deeper specialized resources in telecom (ICSD2a), art/media (ICSD1a), and what they called BPR skills (ICSD1b). These new specialists were deemed necessary because the spectrum of new expertise was beyond the scope of any individual's capabilities and it was needed on nearly every project. When asked, if it was possible for one person to have expertise across all areas in Internet computing, a developer in firm 1 answered, "No, I believe that would be a rare situation." He went on to say:

I typically do not see an expert in one area [who is capable of being] an expert in another. I do know a few rare people that can be labeled as experts in more than one area, but the areas are fairly close in nature [such as] J2EE and XML, or Java and Database, etc. I typically do not see someone who is familiar with say business processes as being an expert in software development and vice versa.

Experts in each area were regarded to be both valuable and scarce, and they were seen as critical for learning new technologies and their uses, and for transferring this knowledge to the rest of the organization. The architect in firm 2

19The major exception is for firm 4. Firm 4 did recognize that ISD complexity was increasing (ICSD1c) and that they consequently required a careful orchestration of projects (ICSD2b). Likewise, the company also recognized the critical need for deep knowledge in the areas of telecom (ICSD2a), interface (ICSD1a), and business processes (ICSD1b). But, unlike other firms, firm 4 did not garner specialized resources. Instead they preferred to maintain their tradition of having overlapping skill sets. Our data set does lend some possible explanation into why the firm behaved differently. The culture in firm 4 had always been one where knowledge sharing was considered a major part of the reward system (often at the expense of competitive salaries) and, therefore, they were resistant to adopt practices that would fragment people into specialized skills. It is also possible that the nature of their business (they provided a niche product and had a near monopoly on a fixed market) allowed them to maintain a broad specialization. They were seeing timelines shorter similar to other firms, but they had more flexibility in their deadlines and did not see the same degree of time pressure.
described the need of specialists in dedicated new areas as follows,

We prefer to have them. But we cannot currently employ them full-time on just one project [at a time]. So they basically travel between projects and guide each project.

The most radical shift appeared in the area of garnering and exploiting graphical and artistic expertise. In three of the six mature firms (firms 1, 7, and 8), specialized media resources were added onto their existing project structures through outsourcing. Firm 2 built their own expertise through grafting and internal training. Firm 3 recognized the need to have specialized media skills and separated media design tasks from software developers by reassigning these tasks to analysts. They soon discovered, however, that the business analysts were not capable of diligently performing both Internet computing process analysis and media tasks. As a result, during the week prior the interview, the firm had decided to take the media tasks away from the business analysts and hire expert media designers from outside the firm. This decision coincided with grafting expertise in technology integration (ICSD11c) via the hiring in of a new chief technology officer.

Interviewees indicated that the new divisions were caused by Internet computing. Evidence of this was most readily seen when we compared the two startups (5 and 6) with two of the older software firms (3 and 7). The two startups were initiated at the beginning of Internet computing, and both found it necessary very near to their inception to build deeper resources along the four areas of telecommunications, graphics, business process engineering, and software. One of them even designed their skill bases along these four dimensions with nearly these exact terms (see Appendix C). In contrast, the older firms observed that these four specializations were necessary and that they lacked expertise in some of the four areas. As a result, they concluded that they needed some combination of either outsourcing along these areas or rapidly hiring new people to graft skills in these areas. These older firms were slower than the startup firms in adopting the new skill bases as a means to manage their competency portfolio. As a result, these older firms more often depended on external resources either by renting deep expertise or by grafting it onto their existing skill base.

### Internet Computing as Disruptive Service Innovation

Both Lyttinen et al. (1998) and Turoff and Hiltz (1998) predicted that a very different concept of services would emerge during Internet computing.

Our data tells largely the same story and lends strong support for Conjecture B. The companies experienced a profound change in the services being demanded and built, as well as in how their customers viewed these applications. They acknowledged that they were building systems that differed significantly from those that they had built prior to Internet computing. An interviewee in firm 2 clarified the change:

The Web enabled us to connect to legacy systems through a non-legacy interface. So it has changed dramatically what we know already. In addition to that, that’s a positive side effect. With the introduction to the Web, [we have learned that] legacy systems…cannot be legacy anymore and they need to interact. Which has forced introduction of XML, etc.…The Web has made it possible [to] share technologies worldwide in a manner and fashion we were not able to do before.

An interviewee in firm 8 characterized the difference as follows:

The Web interface [has to be] more forgiving than…[a] normal client/server application. And the thing that makes it a challenge is to make the user interface

---

20As stated, however, the types of services created were not specifically identified in these studies.
so easy to use and so fast that they really don't have to ask [for] help; they really don't have to push the help button... [because] it might be your grandmother is using it.

The most significant change seen was embodied by the emergence of the two firms in the sample that did not exist prior to 1995. These companies were established specifically to fill unmet demand in software services brought about by Internet computing (and were referred to at the time with terms such as e-Business consultants). Examples of new systems built by these firms included business-to-consumer (B2C) applications that extended existing products from traditional telephone and person-to-person sales channels for a financial company; middleware and front-end applications to link WAP-based B2C; business-to-business (B2B) services; and business-to-employee (B2E) intranet solutions with legacy systems (thus demonstrating both radical and pervasive innovation in services).

For the remaining six firms, the types of services they were developing had changed as well. Each company reported significant changes in their system portfolios and across services. Before, they had each been developing either mainframe or client/server-based applications. Recently, their customers had started to demand solutions that benefitted from many characteristics of base technology innovations and utilized generic service capabilities of ubiquity (ICBase3a), and media orientation (ICBase3b).

In sum, the services being built by the eight firms were typically Internet computing applications that had browser interfaces (public B2C Web systems, B2B extranet applications, and B2E intranet services). While two firms did not develop services in each ICS1-4 area (firms 3 and 4 were making services for a single parent company and were involved in only B2C and B2B applications), the remaining six firms did develop services consistent with each type of service (ICS1a–ICS4a). Collectively, these services span all IT innovation types outlined in Table 4. Examples of some specific services developed in each area by the firms are listed in Table 5.

