7-16-2008

Learning in High Gear: Hyper-learning and Dynamic Capabilityin Seven Software Firms

Kalle Lyytinen
Case Western Reserve University, kalle@case.edu

Gregory M. Rose
Washington State University

Youngjin Yoo
Case Western Reserve University

芽|Sprouts Working Papers on Information Systems ISSN 1535-6078

Learning in High Gear: Hyper-learning and Dynamic Capability in Seven Software Firms

Kalle Lyytinen
Case Western Reserve University, USA

Gregory M. Rose
Washington State University, USA

Youngjin Yoo
Case Western Reserve University, USA

Abstract
Building on the literature of dynamic capability and organizational learning, we examine strategy execution in hyper-competition as a problem of how organizations can re-configure their learning capability to match with their radically different learning demands. Organizations in hyper-competitive environments face an increasing gap between their learning opportunities and needs, and actual learning performance. In order to survive they must improve their absorptive capacity so that they can learn simultaneously broad, deep and fast. We define such a learning contingency as hyper-learning. To do so, the organization must systematically interlace explorationâ that seeks to maximize learning breadthâ and exploitationâ that seeks to maximize learning depth. Unlike in traditional learning cycles, exploration and exploitation during periods of hyper-learning are not insulated from each other through time or structure. We explore seven software firms engaged in Web system development during the hey-day on dot.com frenzy and investigate how these companies were able to hyper-learn. We distinguish two mechanisms to speed up exploration: distributed gate-keeping and extended grafting of external knowledge; and two mechanisms to speed up exploitation: simple design patterns and peer networks. These mechanisms were nearly uniformly recognized in all studied organizations. We also examine the systemic configuration and patterning of these activities, which enables organizations to learn in high gear. This organizational learning model is contrasted with the punctuated equilibrium model of learning articulated in mainstream strategy research. Finally some implications for future research and management practice are drawn.

Keywords: Organizational Learning, Strategy, Hyper-Competition, Knowledge Management, Systems Development, Technology Strategies

Permanent URL: http://sprouts.aisnet.org/2-16

Reference: Lyytinen, K., Rose, G.M., Yoo, Y. (2002). "Learning in High Gear: Hyper-learning and Dynamic Capability in Seven Software Firms," Case Western Reserve University, USA . *Sprouts: Working Papers on Information Systems*, 2(16). http://sprouts.aisnet.org/2-16

Learning in High Gear: Hyper-learning and Dynamic Capability in Seven Software Firms

Introduction

In today's hyper-competitive business environment (D'aveni 1994), a company's ability to quickly learn and adapt to the changing environments is key to its success and survival (Eisenhardt and Martin 2000, Grant 1996, Ilinirtch et al. 1996, Volberda 1996). In such an environment, traditional strategies that seek sustainable, competitive advantage in the long-term (Barney 1996, Chandler 1962, Penrose 1959, Porter 1980) are being challenged by Schumpeterian creative destruction (Schumpeter 1934, Young et al. 1996) and hyper-competition (D'aveni 1994). It is no longer sufficient to jockey into a unique position in an industry (Chandler 1962, Porter 1980) or to possess a configuration of difficult-to-imitate bundles of resources or routines that guarantee competitive advantage (Barney 1996, Nelson and Winter 1982, Penrose 1959, Peteraf 1993, Wernerfelt 1984). Strategic frameworks based on hyper-competition expect continuous and disruptive organizational innovation that continuously creates temporary disequilibria in the competitive system (Volberda 1996).

To date, research on hyper-competition has focused on the nature and structure of hyper-competitive marketplaces (D'aveni 1994) and has consisted of company-level studies that explore companies' strategies through high velocity decision-making and the creation of new competences (Eisenhardt and Tabrizi 1995, Eisenhardt and Martin 2000). What is striking in the existing research is the scant attention paid to hyper-competitive *strategy execution* or how organizations in high-velocity environments manage to rapidly acquire new competencies. This acquisition of new knowledge represents learning by the organization (D'aveni 1994), or rather the organization's learning to learn. Through this learning to learn process, the organization creates capabilities in order to acquire new capabilities (dynamic capability). A focus on strategy execution would enable the observation of micro-level learning processes whereby organizations foster learning. Through these processes, firms subsequently build up pools of resources and routines that either inhibit or foster the creation of dynamic capability.

Building on the literature of dynamic capability and organizational learning (Cohen and Levinthal 1990, Eisenhardt and Martin 2000, Eisenhardt and Tabrizi 1995), we examine strategy execution that takes place during hyper-competition. The objective is to understand how organizations re-configure their learning capability in order to radically match the new learning demands posed by hyper-competition. A primary assumption of this paper is that, during hyper-competition, organizations face an increasing gap between their *learning opportunities and needs* and *actual learning performance.* In order to survive and thrive in hypercompetitive situations, organizations must dramatically increase their absorptive capacity (Cohen and Levinthal 1990) so that they are capable of *hyper-learning*.

We define hyper-learning as an organizational contingency where an organization must concurrently learn *(a) broadly, (b) deeply and (c) quickly.* Traditional views of organizational learning indicated that firms sought to learn in cycles of exploration (March 1991), which aims to maximize learning breadth and *exploitation* (March 1991), which aims to maximize learning depth. In contrast, in hyper-competitive environments, exploitation and exploration become inseparable and interlaced. Hyper-learning is thus characterized by compressed time spans for both exploration and exploitation, and a high level of concurrency between these two. In order

to hyper-learn, organizations must find the mechanisms to close the gap between exploration and exploitation. Methods for traditional learning are likely insufficient and how these two end goals can be successfully merged in time or structure must be understood by those firms entering a period of hyper-competition.

In order to understand these new processes, the following three research questions are examined:

Research Question 1: *What are the specific mechanisms of hyper-learning in organizations that radically increase the absorptive capacity in hyper-competition?*
Research Question 2: *How can such learning mechanisms be configured as core elements of dynamic capability?*
Research Question 3*: How does hyper-learning affect organizational design and strategy execution?*

Below, in a replicated case study, the paper examines hyper-learning mechanisms that were established in seven software development firms (three in the United States and four in Finland) competing in Web development during the peak of the hyper-competitive electronic commerce boom (June to December 2000). This boom was characterized by the following: massive infusions of resources; rapid, radical and pervasive changes to the information technology base; intense competition between service providers; and unprecedented requirements for fast time-to-market delivery. The objective of this paper is to seek answers to the above questions and thus identify the characteristics of hyper-learning in these firms and how they changed their organizational processes in order to hyper-learn.

The remainder of the paper is organized as follows. Section 2 further develops the concept of hyper-competition and hyper-learning and discusses how it sets new demands for firms in developing dynamic capability. Section 3 presents the design of the field study and reports key findings concerning hyper-learning mechanisms in the seven software firms. The paper concludes by discussing the implications of the data, specifically, how the findings provide insight into the acquisition and deployment of new capabilities during hyper-competition. Further, it contrasts these findings with traditional beliefs from the organizational learning, dynamic capability, and knowledge management literatures.