The firms noted that these innovations would not have been possible with earlier computing capability, thus demonstrating a radical departure in services delivered and the weak form of causality suggested by our Conjecture B. None of their clients had previously requested systems for such a broad range of end users (specifically those that interacted with consumers as end users), for such a large number of concurrent users (scalability), and with expectations of 24-hour-a-day availability and reliability. At the time of the interview these intranet, extranet, or public Web systems were the ones either exclusively or most commonly requested.

The developed services had radically impacted their client organizations. In one case, using the Internet computing software developed by firm 4, a company stopped selling their services at a fixed annual rate and instead became an Application Service Provider (ASP), selling their services by the transaction fees. Becoming an ASP was not possible before a wide adoption of Internet computing. In another case, a company implementing an internal purchasing intranet application from firm 8 saved "somewhere around 200 million U.S. dollars, in one year" in contrast to using their traditional computing systems. The technology manager in firm 8 strongly believed that this system and its benefits were not possible prior to Internet computing and, hence it was a necessary condition for these changes. In a third case, services from firm 5 transformed a lower-tier financial services company into the second largest trading company in the world with "transactions over $2 billion every single day" by creating its Internet presence. Again, this business model was not possible prior to Internet computing and thus represents a radical change. These glimpses support the widespread claims that Internet computing has radically transformed many services and their adopting organizations.21

21For example, in South Korea, approximately 9 percent of all consumer retail is now done via the B2C channel that did not exist prior to 1995 (Belson and Richtel 2003). Likewise, in 2002, 45 percent of all low-volume-buying organizations polled said they cut their costs significantly by using Internet-computing B2B services (http://cyberatlas.internet.com/markets/b2b/article/0,,10091_965041,00.html).
Table 5. Examples of Radical Innovations in Services

<table>
<thead>
<tr>
<th>IT Innovations in Services</th>
<th>Examples of IT Innovations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICS1a: Unforeseen applications for administrative core</td>
<td>Intranet (non-strategic): Web-based enterprise reporting tool—system is used to track items such as human resource information (turnover, sick time, etc.) across an organization independent of client location, access point, or platform.</td>
</tr>
<tr>
<td>ICS2a: Unforeseen applications for functional integration</td>
<td>Intranet (strategic): Web-based balanced scorecard system. Tracks key performance indicators that are considered core to business line of service firm. System based on EJB architecture utilizing Java Server pages as a front end. One of the advantages of the EJB architecture is that it enables the system to run on most database systems, including but not limited to SQL Server 7 and Oracle 7.x+. Utilizing the Java server pages as a front end enables the application to be completely browser-transparent.</td>
</tr>
<tr>
<td>ICS3a: Unforeseen applications for customer process integration</td>
<td>B2C Internet computing applications (strategic): e-Government applications. Services of government were expanded to allow tens of thousands of citizens to concurrently submit documents and request public services over the Internet independent of client platform (including mobile service) or the time of day.</td>
</tr>
<tr>
<td>ICS4a: Unforeseen applications for interorganizational integration</td>
<td>B2B Extranet applications (strategic): B2B digital marketplace application for e-procurement in newspaper industry. Application aggregates purchases of such items as newsprint and ink across the entire newspaper industry in order to lower transaction fees and prices. System is available 24 hours a day and is independent of location, client, and platform independent.</td>
</tr>
</tbody>
</table>

Discussion and Conclusions

Main Findings

We have tried to address two questions: (1) How can we model disruptive IT innovations that help us identify and analyze qualitative changes in IS development due to discontinuities in computing capability? (2) To what extent can the observed turmoil among systems development organizations spawned by Internet computing be understood as disruptive IT innovations? By addressing these two questions, the paper contributes to the IT innovation literature in two ways. First, it suggests a model of disruptive IT innovation that helps distinguish and analyze differences in dynamics of IT innovation types. Second, it applies the concept of Internet computing to this model and by doing so provides a thick empirical investigation of innovation behaviors and outcomes as experienced by eight system development organizations that were early adopters of Internet computing.

By focusing through the theoretical lenses of the industrial innovation and the IS innovation diffusion literature, the paper developed a model of disruptive IT innovation. The model fills an important gap in the IS innovation research, which so far has overlooked the significance of kinds of IT innovations in the technology base that act as drivers of other IT innovations. In brief, not all IT innovations are born equal and IS research should recognize this. At the same time, the model suggests that technology characteristics need to be included as important antecedents in IT innovation analyses (push). IS scholars need to move...
beyond pull-driven analyses of managerial choice (Markus and Robey 1988), and the resulting purely social explanations of IT innovation. A careful examination of changes in computing capability should form an important missing ingredient in IT innovation analyses.

The suggested model traces explanations of origins of IT innovations in services and development processes under some strict conditions to antecedent architectural changes in computing capability. Therefore, it does not lean on any variant of technological determinism. The articulated conditions for disruptive IT innovation are strict in the sense that they observe that antecedent technological change can only be a necessary but not a sufficient cause to create subsequent innovations in development processes and services. These strict conditions also observe that only specific sets of relationships between antecedent architectural base innovations and outcome IT innovations qualify as disruptive IT innovations. Changes in behaviors related to development processes and resulting services must be simultaneously pervasive and radical. By recognizing only necessary conditions for a disruptive IT innovation, the model cannot be used in a strict sense to predict disruptive IT innovations—it only provides a set of criteria that help identify when such a disruption took place, or to identify potential conditions under which such changes could take place. Additional sufficient conditions must be woven into detailed investigations around disruptive IT innovation events. These would cover an examination of unmet and unexplored user needs, adequate investments into co-specialized assets, and the creation of socio-technical networks that enable learning around new technology.22 Hence the model allows socio-technical and process-oriented explanations to be woven together to describe how specific disruptive IT innovations took place. Such explanations recognize that additional factors and processes related to timing, complementation, learning, and path dependency must be interlaced into innovation analyses. Overall, we feel that the analysis of disruptive IT innovation provides a good starting point to answer the first research question.

To address the second research question, we used the developed model of disruptive IT innovation to probe IT base changes related to Internet computing and its impacts on services and development processes. We observed how Internet computing has introduced a new architectural computing principle and enlisted a set of accompanying critical capabilities. We also identified a consequent set of IT innovations in services and development processes based on the analysis of the extant literature. Our analysis shows that Internet computing can meet all criteria of a disruptive IT innovation. First, innovations in services and development processes have been driven by new computing capability offered by Internet computing. Second, these innovations are both radical and pervasive (i.e., they exhaust all IT innovation sets). Overall, our analysis shows that a disruptive IT innovation model can be used to understand changes that have accompanied the adoption of Internet computing, and it par excellence offers an example of a possibly disruptive IT innovation. We strengthened this analysis with a rigorous and confirmatory multisite case study of eight leading-edge system development organizations and changes in their development practices and outcomes during the dot-com boom. Unlike earlier studies of Internet computing, the study is theory-driven and forms an integral part of a theory-generation and a theory-validation cycle. The study lends rich and systematic support to how Internet computing has spawned disruptive changes in services and development processes. The study also lends support to the conjectures derived from the disruptive IT innovation model: (1) Internet computing has pervasively influenced systems development and its outcomes, and (2) the influence has been radical in the sense that after adopting Internet computing IT innovators mobilized a different set of skills and resources and engaged in different sets of behaviors. The study echoes and confirms earlier findings and speculations in recent studies.
of Internet computing (Alter et al. 2001; Baskerville et al. 2001; Carstensen and Vogelsang 2001; Glass 2001; Isakowitz et al. 1998; Lytinen et al. 1998; Lytinen and Yoo 2002; Turoff and Hiltz 1998). The study also lends empirical support to the disruptive IT innovation model by showing how adoption of innocent but significant architectural changes in computing capability can trigger a grander wave of related IT innovations. This happened under the following conditions: (1) changes in the IT base innovation embodied a new architectural principle and expanded computing capability significantly by creating a new design space for IT innovators, and (2) the change integrated latent computing forms and possibilities which could satisfy unmet and unexplored user needs.

Discussion

Internet computing is not merely a fashion (per Abrahamson 1996; Newell et al. 2000), even though it was very fashionable at the time of the study. Unlike fashion, it has resulted in non-reversible and deep changes in computing infrastructures, development practices, and cognitive framing of computing problems. It offers a new type of computing capability that can be embedded in many types of IT innovations, thus permitting a high degree of interpretative flexibility (Bijker 1987). Hence, the idea of viewing IT innovations as fashions is insufficient in explaining the adoption of Internet computing and its outcomes. At most, fashion theory can explain premature adoption decisions of Internet computing in both development and adopting organizations, and the cultural hype around such adoptions.

Disruptive IT innovation does not imply that everything will change in services or development processes. Past discussions around the impact of Internet computing often see changes in these dichotomous terms: either everything changed or nothing changed (for examples, see Alter et al. 2001; Pressman 1998). Our analysis denies such simplistic conclusions. The reality seen in the studied companies was that many things had changed, while many things had remained the same. In light of the disruptive IT innovation model, this paper focused solely on changes that could be derived from the model of disruptive IT innovation. The main focus of the investigation was on what services the studied organizations had developed and with what types of new capabilities and behaviors, if any. Hence, we analyzed whether significant changes in services, or development processes had taken place. Indeed, this was what we observed over and over again. The interviewees indicated that they lived in a different world (Kuhn 1996) that had different standards of performance, and relied on different concepts of service (Christensen and Bower 1996).

We also observed some things that had not changed, although this was not the focus of our investigation. For example, recurrent issues in systems development like the need to control development cost and outcomes, the need to address uncertainty and risk, the need to deal simultaneously with multiple design constraints (cost, quality, speed), and the need to maintain discipline over detail had not changed. Because the importance of these residual issues has not changed, some critics of Internet computing have suggested that Internet computing is not different from earlier forms of computing (Pressman 1998). Our conclusion is that these critics are partially correct if we want to emphasize the stability of some tasks and problems in developing systems. In line with this, the companies emphasized the criticality of project management skills, design skills, and thorough command of technologies as critical success factors. Yet, at the same time, they admitted that the context in which these issues had to be addressed had fundamentally changed. Most prominently, the increased development speed undermined old ways of managing projects and the diversity and speed of change in technologies undermined traditional ways of improving designs, or of exercising of quality control. Consequently, we observed a constant struggle in the companies to strike a balance between radical innovations and incremental improvements (Holmberg and Mathiassen 2001). For example, all companies with rigid methodologies admitted that their established development methodologies did not work. At the same
time they all were striving to develop simpler process frameworks, which could help them improve their development processes and make them more repeatable. Hence, there is a need for developing new types of Internet computing methodologies (for examples, see Isakowitz et al. 1995; Lee and Suh 2001; Schwabe and Rossi 1995). However, most of the published work does not carefully address the radically different nature of new methodologies including simplicity, agility, and concern for flexibility.