Theoretical Background: Dynamic Capability and Hyper-learning

Hyper-competition and Dynamic Capability

In the literature, it has been suggested that, in stable competitive environments, companies can sustain their competitive position by engaging in extensive and prolonged exploitation processes that are only occasionally punctuated by relatively short and disruptive discontinuities in markets and technologies (March 1991, Tushman and Anderson 1986). Through the exploitation processes, firms seek to garner and refine competencies through repeated actions over extended periods of time (Dierickx and Cool 1989, Eisenhardt and Martin 2000, Nelson and Winter 1982, Nonaka and Takeuchi 1995), while through explorations they search and create new competences (Christensen 1997, Eisenhardt and Tabrizi 1995, Fredickson 1984, Henderson and Clark 1990, Imai et al. 1985, Kim 1998, March 1991, Nonaka 1994, Pisano 1994, Tushman and Anderson 1986, Winter and Szulanski 2001). In such stable environments,

chaotic and ambiguous explorations of technology and knowledge are exceptions and bricolage is unrecognized (Ciborra and Lanzara 1994).

However, recent studies on hyper-competition and dynamic capabilities challenge these views (D'aveni 1994, Eisenhardt and Martin 2000, Prahalad and Hamel 1990, Teece et al. 1997). D'Aveni (1994) defines a hyper-competition as "intense and rapid competitive moves, in which competitors must move quickly to build advantages and erode the advantages of their rivals" (p. 218). In such high-velocity environments fast change is endemic and it becomes nonlinear and less predictable with blurring boundaries and ambiguous market players (Eisenhardt and Martin 2000). Drawing from Schumpeter (1934), hyper-competition recognizes the relentless and rapid pace of innovation among rivals that creates disequilibria in the markets within compressed time frames through the use of pre-emptive strategies and rapid swings in market competition as a key characteristic of business environments (D'aveni 1994). In hyper-competitive environments, the company's strategic agenda is to increase agility: it must outpace competition by constantly exploring and pursuing new sources of competitive advantage. Therefore, in a hyper-competitive market, companies engage in a chain of small, easily duplicated strategies that maximize their likelihood of survival. Thus, in a hyper-competitive environment, dynamic capability plays a vital role in creating competitive advantage.

Dynamic capability refers to an ability that alters a company's resource base through acquisition, integration, re-combination, and the removal of organizational capabilities (Eisenhardt and Martin 2000). It enables companies to relentlessly integrate, reconfigure, gain and release resources that create and respond to swift and dramatic changes in competition (D'aveni 1994, Eisenhardt and Martin 2000, Prahalad and Hamel 1990, Teece et al. 1997). Dynamic capability thus embodies a meta-capability by which firms build up new resource configurations (Teece et al. 1997) and engage in *second-order learning*, which ischaracterized by a search for alternative routines, rules, technologies and goals (Lant and Mezias 1992). Dynamic capability often results from the realization that specific experiences and opportunities cannot be interpreted within the current theory-in-use (Argyris and Schön 1978).

Together, these concepts offer new theoretical lenses that helps us understand the importance of dynamic-capability building that underlies effective hyper-competitive strategy. A growing body of the literature has examined decision processes and organizational designs by which organizations acquire dynamic capability (Brown and Eisenhardt 1997, Eisenhardt and Martin 2000, Eisenhardt and Schoonhoven 1996, Eisenhardt and Tabrizi 1995, Galunic and Eisenhardt 1996, Galunic and Eisenhardt 2001, Henderson and Cockburn 1994, Henderson and Clark 1990, Montealegre 2002, Teece et al. 1997). They suggest a different logic for organizational design that underscores the importance of organizing through semi-structures, developing simple organizational routines and decision rules and the critical role of experimental learning and iterative strategy.

The impact of squeezing more into less time—a hallmark of hyper-competition— on firms' dynamic capability, however, has not been adequately investigated in the extant literature. Furthermore, these studies pay scant attention to strategy execution. Specifically, little is understood as to how organizations *in action* launch and configure behaviors and specific learning processes that enable the creation of dynamic capability. This invites investigation into specific organizational routines and structures as well as the consequent formulation of theories and concepts that explain how organizations quickly "learn to learn."

Dynamic Capability and Hyper-learning

There is a dearth of knowledge regarding what organizations do to dramatically expand their capability to learn. Expansion of the capability to learn requires that firms decide which knowledge assets to retain and which ones to purge. Likewise, little is understood with regard to learning under intense time pressures. While learning under these conditions is not well understood, these capabilities could well become a main competitive asset in times of hyper-competition. In order to address these gaps in the literature, we propose a notion of *hyper-learning*.

Building on the organizational learning literature (Cohen and Levinthal 1990, March 1991, Zahra and George 2002), we conceptualize hyper-learning as an organizational attempt to radically increase absorptive capacity via simultaneously pursuing the *exploration* "of new possibilities" and the *exploitation* "of old certainties" to cope (March 1991) with radically compressed time demands of hyper-competition (Figure 1). Simply put, hyper-learning requires an organization to learn *broadly, deeply, and quickly* at the same time. Absorptive capacity can be defined as the "ability of a firm to recognize the value of new external (and internal) information, assimilate it and apply it to commercial ends" (Cohen and Levinthal 1990 pp.128). Absorptive capacity is a function of an organization's prior learning and it shows path dependencies in cognitive framing (Cohen and Levinthal 1990). Thus, a shift to hyper-learning often requires deep changes in organizations' cognitive frames and belief systems, thus calling for second-order learning.

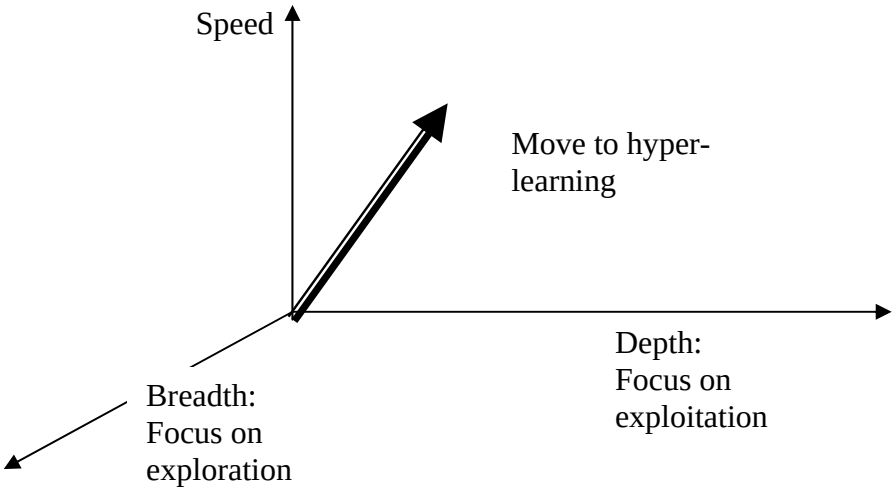

Figure 1. Pressures Towards Hyper-learning

The need to rapidly increase absorptive capacity must thereby address how much an organization is proportionally engaged in these activities and when and how they become interlinked through communications, knowledge sharing principles, incentives and culture (Cohen and Levinthal 1990). First, in order to increase the *breadth of learning*, an organization must expand quickly in a given time period relative to its resource pool. Moreover, the organization faces higher uncertainty as to what topics may be relevant for its future competitiveness so the risk of learning something that quickly become irrelevant is high.