**Limitations**

The first limitation deals with generalizability of the findings. Because this paper is engaged both in theory generation and validation, we must ask how generalizable the findings are from multiple angles. First, it is not clear to what extent the developed innovation model can be generalized to other theoretical contexts. By drawing upon multiple streams of theory building, we were able to develop an appropriate set of constructs that apply to disruptive IT innovation. Its theoretical generalizability to other types of contexts needs to be more carefully observed in future studies. Second, we applied the model to study the impacts of Internet computing in an early adoption cohort. Due to the nature of our sample, it remains unclear to what extent the findings can be generalized to the entire adopter population. Without further research, we can only faithfully generalize our findings to first movers, or to early adopters in the next disruptive wave (whatever form that takes). In later adoption cohorts, IT innovations related to adopting Internet computing may take different forms, and its adoption may not necessarily trigger similar effects. The radical nature of Internet computing may weaken over time if (and when) Internet computing comes of age, and standardized co-specialized assets of conceiving and developing services emerge. In relation to non-adopters (i.e., the lack of a control group in our study), we can refer to other studies and experiences from our fieldwork, which point out that similar (predicted) changes had not taken place in non-adopter organizations. It is important to reiterate that our sample included organizations that continued to develop non-Internet computing software in ways that are more traditional. These firms witnessed that processes and outcomes in those traditional parts of their organizations had not been affected similarly to the ways Internet computing had seen change. Hence, we are relatively confident that the circumstances that establish necessary conditions for observed change were faithfully detected. A third issue of generalizability deals with how it can be applied to understand other waves of disruptive IT innovation. The disruptive IT innovation model, as outlined above, is broader in scope than what has been said about the effects of Internet computing. Therefore the model should be transferable to analyze other (earlier or later) disruptive innovation processes. This requires, however, that appropriate instantiation rules as formulated in Tables 2, 3, and 4 for Internet computing need to be developed for other disruptive IT technologies.

The second limitation is a product of the organizations studied. Results concerning the radical content and pervasive scope of changes in IT innovations are based on data collected from organizations that develop services, not from those that adopt and exploit them. Due to the confidentiality restrictions we could not expand the data collection to organizations adopting services. This task remains an untapped research challenge because the observations concerning the disruptive nature of IT innovation are collected from multiple sources covering both developing and utilizing organizations.

The third limitation involves the possibility that our data collection method may bring both a selection and timing bias concerning the disruptive nature of Internet computing. Assessments of the radical nature of a specific technology are always sense-making exercises that are bound to the specific context, and the selection of evaluators within that context. In our case, we interviewed early adopters of Internet computing who operated under strong incentives that related to the success
of these technologies. Usually, in situations like this, the pushers of technology overestimate its originality and impact.\textsuperscript{23} The same holds true when collecting data about historical epochs where the technology has become fashionable. Finally, the interviews were carried out at the peak of the e-business fever characterized by hyper-competition, abundant financial resources, and fast-paced market change. Hence, using promoters of technologies, which at the time of the study were fashionable, under the circumstances of hyper-competition may have resulted in overestimating the disruptive nature of Internet computing. There is no way our study could remove the influence of such contextual bias. Therefore, some statements of the revolutionary nature of the changes may have weakened after the recent whims within the markets that deal with Internet computing. It is important to note, however, that we sought systematically to reduce such bias during interviews by including in our sample traditional companies that had developed software for over 40 years. Most companies in the sample had developed software in traditional ways for traditional platforms and most interviewees had extensive experience of such development. Therefore, during the interviews we purposefully contrasted the respondents’ experience with Internet computing with their experiences with earlier computing platforms. We have visited these companies several times since the study and observed that their interpretation of the disruption has not changed (see Epilogue below).

**Managerial Implications**

Our findings have several managerial implications. First, managers need to recognize that architectural innovations set up unprecedented demands of shifting competency bases. Depending on the level of disruptive impact, such demands can become overwhelming and require solid leader-

ship. Companies consistently mentioned this as their foremost challenge and were developing distinct strategies to address this, including the creation of separate innovation units and hiring only top-level experts. Second, managers in system development organizations must understand the importance of correct timing in managing the diffusion of disruptive innovations—they cannot wait too long if they want to shift into new technologies and services, because the learning curves are steep. Finally, organizations must understand what the true implications of architectural innovation are: they reshape how organizations will think, what types of systems they need to develop, and how they develop the systems.

In a more general sense, research that develops theory explaining specific IT innovations in the context of the broader business environment is helpful in three ways (Swanson 1994). First, it provides an understanding of the role IT can and will play in the larger business environment. In the case of this paper, this includes the IT base technology changes and the subsequent changes in services and development processes. Second, a good IT innovation theory allows managers to better shape their units with respect to expectations for IT innovation. Third, it allows managers to better ascertain when it is good to be among those who lead, and when, in contrast, it is good to be among those who follow, learning from the accumulated experience of their predecessors (Swanson 1994, p. 1089)

**Future Research**

The concepts raised in this paper indicate a need for additional research. First, we feel that the model of disruptive IT innovation is only a preliminary step toward a more encompassing dynamic theory of IT innovation. Therefore, IS scholars should engage themselves in more theoretical analyses of how IT innovations emerge and how

\textsuperscript{23}History of computing is full of such predictions from the invention of high-level programming languages to the madness around “expert systems.”
they interact. This demands that we develop adequate typologies of change in computing capability, and explore carefully how these changes are transformed into new IT services and development processes. Future research should investigate such questions as:

- What are the necessary and minimum components that differentiate a major architectural change from an incremental IT innovation?

- What are the factors that drive architectural and incremental innovation and to what extent are they different?

- How do IT innovation sets interact over time, and what are their impacts at different stages of adoption?

Future IS research also needs to include more empirical investigations of macro- and micro-level processes that enable and drive architectural IT innovations, as well as their subsequent absorption and refinement. Future studies should also be extended to allow generalizations across a set of both pushing and pulling organizations of different adoption cohorts. Such future studies could help validate behavioral differences between adopting and non-adopting organizations on both the pull and the push sides of a disruptive IT innovation.

Last, this study examined only the first shock wave of a disruptive IT innovation, but did not trace its continued deployment. Future work should investigate how the innovative potential of a disruptive IT innovation is exhausted over time in adopting organizations and how it influences specific IT innovation sets. To this end, we plan to conduct a longitudinal study in the development organizations to observe the changing pace and tone of continued IT innovation. Disruptive innovations also raise the question of how adopting organizations adapt to disruptions. What we observed in organizations we studied was a frenetic and chaotic learning activity—which we call hyper-learning—while their adoption decisions were relatively fast and painless. The organizations had to learn quickly, deeply, and broadly to absorb Internet computing. Circumstances demanded that they ride two horses at once—both technology exploration and exploitation. The critical role of knowledge and its volatility during disruptive periods should be investigated more carefully. Demands of knowledge exploitation seem to push governance structures on the one hand toward outsourced solutions because only specialized and innovative units can identify and absorb knowledge with adequate speeds, while on the other hand transforming internal structures of these organizations to highly organic forms.

**Epilogue: Success in an Economy Gone Bad**

The gloom in the global economy subsequent to the interviews has had only a limited impact on the firms participating in the study and shows the lasting impact of the adoption of Internet computing. Only two of the eight firms have gone out of business in spite of the economic recession that began in 2000, and neither of these two firms can be considered failed economic models. A larger competitor bought firm 4 to eliminate competition; its software products were purchased, and immediately phased out. The downturn in the economy did not eliminate this firm—its success did. Likewise, the parent of firm 5 went bankrupt even though the division included in the interview had been financially successful up until the parent company closed them down.

Of the remaining six, two smaller firms (1 and 6) experienced a sharp drop in customer demand as the economy shrank drastically in 2001. However, firm 1 saw its business improve in 2002 and 2003, and they are doing better financially and employ more developers than they did at the height of the dot-com boom. Firm 6 is similarly succeeding in spite of the tough environment and is continuing to focus on Internet computing solutions.

Each of the remaining four firms is experiencing success with Internet computing. Firm 2 merged
with another firm and demonstrated the value of their new skills. When the new merged company needed to eliminate redundant jobs, the employees from firm 2 were almost entirely spared at the expense of the developers from the other group. These other employees were not as skilled in Internet computing and were deemed expendable.

Firm 3 was seen to have successfully completed its mission of developing new services in the Internet computing realm for its parent company. Once the services desired had been created, the parent company decided to reabsorb them. Firm 3 had originally been given its autonomy in order for the group to innovate without the burden of the large parent. Not unexpectedly, firm 3 acknowledged they had lost their ability to innovate rapidly now that they are back in the larger firm. One developer indicated that during this time they have lost their competitive advantage. They had gone from "18 months ahead" to "12 months behind" the competition. However, the reabsorbed unit is focused entirely on Internet services and is now a prominent part of the parent company.

The remaining two firms (7 and 8) were still economically strong in 2003. These two firms had large market penetration going back 40 years or more in the IT services area in Finland. This advantage continued during the downturn. Each of the six remaining firms is still engaged in Internet computing and sees this as their future. For each of these surviving firms, the direction of change observed in the study has not deviated from our early observations.

Acknowledgements

We are grateful for insightful comments obtained from Dick Welke, Youngjin Yoo, Betty Vandenbosch, the senior editor, associate editor, and four reviewers. We want to thank all interviewees who participated in the study for their openness, interest in our research, and willingness to share their time that was ticking at the speed of Internet time.

References


MIS Quarterly Vol. 27 No. 4/December 2003 583


Tapscott, D., Ticoll, D., and Lowy, A. *Digital Capital: Harnessing the Power of Business*


About the Authors

Kalle Lyytinen is the Iris S. Wolstein professor at Case Western Reserve University in Information Systems, and an adjunct professor at the University of Jyväskylä, Finland. He serves currently on the editorial boards of several leading IS journals including Journal of the AIS, Information Systems Research, Journal of Strategic Information Systems, Information & Organization, Requirements Engineering Journal, and Information Systems Journal. He has published over 150 scientific articles and conference papers and edited or written eight books on topics related to system design, method engineering, implementation, software risk assessment, computer supported cooperative work, standardization, and ubiquitous computing. He is currently involved in research projects that look at the IT-induced innovation in software development, architecture and construction industry, and is developing a high-level requirements model for large-scale systems. He is also engaged in a project supported by NSF focusing on the institutional forces involved the development of global electronic commerce. His research interests include information system theories, computer-aided system design and method engineering, system failures and risk assessment, computer-supported cooperative work, nomadic computing, and the innovation and diffusion of complex technologies and the role of institutions in such processes.

Gregory M. Rose is an assistant professor in the College of Business and Economics at Washington State University. His current work focuses on development, design, adoption, and use of Web-based systems across various nations and cultures in the Americas, Europe, and Africa. He has published in such journals as Accounting, Management and Information Technologies, IEEE Transactions on Engineering Management, Information Systems Journal, Journal of Global Information Management, Electronic Markets, and Communications of the AIS. He holds a Ph.D. in Computer Information Systems from Georgia State University, was a postdoctoral fellow at the University of Jyväskylä in Finland, and serves on the editorial review board of the Journal of Global Information Management.
Appendix A

Interview Script

I. Background information
   Please tell me about your corporation:
   1. Type
   2. Size
   3. Age
   4. Brief history

   Please tell me about your IS organization:
   5. Size, typical work week, and turnover per year
   6. Budget
   7. Organizational structure