Second, in order to increase the *depth of learning*, an organization needs to increase the speed at which it replicates its knowledge from one strategic context to another (Winter and Szulanski 2001). Moreover, the risk of failing to do so renders the organization vulnerable to higher market risks. This unique combination of fast simultaneous exploitation and exploration for hyper-learning is clarified further in Table 1 by contrasting it with more traditional forms of learning.

		Exploitation Needs	
		General and Superficial	**Specific and Deep**
Explo-ration Needs	**Stable and Narrow**	Cell I: Low-level learning in a stable environment Results in slow change and/or decay in organizational capability and change; Typical in natural monopolies and bureaucracies	Cell II: Incremental innovation through continued exploitation Results in focusing on exploitation in a stable environment to achieve or sustain dominant market position; Focus on refining rigid core competencies; Harnessing specialized capabilities in an established area, e.g. oil drilling, or software development for telecommunications industry
	Dynamic and Broad	Cell III: Exploration of disruptive business opportunities High level out-of box learning in a dynamic turn-around environment; Focus on technology and market monitoring and assessment for emerging markets to minimize business risks; Typical in consulting and the research industry	Cell IV: Hyper-learning Broad and deep knowledge acquisition and utilization with interlaced exploration and exploitation cycles in turbulent and hyper-competitive environments; Focus on fast learning as core competence; Typical in high technology industries with rapid technology changes and emerging markets

Table 1. Organizational Contingencies for Exploration and Exploitation Needs

As shown in Table 1, during hyper-learning (cell IV) the demand for *exploration* increases drastically, while at the same time organizations must radically improve their *exploitation speed* and how they shift between exploration and exploitation. Such a combination is unique to hyper-competition and makes it different from the other organizational learning contingencies that have shaped strategic thinking; in these situations, organizational learning is seen as a slow dance of organizational focus over time, or through structure, between cells II and III. In a stable environment, organizations try to prolong the period of exploitation of the dominant design of the current generation technology as their core competence for as long as possible (Leonard 1995, Prahalad and Hamel 1990), thereby focusing on cell II, until they face technological and environmental uncertainties caused by the next generation of technology (cell III) (Christensen 1997, Tushman and Anderson 1986). Therefore, stable environment exploration (e.g., R&D and new venture) and exploitation (e.g., process improvements and best practices) form distinct activities separated by time and structure. Instead, for organizations pursuing hyper-learning (cell IV), exploitation and exploration form dialectics as two opposing and separate facets of the same activity that builds up absorptive capacity over time . The core

competency is the unique configuration and interlacing of these learning capabilities that allow competition based on learning speed in terms of both depth and breadth.

There has been no prior research in the literature that specifically examines the distinctive characteristics of hyper-learning and its outcomes. In order to begin to address this gap in the literature, we conducted an exploratory multi-site case study in which we examined the following questions: (a) what are the specific mechanisms of exploitation and exploration that are used to radically increase the absorptive capacity (question 1); (b) how are these learning mechanisms configured (question 2); and (c) how are the outcomes of hyper-learning reflected in their organizational design and strategy execution? By answering these questions, we seek to build a theory of hyper-learning.

Research Methods

We chose to conduct a multi-site case study (Yin 1994) to address these research questions. The value of this type of study is that it allows for a replication logic by which we can test emerging theoretical insights that permits us to triangulate both theory and data during the research process (Eisenhardt 1989, Strauss and Corbin 1990). We conducted a 7-month field study in seven software development firms that develop Web-based systems using emerging Internet computing (March et al 2001). Here, "Web-based" systems refer to computing applications that utilize Internet browsers, such as Netscape or Microsoft Internet Explorer, and a set of open standards and protocols that include XML, HTML, http, URL, TCP/IP, combined with the extensive use of middleware[1] architectures in leveraging the computing service.

We identified seven companies that met the following criteria: (1) they were developing Web-based systems; (2) they were recognized by their peers as users of the most advanced technologies available; and (3) they worked mostly for outside clients through contractual relationships. Web-based systems development formed a key strategic technology direction of these firms and they were active members of a technology community experimenting with Web-based technologies. They were constantly seeking to manage the uncertainty of their leading-edge technological solutions.

As the companies were developing software for external customers, they had to also simultaneously assess, manage and share business and technology risks with their customers. As a result, these companies were struggling with the challenges of identifying and managing the knowledge assets of how to work with the "bleeding edge" technologies and demonstrate their value in a business environment. We believe that this environment can be characterized as hyper-competition. A summary of the firms' characteristics is included in Table 2. We explore in detail the various learning mechanisms that these seven firms used to acquire, integrate, reconfigure and purge knowledge assets at an unprecedented speed in order to cope with hyper-competition.

To minimize potential bias in our study, we sought to maximize the variations in our sample. This also served the goal of improving the external validity of the study using the replication logic by which the studied phenomena can be generalized into a theory of organizational hyper-learning (Yin 1994). Companies had different sizes and operated in different industry sectors in terms of the services provided (ranging from manufacturing,

[1] In the computer industry, middleware is a general term for any programming that serves to "glue together" two separate and usually already existing programs.

financial services and public administration to retail and transportation). They had experience using Web-based technologies in several application domains (back office, front office, and inter-organizational applications). The geographical scope of their operations varied largely as some were local software firms while others were part of large global companies. The firms also had large variations in their software development experience, ranging from as few as four years to 40+ years. We also included three firms from the U. S. and four from Finland in order to minimize cultural or regional bias. Both countries are known to be leading exploiters of Internet technologies.

The data was gathered primarily through semi-structured interviews with senior management and by analyzing archives of the company documents, including systems development documentations and technology strategies. We approached the companies and requested access to their documents and senior executives and developers who made decisions about technology investments and were in charge of strategy. The interviewees thus managed the organizational knowledge bases and skills needed to execute a chosen technology and business strategy. A range of three to six individuals from each company participated in the study. Each interview lasted from one to three hours. The interviews reviewed the applications and solution portfolios provided by these companies to their clients and examined how they delivered these applications. We probed for changes taking place in the business and the technology domains of their operations as a result of Internet computing. Specifically, we asked the firms to clarify the extent, scope, depth and speed of changes in their software development when compared with their situation prior to the Web development and to focus particularly on how their mechanisms for learning had changed. We further examined how these firms coped with these changes and how they had created dynamic capabilities.

All interviews took place between June and December 2000 and were tape recorded with the permission of the firms. The interviews were transcribed and the summaries of these transcripts were sent to the companies for correction and validation. Additional notes were made during the visits concerning physical sites, personnel age, general atmosphere etc. We also collected written materials from these companies, including their annual reports, Web sites, advertising materials, manuals, and system handbooks. A summary of firms and their characteristics is represented in Table 2.

Data analysis was done using the traditional inductive method (Boyatzis 1998, Glasser and Strauss 1967, Strauss and Corbin 1990). The transcript of each company was subject to a within-case analysis that involved repeatedly reading the transcript and taking thorough notes about firms' perceptions of the competitive environments and their reactions to the environmental changes and the need to create dynamic capabilities. After each individual case had been analyzed, we began cross-case comparisons that involved listing the similarities and differences among the firms we studied. The results of the cross-case comparisons provide a vivid picture of the hyper-competitive environments that these firms were facing along with the learning challenges. Furthermore, the results provide a consistent picture of how these firms developed various learning mechanisms in order to simultaneously speed up exploitation and exploration and how they configured them to radically increase their absorptive capacity. The main findings are summarized in Table 3.

Hyper-learning in Web-based Development Organizations

Web-based System Developments as a Hyper-Competitive Environment

The emergence of thin browsers such Mosaic, Netscape, or Internet Explorer, combined with the introduction of Hyper Text Mark-up Language (HTML) and Hypertext Transfer Protocol (http)— a simple request-response protocol taking advantage of an open network— fueled the development and diffusion of the Internet beyond its original purpose (Lyytinen et al. 1998, Tuomi 2002). While the initial implementation of the Internet dealt with the relatively narrow technical challenges of packet switching, the success of this architecture in an open network later on spiraled a wave of unforeseen computing solutions, tools and methods known as Web-based computing. Consequently, the second half of 1990s witnessed wide and rapid developments in all types of new tools, standards, protocols, and platforms for Web-based computing. The innovations include standards for rendering multimedia (e.g., JPEG, MPEG, Real Audio, Quick Time, etc), high-level protocols for data exchange, transfer and storage (e.g., XML, SSL, HP's e-speak, EbXML, or Microsoft's E-Biz etc.), scalable and reliable web servers (e.g., Apache, Microsoft IIL, etc), new forms of middleware (e.g. BEA WebLogic, IBM Websphere, Microsoft's IIS), application servers (e.g., BEAs Weblogic, IBM Websphere, Oracle's PortaltoGo, Microsofts .NET), and component-based software (e.g., .Net, Java Beans, J2EE, etc). These innovations were triggered by and subsequently triggered new business opportunities that had not previously existed (Basu and Kumar 2002, Sambamurthy and Zmud 2000, Straub and Watson 2001). New companies with radically innovative business models, including eBay, Amazon.com, Priceline.com, and DoubleClick, emerged, an outcome that necessitated high-level Web-based computing capacity. At the same time, traditional companies embraced Web-based computing as an integral part of their on-going business activities.

Such developments created an unforeseen demand and competitive pressure on the studied firms. Interview data made it clear that software development in these firms differed greatly from the prior client/server and mainframe-centric computing. As one senior developer said, *"I know that the way that we build and integrate software is much different in this space than what it used to be."* Specifically, these firms characterized development in the new environment as having *compressed technology lifecycles* that resulted from intense and diverse innovation in Internet computing, as indicated in new programming languages (Java, C#, J2EE), development tools (.NET, Coldfusion, IIS etc), and data base systems (BEA WebLogic, Oracle PortalToGo, IBM Websphere). They faced *increased* and *different* customer demand as observed in completely separate applications (e.g. portals, front-stores, mobile applications) and in the compressed time horizon in which they had to deliver them (from several months to several weeks). At the same time, they observed *unprecedented levels of intense competition* created by lavish access to venture capital, which responded to new market opportunities (as evidenced by the emergence of new companies and competitors in the development market). Finally, they had to develop *new competencies and business models* (i.e., content editing, communication skills, partnerships, cost/profit sharing practices), which had traditionally not been part of their business logic.

All firms felt that their ability to monitor, identify and assimilate new technologies was critical to their survival in the business. These technologies dramatically shortened the life cycle of a large portion of the skills and expertise they possessed. As a result, these firms needed constantly explore and assimilate new technologies in order to stay ahead of the competition. Such constant explorations resulted in an expansion of the breadth of capabilities necessary to

compete successfully. Firms did not only need to acquire more knowledge, but new types of knowledge that could not easily be understood within their cognitive frames.

In the past, firms had focused primarily on honing their skills in managing software development processes as well as their associated technical competencies in capturing software requirements and designing and implementing the system. Therefore, they all followed a structured problem-solving approach in their software development activity. This was documented in a set of roles and guidelines that stated what tasks had to be carried out, in which sequence, with what outcomes, and by whom. Such guidelines established the backbone of expected organizational responses to market competition. They stated how the organization would execute its software delivery task and with what technological means in the market place. Many times the guidelines had been audited and formally documented through such certification processes as ISO 9000 (Peach 2002) or CMM Software capability model (Humphrey 1989) in order to indicate the predictability and repetitiveness of an organization's responses to their clients.[2] These methodologies were detailed and implied a high level of detailed routinization through the reliance upon stable and well-tried technological platforms. In short, these organizations had been well equipped to effectively exploit their certainties in the past pertaining to the characteristics of good software development (see also Pressman et al 1998).

As noted in Lyytinen et al (1998), however, Web-computing embraced a radically new set of technological skills and other competencies that questioned many earlier capabilities and views the companies had valued. Many companies were thus taken by surprise by the scope and speed of change in these competencies. They had to rapidly expand their exploration of new uncertainties, which included such diverse areas as telecommunications and component based software development, (multi)-media oriented graphics, content and audio-visual representation, process development practices that improved customers' business processes around web-based service delivery, and change management that initiated organizational change in their clients. This rapid change of skill sets is echoed in the following quote:

> "[For Web-based systems] it's typically the [specialized] front-end developers who will do the design. Or there will be a specific design person. If there's a specific design person, that person might know HTML, but it's unlikely to know JAVA. If you have a developer, that person's likely to know JAVA, but not have the best design skills. And, obviously there is a disconnectivity between those two because you want your web page to look good and run fast. And, it's hard to find resources that can do both. It's not impossible, it's just hard. Again, because developers tend to be more logical. They are not focused on the artist approach to what a website should look like. They need usually guidance."

A significantly increased breadth of knowledge also brought to the fore the criticality of meta-knowledge - how to reflect upon and organize these competencies within an organization, how to build learning processes that provide solutions to burning questions, what to acquire, what to keep, and what to throw away. Such meta-knowledge was related to the strategic thrusts and drivers of these firms - in what technologies and markets would they be in and need to compete. This change is reflected in the following quote by one business manager:

[2] Three out of seven companies we studied had obtained such certificates in the past and they were regarded as important signs of maturity and reliability in developing software solutions.

> "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."

According to another manager:

> " (I) don't have enough people yet to say that we have very good expertise on every technology we employ."

While these firms were rapidly expanding the breadth of capabilities, they simultaneously had to exploit the newly acquired skills. To assimilate these knowledge assets faster, they needed to expedite the experimental processes. For example, in firm 1, the team-leader would recognize when the team was deficient in a new skill and would task his subordinates with learning the particular skill through their own research and study and then implement the new knowledge as soon as possible.

> "Well I mean, it's more of a mentor thing, where [my subordinates say], 'well you know I'm getting blown up on this,' and I'll go and look at what they're doing and [say] you really need to go read up on this and here's a book...[and these reading assignments are the equivalent of] tutoring on demand."

Furthermore, the firms had to apply the assets in a wider scope in order to better understand technology potential. This occurred in part by learning from others and in part by learning from personal experience. The firms also had to internally transfer the knowledge faster so that problems amenable to Web computing could be formulated and solved effectively. The rapid expansion of depth helped them to see what was truly novel in technologies and what could be reused from their prior competency bases.

Similarly, according to one of the managers at Firm 7, they saw themselves as having two types of projects - one for exploiting the technology skills they had managed to disseminate throughout their development group and another, which was risky, but allowed for the fast exploration of new skill sets while simultaneously exploiting them.