II. Specifics about your IS Development (please indicate if you have multiple techniques and strategies depending on the project type—such as traditional systems vs. e-Commerce systems and explain why)
   8. Please tell me about your application portfolio (describe the functionality and technology)
   9. Please tell me how you manage your application portfolio (as it exists now and as it grows, what do you do to track and manage it)
   10. Tell me about reuse in your organization (software code, patterns, methods, etc.)
   11. Please tell me about the technology platforms, tools, and programming languages used (for both development and end-users), and the specific reasons for choosing them.
   12. Please tell me about your IS organization’s current development methods, project management practices, and their evolution
   13. Please tell me about your application architectures and possible design patterns used (architecture means here a specific way in which different components of the software system are designed, e.g., distribution of data, control and processing, or means for relating different software system parts)
   14. Please tell me about what training the IS team members receive prior to joining and during their time working here.
   15. Please tell me about your interactions with the customers—do you have interaction and implementation strategies for different phases of the project. (i.e.) Please give details of how you initiate contact, record contact, with whom you make contact, and why is the process done this way.
   16. Please tell me about your insourcing /outsourcing issues—who and why?
   17. Please tell me about your skills distribution, roles, task/role assignments [Specifically, do they break down according to the TABS model of deep resources in Telecom, Art or the graphical interface, Business process, and traditional Software development.]
   18. With regard to your development management:
      a. Are processes mostly ill-defined and does success generally depend on specific individual effort and heroics? [If no, then continue; else go to q. 19.]
      b. Do you track costs, progression through a schedule, and completed functionality? [If yes, then continue; else go to q. 19.]
c. Does an approved, standard software process for developing and maintaining software exist? Is it uniformly used? [If yes, then continue; else go to q. 19.]

d. Are detailed measures of the software process and product quality collected?

III. Web development practices (did you change when you moved to development for the Web)
19. Please tell me about how the Web changed your technologies/platforms/standards used
20. Please tell me about how the Web changed your development strategy, development phases, goals, deliverables
21. Please tell me about how the Web changed your project management practices and your organization
22. Please tell me about how the Web changed your methodological guidelines, tools and environments used, reuse strategies, and prototyping methods
23. Please tell me about management of your Web application portfolio

IV. Implementation of web enabled practices
24. Please tell me about organizational strategy followed in introducing and carrying out the changes you described above due to the Web. Please make sure to tell me about your “selling the idea” to management.
25. Please tell me about your major inhibitors/enablers in transforming organization
26. Tell me how the Web-centric practices you just described impact the speed of development when compared to previous practices used or alternative ones you have seen [Include questions on how speed is increased such as: How do you do that? Do you increase the number of people on the project? Do you reuse, do you buy components, outsource? What do you do?]
27. Tell me how the Web-centric practices you just described impact development quality when compared to previous practices used or alternative ones you have seen
28. Tell me how the Web-centric practices you just described impact your application architecture when compared to previous practices used or alternative ones you have seen
29. Tell me how the Web-centric practices you just described impact user participation when compared to previous practices used or alternative ones you have seen
30. Tell me how the Web-centric practices you just described impact user friendliness when compared to previous practices used or alternative ones you have seen
31. Tell me how the Web-centric practices you just described impact cost or benefits of the systems when compared to previous practices used or alternative ones you have seen
32. Tell me how the Web-centric practices you just described impact application quality (portability, scalability, reliability, maintainability) when compared to previous practices used or alternative ones you have seen
33. Tell me how the Web-centric practices you just described impact business value/assessment when compared to previous practices used or alternative ones you have seen
34. Tell me how the Web-centric practices you just described impact risks/benefits when compared to previous practices used or alternative ones you have seen

V. Web enabled future—how will changes which have occurred or are expected to occur in Web technology effect you in the next 5 years
35. Please tell me about major likely changes in the system delivery mechanisms (these are changes in the tools, methods and practices by which systems are developed and maintained) in the next five years
36. Please tell me about major likely changes to your technological base (moving to different run time or development environment)
37. Please tell me about major likely changes to your application scope/functionality/size 
39. Please tell me about major likely changes to your development processes 
40. Please tell me about major likely changes to your business or your clients 

Appendix B

Research Methodology Details

B.1 Choice of a Research Methodology

A multisite case study was justified because we were focusing on a contemporary phenomenon with strong contextual dependencies. We followed Fichmann’s suggestion that IS innovation researchers need to examine “fewer organizations, but in greater depth using replicated case study, or ethnographic research methods” (Fichmann 1992, p. 204) when dealing with technologies that are advanced, and which relate organization level behaviors. A multisite case study enables researchers to carry out multiple tests for the existence of subsumed regularities. Each site was interpreted as a separate analysis point to validate the theoretical model. This was also justified by our understanding that we were investigating “a too varied and subtle phenomenon” (Fichmann 1992 p. 204) to be studied with cross-sectional methods.

There were specific concerns of both validity and reliability related to our research design, which justified the choice. Because of the fast change and ambiguity related to organizational exploration and exploitation of Internet computing, we felt it necessary to grasp actors’ own understanding and experiences instead of using a structured questionnaire. We felt that this was particularly important in analyzing the radical nature of IT innovation, and how actors recognized dependencies between different innovation sets. We felt that the risk of missing critical information and thereby decreasing the internal validity of the findings due to poor construct validity was too high. Because we were focusing on disruptive behaviors—which by definition are broad and varied in forms—it would have been impractical to use survey based research. In addition, we felt that it is difficult to anticipate innovation forms across a set of organizations. We therefore saw value in gathering an open qualitative data set to understand what shapes the innovations were taking. Overall, the case study format enabled the use of multiple sources of evidence to improve construct validity (Yin 1994).

Entering into the field and collecting a rich data set enabled data to be understood in context and to be accounted for observed differences. This improved internal validity of the findings as it helped exclude alternative explanations (Yin 1994). The rich data also enabled process accounts to be developed for how the changes were being carried out and how the studied organizations responded over time to innovation.

B.2 Sampling and Data Collection

As noted, we followed a purposeful (theoretical) sampling strategy (Patton 1990) that increases the generalizability of results. It also serves the purpose of improving the external validity of findings by using a replication logic (literal replication) whereby the studied phenomena can be generalized into a theory of disruptive IT innovation (Yin 1994). Thereby, “ideal” types of organizations were sought out to be included in the sample of replicated “examples.” These cases were to “be representative of a presumably large
class of cases that fits with the requirements of a theory or theories to be tested” Markus (1989, p. 24). Sampling data within organizations that met these criteria achieved two important goals: it enabled new information about constructs of interest (like types of applications developed) to be obtained, and enhanced confidence in the measurement of the constructs through constant triangulation. By doing so we could aggressively use all strategies outlined by Patton (1990, pp. 182-183) to improve construct validity: criterion sampling (using interviewees and selecting questions based on predetermined criteria), theory based sampling (picking interviewees that were pertinent to constructs including radicalness and pervasive nature), chain sampling (checking that we went to right organizations), and opportunistic sampling (including data and interviewees as they emerged during the field work).