> "[We] have like two types of projects. Those that [utilize our well understood technologies, and those that] introduce new technologies in some kinds areas. And those latter ones are highly exposed [i.e., risky], and for those you use your [specialized] skilled resources. [That group] is the type of talent who are very good people in these technologies and you allocate these people to those highly exposed projects"

In addition to the rapid expansion of the breadth and depth of knowledge, the companies experienced a rapid contraction of time horizon. This was a result of two factors. First, the fast developments in technology placed pressure on these companies to accelerate their learning processes. This primarily affected the speed and scope of their exploration. Second, their clients demanded a radically faster delivery of solutions, which deeply affected their exploitation speeds and capabilities. Companies sensed a great urgency among their clients to "get the damn system out" and to become the first to market. The applications had to be turned out at "Internet speed" and firms had to set up dramatically shortened project targets. This happened at the same time when the amount of knowledge that needed to be assimilated increased. In fact, firms' efforts to expand the breadth of capabilities were caused in part by the dramatic compression of development timeframes and the shortened life expectancy of technical knowledge. Firms were

less sure how long they would require specific capabilities and how critical they would be in the future. In order to deal with these uncertainties, the firms had to become more selective and conscious about what to keep and what to give up and quickly comprehend the potential value of a new capability before extensive exploitation occurred.

Collectively, our data reveals that rapid technological change, increased pressure from clients and high uncertainty and ambiguity of markets resulted in hyper-learning. Firms had to continuously juggle the processes of exploration and exploitation in order to stay at the forefront of the wave of innovation.

Rapid Expansion of the Breadth of Knowledge

How do companies rapidly acquire new capabilities? Our data reveal a remarkably consistent pattern by which the firms we studied rapidly acquired new capabilities. While each firm had a slight variation in its capacity building, all firms followed the mechanisms of (a) distributed and active gate-keeping and (b) extended grafting of external knowledge in order to rapidly acquire new capabilities.

Distributed and Active Gate-Keeping. In the new situation, companies closely heeded the changing environment and allocated a large proportion of their resources to boundary spanning and environmental monitoring activities. One prevalent mechanism was the active use of gate-keepers that scanned their environment and translated technical and business information into a form that was comprehensible to the rest of the organization and its agenda (Cohen and Levinthal 1990). Such a gate-keeping function took several forms, but all of the firms moved this activity in the front line where the strategy was executed. It was also distributed and decentralized across a larger number (up to 80% of the organizations' members) of organizations' members than the traditional form of gate-keeping, which typically is limited to a small number of people in the organization.

For example, the strategy of Firm 1 was primarily driven by the capabilities of lead developers who were attracted to the company's technology jobs. Therefore, the firm sold only what the developers wanted to produce and exploit, based on their exploration. For example, one executive said:

> "We use J2EE because it is considered to be the attractive technology by the elites in the development community. Those are the type of people we like to hire. So, if we tried to give less attractive technology jobs to those people, the developers would quit… We hire good people and good people want good technologies. And, we go with the technologies we go with because that is where people want to go."

By having at least one person in the organization keep up with each of the newest technologies, the organization was able to place this person in the role of lead mentor in the development group to disseminate knowledge quickly once the technology needed to be exploited:

> "Typically, if you go into a new project and there's new technology, [we make sure to already have] a few people who have [familiarized themselves with that technology] before. [That way we] won't have everybody who's not done it. The people who have done it would review the code or the work products of people who haven't and then show them what they're doing wrong."

The firm invested heavily in off-site training for these developers in order to enable them to acquire new skill sets. Once they acquired these skills, they became the main drivers of the firm's future services.

Firm 2, a large global management and technology consulting firm, relied heavily on external sources such as open-source communities and Internet newsgroups as means of acquiring new knowledge. An executive of the firm observed that, "*geeks in newsgroups have the best information and we try to piggyback on them.*" The firm encouraged all employees to participate in the newsgroups, but in most cases this activity relied on a cadre of "technology gurus" who were deeply enmeshed in the fabric of those newsgroups and could explore and understand new technological solutions. The reason for this was the depth and fast pace of change in knowledge and its contextual nature, which required high level expertise and social networks to truly grasp it.

Firm 3 had been spun off from its parent company, which focused on "traditional" mainframe based services. It was spun off "*because of the recognition that decisions could not be made fast enough in the parent corporation moving at the large company pace to meet the requirements of the Internet pace.*" It originally started with 22 people, but quickly tripled in size. Since the core members of the firm came from the distributed network group, the firm had to quickly acquire new skills for the Web-based developments outlined above (media skills, business change skills, and component development based skills). To fill this gap, the firm hired a new chief information officer (CIO), who had substantial experience in Web-based development, to infuse these developments skills to the rest of the firm and accordingly establish the gate-keeping capability.

Firm 4 and Firm 5— both start-ups— focused exclusively on Web systems development and had similar structures due to a lack of organizational inertia and earlier cognitive framing. These were optimally matched with their local learning needs and strategy execution. Both of them identified a small group of "experts" in specified areas and relied heavily on them to identify and assimilate new capabilities. Firm 4 had "Skill Leaders," in eight different areas who were responsible for "*scanning environment of external technologists, the press, and newsgroups.*" Similarly, Firm 5 maintained a list of employees who were identified as "experts." They amounted to about 10% of the employees. In addition, the Human Resource department kept the list of employees who were interested in learning new technologies. About 25-30% of employees fell into this group. The firm sent those people for training and tried to elevate them into experts. Sometimes, they also recruited people to bring new technological capabilities into the company. For each key technology, the firm expected to have at least one to three experts. These experts had to share what they had learned from scanning the external environment and through the technology trials with the rest of the firm. Within the company, everyone knew who those experts were, and sought their help in addressing their technical problems.

Firm 6, a consulting branch of a global computer hardware, software, and service provider used "boundary scanners," who worked closely with the firm's business partners. Each of them was an expert in a set of selected technology areas. Typically, the firm negotiated with their clients to run experiments in emerging technologies, such as mobile commerce and wireless services. These projects involved a higher degree of risk and the firm sometimes agreed to take on the project with lower profit margins. Since the firm's employees did not have all the necessary capabilities and skills, it sought out the expertise of its business partners. The business partners kept the firm informed of what they had learned in these projects so that the firm could offer proven solutions later to its clients through their expert network. The firm also utilized a

large global knowledge management system that was dedicated to collecting data on solutions that had been offered by boundary scanning business partners.

Finally, Firm 7, a large Finnish information technology service firm, ran a separate technical support group for its frontline consultants. At the highest level of technical support, the firm had "mentors" who comprised about 1% of all employees. These mentors were experts in chosen technical areas. They were assigned to key pilot projects when a new technology or strategy was being developed and deployed for the first time. Typically, a mentor stayed with the project about three months before he or she moved on to the next project. Mentors were also expected to bring frameworks, methods, and guidelines to the rest of the organization from their fields of expertise.

In summary, all firms had active and explicit structures and processes of gate-keeping for identifying, monitoring and acquiring new capabilities. These individuals were highly regarded in the firms for their technical skills and domain expertise and were actively recognized and supported by senior management. Many times these gate-keepers were deeply immersed in the Internet communities, hardware and software vendor communities, and academic research communities where they constantly sought out new opportunities and threats. Many firms strategically selected their projects in order to acquire new capabilities and placed gate-keepers in those projects. Thus, active, distributed and decentralized gate-keeping seem to be a unique feature of organizational hyper-learning; the gate-keepers provide strategic access points in the "front line" to critical knowledge and competencies and thereby become the nexus of expanding the breadth of competencies and speeding up their exploitation.

Extended Grafting of External Knowledge. As a result of diminishing timelines and the introduction of several new technologies, the firms had to import a significant amount of knowledge from external sources by directly acquiring software components, through outsourcing or strategic partnerships. Extended grafting resulted in the rapid acquisition of capabilities in the form of tangible resources or codified knowledge that could be easily disposed of when no longer needed. This strategy was followed primarily when there was no time to develop competencies internally, when these competencies did not fit with the strategic management "vision," or when the external sourcing exposed them to a low power position due to resource dependencies. Sometimes extended grafting was driven by the need to keep an organization's identity and its cognitive models simple and uniform. For example, older software companies resisted the idea of hiring graphics designers and instead outsourced graphical designs. Grafting was a radical change for these firms and was adopted out of necessity to deal with the demands of Internet computing. As one manager indicated, *"moving towards assembly, it just happened overnight."*

Because of these changes in building applications, firm 1 found that they now sought to purchase software components from external sources whenever they were available. At the same time, it no longer actively built its internal software libraries. This was a significant change from the past because the firm used to develop most of the software internally and draw upon its internal software assets. Although the increased technological capabilities of component based software[3] certainly contributed to this change, the compressed development time was an

[3] This is echoed in the idea of being able to "plug in" software components like Lego blocks into different parts of the software "architecture" by concealing the differences in the runtime platforms and interfaces through middleware "glue" like Object Request Brokers (Corba), or Distributed objects (DCOMM), see (http://www.omg.org/news/whitepapers/).

important contributing factor for this change. An executive of Firm 1 stated the following:

> "We always take the fast way out. 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 quality assurance."

Further, he believed that they were going to have to drastically increase component buying from the currently high levels as the way to deal with increasingly shorter timelines. As he indicated, *"You know we have to move in that direction in order to be able to respond faster."*

Firm 2 imported external knowledge in the form of software components as well. It relied heavily on open-source communities that were monitored through their gate-keepers to check the new entries or changes in the community directories. The firm also outsourced software development to external contractors to bring in "developers" whose skill sets the firm did not internally possess. Firm 3 also experienced a significant increase in outsourcing. It extensively used external Java programmers after it hired its CIO to acquire the skills utilized in Web-based systems development.

Firm 4 actively utilized both outsourcing and components as a way to expand the breadth of its knowledge. One executive of the firm said the following:

> "Customers often have many parts of the project that we don't give to them. We outsource [web site] hosting and ASP [application service provider]. We purchase the platform and buy components. We purchase components because we don't have enough resources. We don't "do Web sites" and we will act as project manager for the existing clients to find a small company to do it."

This firm chose Finland as its location as it enabled outsourcing for the following reasons: (a) the firm was rapidly expanding its knowledge resources in the mobile computing area (b) Finland was a convenient area for recruiting people with mobile computing knowledge; and (c) Finland had a deep pool of local resources.

Firm 5 was least active in grafting external knowledge. Still, its executives admitted that they tried to buy components when good ones were available. Along the same vein, the firm used outsourcing infrequently. Sometimes, when a customer wanted them to develop systems with the newest technologies, the firm some worked with external contractors to bring in such technologies.

Firm 6 actively used outsourcing when clients required special customized developments. In particular, the firm extensively outsourced graphic and telecommunication designs - especially in its growing mobile business. The firm also extensively purchased components when required. Finally, Firm 7 outsourced the areas in which they neither had strong capabilities nor saw any reason to expand them. Those areas included testing, graphics design, and training. In addition, the firm sought to acquire other companies that possessed key strategic skills that would be useful in their future businesses, including mobile services, enterprise integration platforms and so on.

In summary, except for Firm 5, all other firms used outsourcing and strategic alliances to rapidly acquire new capabilities. Many firms purchased software components when they did not have time, or when they did not see the need to develop deep expertise in that area. On the whole, firms struggled to find a balanced and "strategic" way to graft external knowledge. Outsourcing, strategic alliances, and component purchasing were pursued mostly to gain access to capabilities where their expected lifespan was seen as short and insignificant to their core

strategy.

Ironically, at the same time, the firms relied less on internal documentation and the reuse of their software assets. This was due to the rapid expansion of technology domains, such as multimedia and communications. Another factor was the fast rate of diffusion of Web solutions to different industries. As a result, companies worked with many disparate industries, which reduced the portability of functional code. Some of their clients also required that the delivered code should not be reused. The rapid exploration cycles caused also constant information overload making the reuse of code less effective. For example, Firm 5 had set up internal libraries for software reuse, but these were not effectively used. There were too many and diverse software components stored, which over time made it confusing, uncertain and dangerous to use the components.

Rapid Expansion of the Depth of Knowledge

How do companies in hyper-competitive environments exploit the ambiguous and uncertain knowledge that they have acquired and transform it into a certainty? How do they deepen the knowledge and yet manage to keep abreast of rapidly changing environment? Our data provides several insights into how companies accomplished this without losing their ability to further expand their knowledge bases. Again, we observed the following two mechanisms that increased rapid exploitation: (a) the reuse of simple rules and artifacts, and (b) the building of peer networks.

Reuse of Simple Rules and Design Artifacts. As noted earlier, firms relied extensively on external knowledge sources instead of reusing of their internal code assets in order to free cognitive capabilities to acquire new capabilities. At the same time, to enable the rapid exploitation of their codified knowledge, they pursued a new sort of artifact-based exploitation. Contrary to the past when the reuse of the code was internally developed, the companies were increasingly reusing higher-level system knowledge that pertained to the development processes and design patterns – a reuse which yields higher productivity gains, but requires a higher abstraction of task domains (Zhang and Lyytinen 2001).

Firm 1 retained the effective development patterns from their previous projects. These covered both the processes and design patterns that offered boilerplates for solutions. In particular, they reused templates for web pages they had developed. This resulted in a methodology that was dramatically simpler and favored the use of prototypical design solutions. The reason for this change was that the older detailed methodology did not work. One of the interviewees said the following:

> "I did out of sheer 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" (Architect, Firm 1).

In Firm 2, patterns and processes were also highly reused. Design patterns were made available in manuals that were shared over the Internet. Many of the analysis patterns were developed in the headquarters and the emphasis was on the significant reuse of patterns. This was a significant departure from the firm's past practice that emphasized code reuse. Furthermore, the executive of the firm said that accelerated changes in technology forced them to loosen up their rigid development methodology, as detailed guidelines did not work. Again, this was a significant departure from their past practice where the firm was known to take huge pride

in having a well-established development methodology.

At Firm 3, the goal was to simplify their methodology and reuse mostly functional business logics behind the system design. According to one executive,

> "These products [web-based systems] were developed without a formalized set of methodologies. Current versions of software did not develop at the Internet speed. Some took 18 to 24 months, which we don't consider acceptable. We have been "winging it" using some very basic methodology rules and tools to make a quality product."

Although the firm had a well formulated system development methodology it had inherited from the parent company, the developers argued that it did not fit with the new environment. Consequently, the firm had adopted a simple and informal 3-step method. Similarly, Firm 4 had developed a simple 3-step methodology that consisted of the "envision," "shape," and "realize" phases. The methodology enabled the firm to reuse the development patterns without reusing the code over a variety of technological platforms and business environments.