To minimize potential bias in our study we sought to maximize the variation in firm characteristics in our sample (characteristics of the firms sampled are shown in Table B.1). Therefore the sampled companies had different sizes and operated in various industry sectors ranging from manufacturing, financial services, public administration, to retail and transportation. The geographical scope of their operations varied largely as some were local software houses, while some were parts of large multinational companies. The companies also had large variation in their age in that their time of operations ranged from as few as four years to as many as 40+ years in software development.

We collected our data by carrying out data-intensive interviews in each company. Each interview lasted from one hour to three hours, and followed a semi-structured interview format shown in Appendix A. Most interviews were approximately two hours in length. The number of people participating in the interviews ranged from three to six and they were selected based on their knowledge and expertise in the area. The people had job titles like technical manager, technical expert, or business manager for the Web business. Most interviewed people had extensive experience from the field and made decisions about technology investments in studied companies, or were in charge of the strategic directions of the organization. They managed organizational knowledge bases and skills needed to execute a chosen technology and business strategy. When possible we tried always to include both managerial and technical experts into the interviews to remove the possible managerial “bias” when interviewing only the managers (Alvesson 1995). This guaranteed that we obtained deep and rich insight into the technologies and the system development practices followed as the technical experts we interviewed were all the time fighting in the forefront of such problems. In two cases (firms 5 and 8) we had only one manager that participated in the interviews. Both firms also oversaw their development projects and were heavily involved in the technical aspects of systems development and in this sense the results of these interviews did not significantly differ from others. Overall, we felt we met the goals of criterion sampling, theory based sampling, and chain sampling. We applied a realist ontology (van Maanen 1988) and tried to find out what participants said or did rather than what we thought they meant. We also collected background information of personal career histories, and sought to obtain personal experiences and observations of changes. The data collection was driven by the theory based propositions outlined above and expressed in Appendix A. Additional notes were made during the visits concerning physical sites visited, personnel age, general atmosphere of the company, etc. We also collected written or published materials from these companies including their yearly reports, Web sites, advertising materials, manuals, or system handbooks to increase the quality of the data set.

24In four of the interviews (in Finland) we had two interviewers.
<table>
<thead>
<tr>
<th>FIRM</th>
<th>USA1</th>
<th>USA2</th>
<th>USA3</th>
<th>USA4</th>
<th>FIN5</th>
<th>FIN6</th>
<th>FIN7</th>
<th>FIN8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interviewee Details</strong></td>
<td>Six senior employees including an executive, managers, and software architects.</td>
<td>A senior manager of an IS development group and one of his key developers.</td>
<td>The CIO, and the key five senior technologists who were responsible for the creation of the spin-off.</td>
<td>Two employees consisting of the senior systems architect and a senior developer.</td>
<td>Five senior employees including ISD project managers, developers, and the senior technology architect.</td>
<td>One of the founding executives who was responsible for development of business processes.</td>
<td>Four senior employees including a systems architect, manager, and software engineer.</td>
<td>One senior manager of IT development services.</td>
</tr>
<tr>
<td><strong>History</strong></td>
<td>15-year-old firm had been mainframe and client server shop with 500 employees in 4 locations.</td>
<td>Part of a large, multinational business consulting company.</td>
<td>Part of a large financial company with many thousand employees.</td>
<td>10-year-old financial industry software company. Revenue traditionally came from sales of software. Now ASP.</td>
<td>Multinational e-Business consulting firm founded in 1995 with several thousand employees.</td>
<td>e-Commerce development firm founded in 1996 starting with 6 employees.</td>
<td>Large multinational e-Business consulting and software development firm.</td>
<td>Mature, large, multinational development and IT service firm.</td>
</tr>
<tr>
<td><strong># Employees in Division</strong></td>
<td>Several hundred</td>
<td>Several hundred</td>
<td>70</td>
<td>28</td>
<td>100+</td>
<td>200+</td>
<td>700+</td>
<td>Several hundred</td>
</tr>
<tr>
<td><strong>Typical work week</strong></td>
<td>40 hours</td>
<td>50 hours</td>
<td>50 hours</td>
<td>Had traditionally been 40 hours until the recent pressures of Internet time.</td>
<td>60 hours</td>
<td>37.5 hours</td>
<td>Varies</td>
<td>37.5 hours</td>
</tr>
<tr>
<td><strong>Employee turnover/year</strong></td>
<td>18-30%</td>
<td>15-30%</td>
<td>&lt; 10%</td>
<td>0% in last 18 months</td>
<td>3%</td>
<td>3%</td>
<td>Uncertain</td>
<td>Uncertain</td>
</tr>
</tbody>
</table>
Table B.1. Firm Characteristics (Continued)