Developers at Firm 5 mentioned that they did not reuse the code, but reused their simple methodology. Their methodology was also simpler than the traditional software engineering methodologies to which they were accustomed. They believed that traditional methodologies would be too "heavy" for the size and speed of typical Web development. Firm 6 was similar to Firm 2 in terms of its global scale of operations as well as its heavy reliance on an established methodology. Like other firms we studied, Firm 6 had adopted an informal methodology for its Web systems, which reused analysis patterns. While they reused their analysis and overall solution designs, they integrated them with outsourcing and external component purchasing practices during system implementation.

Finally, among the firms we studied, Firm 7 most aggressively pursued code reuse and highly standardized development processes. However, an executive acknowledged that following their strict development processes was difficult and the firm had to focus on "bare essentials." As a result, it was stripping off less essential aspects of the methodology.

Overall, it is striking to note that all firms were moving away from rigid, detailed and standardized development methodologies that had been the pride of these companies over the last ten years. At the same time, they used design artifacts and simple patterns as a means of exploiting their knowledge assets. This resulted in radically simplified and informal development methodologies, which represented a sharp departure from the prevailing tradition within the software industry of reusing code and adhering to detailed (routinized) methodologies in order to successfully manage software development. Moreover, some firms had extensively used CASE (computer aided software engineering) tools to automate steps of their formalized development process. Not surprisingly, those tools were not used any more. This "un-bundling" was necessary since their methodologies did not fit with the changing environments where speed and change was everything. Most technical experts and executives attributed the shift to simple artifact-based reuse and patterns to rapidly expanding business and technological uncertainty. The new technological landscape made software reuse less appealing: software vendors introduced tools that were not compatible with their existing code. Rapidly expanding business environments made software reuse less attractive, since client requirements varied dramatically across different industries. Furthermore, the shortened development times placed a tremendous amount of pressure on the timely completion of projects whereby the reusability of software code became a less important goal.

Building Peer Expert Networks. The firms relied on peer networks, in order to effectively exploit their low-level detailed technical knowledge necessary to implementing the systems. All firms used mentoring mechanisms and peer networks to share and maintain technical knowledge. In most cases, gatekeepers became the critical nexus of mentoring processes and thus acted as the key integration mechanism in interlacing the exploration with exploitation. For example, in Firm 1, the top management tried to *"spread the knowledge around by placing the people with appropriate skills sets to mentor available people who are assigned to a project."* They also reviewed the software code written by others in order to share new knowledge and transfer good coding practices.

Firm 2 used to rely extensively on formal training. Traditionally they had relied on an elaborate system development methodology in which all new employees received "indoctrination" training that lasted for three weeks. Additional CDs, technical manuals, technical databases, and satellite video facilitated knowledge sharing around the methodology and designs. However, the firm soon realized the shortcomings of their rigid knowledge transfer mechanisms. Accordingly, it started informal on-the-job mentoring in relation to Web system development and relied on gatekeepers to provide technical expertise in their specific areas. According to an executive of the firm, about 75% of people ask the gatekeepers for existing solutions during design tasks in order to minimize duplicate efforts. Most often, these gatekeepers were contacted via e-mail.

At the time of the study, Firm 3 was planning to implement a mentoring program under the leadership of a new CIO. The firm planned explicitly to tie their employees' promotions and salary increases to their mentoring activities. Firm 4 actively used mentoring as a means of sharing knowledge. The firm often hired people without adequate technical background as long as they had a minimum amount of technical knowledge and were willing to learn on their own. Most training was done during the projects through mentoring and each new employee was assigned to a mentor. (S)he also had access to regional training programs where they met other people and could exchange technical and tool knowledge and solution guidelines.

Firm 5 used "unofficial" experts to share and deepen its knowledge base during exploitation while simultaneously promoting experts to exploit acquired technical knowledge at a detailed level. The projects of the firm had a "flat" structure, which was expected to encourage people to take ownership of their own areas. As noted earlier, the firm expanded its knowledge base mostly by relying on experts who had to deepen their knowledge voluntarily and through new hired employees. In either case, the employees who had expertise were expected to share their knowledge with others through mentoring.

Firm 6 had a firm-wide training program for new hires. It began with a conference that was designed to educate people before they joined the team. Each new-hire was then brought under the tutelage of "on-boarding personnel development managers" who acted as mentors for three months. After this period, new team members were released into the real team environment to work on real projects. Once they were assigned to these projects, the firm assigned a local mentor to each new-hire for 2-4 years. Each mentor, in turn, had 5-10 local apprentices. Apprentice development became part of the evaluation of the mentors' performance.

Finally, Firm 7 also actively used a mentoring mechanism for sharing knowledge. Once the exploration group understood a technology or skill set, this firm would then transfer its knowledge into a knowledge base. This was a codified, three-stage process where a select group of "mentors" or knowledge explorers (there were approximately ten in a development group of

1000) first gather and then transfer knowledge to a select group of support personnel (approximately twenty people). The support personnel would then take this transferred knowledge, enter it into a knowledge base and act as support staff for the general group at large (the remaining 950 people):

> "Mentors guide the support groups, and the support group, which can provide the FAQ [frequently asked questions] acts as an answering service and [provides] help for the [subsequent] projects"

These mentors were put in a project long enough to transfer their expertise to one key person in that project. These people later became mentors themselves after obtaining adequate technical depth. According to one executive of the firm,

> "Projects where mentors were assigned were typically those where some new technology or strategy was being developed and deployed for the first time in the company. Once the lessons were learned from the pilot project, the knowledge was transferred to the rest of the technical support organization from the mentors so that future groups working in this area would not need mentors."

In summary, our data reveal how firms were increasingly moving away from traditional, formal, and extensive training methods that relied on codified knowledge and standardized evaluation methods. This became particularly evident among established and large companies like Firms 2, 6, and 7. These firms had used these formal training programs to "indoctrinate" their employees into their routine software development approach. When the companies recognized their limitations, they started to increasingly rely on simple rules and design artifacts that acted as "boundary objects" for development teams and began to offer flexible and efficient structuring and solution mechanisms. These "boilerplates" could later be filled with increasingly varying and complex technical detail through mentoring and expert based reviews. When determining how to integrate and embed these highly complex and volatile technologies into evolving designs, they thus established social networks of mentors and apprentices. Many times, peer networks were integrated with organizational memory approaches like Q&A or FAQ databases where people could search for and discover existing mentoring knowledge that was partly codified through conversations or shared opinions in the network. Such knowledge often expanded to Internet news groups or to external peer networks in the community.

These organic knowledge-sharing networks offered several advantages. First, new capabilities could be rapidly shared with selected small groups. Unlike traditional training programs, in which it takes months to prepare the training materials, mentors could quickly share what they know with groups of people when the technical context was adequately established. This knowledge could also be audited and modified through the use of collaborative and distributed technologies. When the companies shifted toward the use of increasingly simple rules and artifacts as coordinating mechanisms for development, peer networks offered a complementary mechanism to share and maintain detailed technical knowledge that had been bundled together with the routines (methodologies) in the past. Mentoring processes were thus critical and necessary to rapidly sharing uncertain and detailed technical knowledge. As a result, the detailed technical knowledge became unbundled from the methodical routines of the organization and instead became the property of the socio-technical networks and the smart people engaged in them.