<table>
<thead>
<tr>
<th>FIRM</th>
<th>USA1</th>
<th>USA2</th>
<th>USA3</th>
<th>USA4</th>
<th>FIN5</th>
<th>FIN6</th>
<th>FIN7</th>
<th>FIN8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Organizational Structure</strong></td>
<td>President, Branch manager, Field manager, Project manager</td>
<td>Partner, Director, Project and technical managers</td>
<td>CIO, then flat</td>
<td>VP (architect), then flat.</td>
<td>Client manager, Project manager</td>
<td>Entirely flat except for salary issues.</td>
<td>Rigid vertical hierarchy with formalized methodologies for all aspects of business.</td>
<td>Company is divided into autonomous units based on market sector of client.</td>
</tr>
<tr>
<td><strong>Project Team Characteristics</strong></td>
<td>15-20 people including: business analysts, architects, lead developer, other developers, QA person.</td>
<td>Architects, analysts, expert developers, rookie developers.</td>
<td>Informal</td>
<td>Project teams create temporary hierarchies. Decisions based on consensus and volunteers.</td>
<td>Flat with the following roles: project assistant, technical lead, designer, information architect.</td>
<td>Informal</td>
<td>Rigid vertical hierarchy.</td>
<td>Broken down by customers (approximately 50/customer) and subsequently by teams (of 10 each).</td>
</tr>
<tr>
<td><strong>Miscellaneous</strong></td>
<td>80% of their business is repeat.</td>
<td>Roughly $750 million budget.</td>
<td>Spun off to make decisions at faster pace to meet needs of market.</td>
<td>Within 12 months of joining organization, employee can expect to make as much salary as highest paid person.</td>
<td>Company expands through vertical and horizontal acquisition.</td>
<td>20-30 projects ongoing at any time.</td>
<td>Mostly analysts. Only 30 java coders as primary job. 200 doing HTML construction and Scriptlets, etc. Remaining development done by partners.</td>
<td>Offices in 13 countries.</td>
</tr>
</tbody>
</table>
Appendix C

Strategies for Coping with Specialization

<table>
<thead>
<tr>
<th>FIRM</th>
<th>USA1</th>
<th>USA2</th>
<th>USA3</th>
<th>USA4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICSD1a Specialized Interface skills</td>
<td>Yes. &quot;We have some third party partners we will bring in...[and] at a few various points in the past two years have had art experts on staff [that] were fairly specialized in web GUI or graphic design.&quot;</td>
<td>Yes. &quot;Typically the developers will do the web page and then we'll have a designer go in afterward and design it, putting on the nice buttons, changing the fonts, all that stuff...If you have a developer, that person's likely to know JAVA, but not have the best design skills.&quot;</td>
<td>Yes but also evolving to include more. Currently has been &quot;done by [specialists] on the business side [instead of by the programmers].&quot; Were looking in the next 6-12 months to get &quot;a high-end specialist to help us fine-tune our GUI making sure that the business, the users view and the business view is solid.&quot;</td>
<td>No. Is shared responsibility.</td>
</tr>
<tr>
<td>ICSD1b Specialized Organizational change skills</td>
<td>Yes. &quot;[We have deep] experience with re-engineering.&quot;</td>
<td>Yes. &quot;We have people that are totally focused on it and then we have developers who get trained in it but without being the experts. So if you want to do real [BPR] engineering, we're gonna bring in the real [BPR] engineering guys.&quot;</td>
<td>Yes. &quot;We have the business analyst role...we have some very seasoned people that do that sort of thing...his job is to come in and from his experience and looking from the outside and say 'what is that process?'&quot;</td>
<td>No. Is shared responsibility.</td>
</tr>
<tr>
<td>ICSD1c Specialized Software Programming Resources</td>
<td>Yes. They have specialists within different programming languages and design including technologies for &quot;COBRA, J2EE framework, or the DNA framework.&quot; Also by industry. &quot;Most of our developers are becoming proficient at distributed computing. Have to understand that level of development and the evolution of that, of course, is it's becoming much more complex now.&quot;</td>
<td>Yes. Specialization exists based on 'experience and knowledge or tools. Secondarily on the business problem we're trying to solve.'</td>
<td>Yes. Have specialists in software patterns and methodologies. But outsource for new technical development skills as the needs arise. Most notably, outsourced most of J2EE coding.</td>
<td>Yes. Have deep resources in various programming languages including Java and C and CORBA IDL.</td>
</tr>
<tr>
<td>FIRM</td>
<td>USA1</td>
<td>USA2</td>
<td>USA3</td>
<td>USA4</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>ICSD2a</td>
<td>Yes. &quot;We have some of that. I came out of the telecom industry and several other people did here. We've got some [regional Bell] folks around. I came out of [large telecom company].&quot;</td>
<td>Yes. &quot;We do have some deep resources that deal with the different telecom issues.&quot;</td>
<td>Yes. Had created new group to extend their business to be an Application Service Provider (ASP) to host their Internet services for their clients. &quot;We have a network or hosting operations group that deals with all that stuff...and then we have a team at [the parent company].&quot;</td>
<td>No. Is shared responsibility.</td>
</tr>
<tr>
<td>ICSD2b</td>
<td>Yes. Specialized skills for the roles of &quot;project manager, business analyst, data architects, and system architects.&quot;</td>
<td>Yes. A &quot;technical manager&quot; and experts who &quot;build the architecture...which is only very few number of individuals partly because it’s hard to get the experience. You need a lot of experience to design the architectures.&quot;</td>
<td>Yes but increasing. Hired a new CTO to act as architect and skills integration manager.</td>
<td>Yes. Vice president and senior architect manages integration and collaboration.</td>
</tr>
<tr>
<td>ICSD1a</td>
<td>Yes. Specialists in a group called &quot;Bracket of Experience&quot; with &quot;the graphical designers, the branding for being on information architecture, application design.&quot;</td>
<td>Yes. This group is called &quot;Media.&quot;</td>
<td>Yes. Outsource to specialists. &quot;In that web area...we are not so good in graphical design.</td>
<td></td>
</tr>
<tr>
<td>ICSD1b</td>
<td>Yes. Specialists in a group called &quot;Bracket of Strategy where basically have the business strategy, front-end strategy, service strategy, refined for our clients.</td>
<td>Yes. This group is called &quot;Organizational Communication.&quot;</td>
<td>Yes. Most of their employees are &quot;business consulting people&quot; with a focus on &quot;change management from the organizational point of view.&quot;</td>
<td>Yes. &quot;They are specialists [with] methods and work close to the work flow, work management...[they are] experts in the work flow business.&quot;</td>
</tr>
</tbody>
</table>