The second advantage of mentoring was its flexibility. Unlike traditional software engineering knowledge transfer mechanisms where everything is embedded in a rigid

methodology or a CASE tool in order to ensure the consistency and predictability of the development processes, peer networks provided increased flexibility as to when, how and to what extent the knowledge was shared. This was valuable when organizations faced a high uncertainty of how lasting the knowledge was. Through peer networks, gatekeepers could quickly and adequately learn new technologies and share their knowledge with others.

Finally, to pursue fast exploitations through the peer expert network, these firms sought to hire smart people and create the "aura" of an intelligent organization through hiring policies and incentives. In many cases, it was more important to find a person who was excellent at acquiring new knowledge than it was to find someone with a specific skill set. As a result, some of the most valued technologists in firm 5 came from a variety of backgrounds. The manager noted the following:

> "Probably something more about the experts is that I think that we have more diversity in the backgrounds of our people…we have one biologist, one teacher, actually two teachers, one librarian, we used to have one journalist but he left, so they are not all just, you know, coders."

Also, an executive at Firm 1 commented that the firm intentionally hired people who could thrive in this environment:

> "We hire really good people… we believe that for something like language they can pick it up themselves and get some mentoring. Now, if there was something really drastic, then we will probably send them for training. Most cases they can pick it up themselves. We can pick up things fairly easily."

Incentives for these experts included making their job interesting, not simply offering them more money. As the owner of firm 5 said, "*Money is not the issue. I know from the experience that we've had, that's not everything.*" Specifically, his employees had turned down more money elsewhere because, "*the stuff they're doing is so boring, [they were] not interested.*" In addition, the gifted and valuable learners or "experts" were allowed to choose their own career path in an effort to keep them interested in staying with the company: *"[The positions they end up with] depends on what they wish for. We try to take their wishes into account because all we have is what they have between their ears, actually."*

Through smart people, the grafted components become a key ingredient of hyper-learning. "Smart" people picked up these components and "filled" them into simplified development patterns. These simple design patterns that acted as cognitive platforms for agile development and rapid sense-making had to be complemented both with external codified knowledge and rapidly configured local knowledge through peer networks, both of which depend on smart activities. Therefore companies tried to graft large amounts of knowledge from external sources, at some points exceeding the knowledge of the experts in most other firms and the world community. Some of these gatekeepers found that they were among the first in the environment to be searching for answers to specific questions. As the gatekeeper in firm 4 said, *"I have found usually nobody [in his external network including Internet bulletin boards and throughout the international 10,000 employee network] knows answers to my questions, so that makes it pretty hard to ask those questions. So I learn to find the [knowledge] on my own."*

Discussion

During hyper-competition, organizations gain, integrate, reconfigure, and release capabilities to cope with unpredictable and swift changes in competition. This is accomplished by transforming their learning activities in a manner that is consistent with hyper-learning. We began this paper with three research questions. Our exploratory study provides answers to these questions. Our first research question deals with the specific learning mechanisms that organizations use for exploitation and exploration in order to rapidly increase the absorptive capacity. Our second research questions addresses how organizations configured these four learning mechanisms as an element of their dynamic capabilities. Finally, our third research question is to identify the outcomes of hyper-learning as reflected in organizational design and strategy executions. Below we will discuss our findings in light of these three questions.

Questions 1: Learning Mechanisms for Hyper-Learning

What were the specific learning mechanisms by which organizations rapidly increased their absorptive capacity? We identified two specific mechanisms— *active and strategic gate-keeping* and *extended grafting of external knowledge*— that were instrumental in fostering rapid exploration. Likewise, we identified the *reuse of simple rules and design artifacts* and the *building of peer networks* as two primary mechanisms that quickened exploitation. In both areas, organizations invented behaviors and processes that significantly departed from their established responses to learning needs.

Question 2: Configuration of Hyper-Learning Mechanisms

All firms *simultaneously* engaged in exploration and exploitation. Therefore, they moved their technological and business expertise to the front line through the decentralized and distributed organization of gatekeepers. This increased their demand for people who could think out of the box–those they called "smart." At the same time, these gatekeepers became the nexus of fast exploitation through their position in peer networks. Another mechanism that enabled fast exploration was extensive grafting through codified external knowledge and skill sourcing. This freed additional cognitive resources for identifying and monitoring rapid movements in the environment. A shift towards using simple rules and artifact-based exploitation enabled vicarious learning through trials and the consecutive refinement of cognitive frames that enabled fast exploitation. When combined with informal knowledge transfer in peer networks, this resulted in the more frictionless and agile mobilization of cognitive resources.

The simultaneous engagement in both facets of learning had temporal and structural implications. Temporally, firms did not follow the 'punctuated equilibrium' model of learning (Tushman and Anderson 1986). Instead, they had to explore new technologies and at the same time find ways of applying them on the fly. In line with this, firms did not create structurally separate units dedicated to either exploration or exploitation, but rather expanded their competencies in the front line by making it smarter to explore the hyper-competitive environment and move competencies around. In most of the studied cases, nearly the entire organization was dedicated to both exploration and exploitation.

Another important characteristic of hyper-learning mechanisms was the reciprocal—or systemic—interlacing of exploration and exploitation. Hyper-learning organizations not only evenly pursue exploitation and exploration, but had to deeply embed these in their social fabric. Exploration and exploitation therefore become dialectical moments in the rapid movement of technological and business innovation where gatekeepers share, create and maintain knowledge

to make it a certainty within months instead of years. Much of this movement is tacit in the sense that all the necessary technical knowledge cannot be foreseen and is not available through traditional knowledge sharing practices including training, documentation of procedures etc.

Question 3: Outcomes of Hyper-Learning

Another important novel characteristic of hyper-learning are its outcomes. The resource-based theory of the firm suggests that firms exploit the learning outcomes primarily in order to gain sustainable competitive advantage; they therefore need to routinize the new capabilities and skills so that they can be easily be replicated through the continued refinement of core competencies (Nelson and Winter 1982, Winter 1995, Winter and Szulanski 2001, Winter and Szulanski 2002). Therefore, firms bundle key resources with other resources to minimize imitation by competitors (Dierickx and Cool 1989, Lippleman and Rumelt 1982, Teece 1987). Our analysis of hyper-learning firms challenges this wisdom. We instead found that firms radically simplified their rules to enable fast movements between different competency bases since this was the only way possible to keep up with the pace of innovation. Consequently, some of the studied firms had to abandon their previously established software development methodologies, which had been bundled into their CASE tools in order to replicate and refine their core capabilities in a larger scale, while impeding competitor imitation. Earlier, this bundling had been treated as an important source of competitive advantage and a core competency. However, none of the firms were pursuing such a strategy in the new situation they faced, but instead had largely unbundled their knowledge resources and capabilities.

To un-bundle resources, firms had to deploy simplified rules and design patterns. Such patterns represented additional high-level meta-knowledge that could correspond with the explosive growth of low-level detailed knowledge, which occurred as they explored new capabilities and tested different competency bases. Such low-level local knowledge was assimilated either in the form of tacit knowledge, through socialization such as mentoring (Nonaka 1994) whereby it remained tacit, or integrated into the socio-technical network in the form of loosely organized artifacts including e-mail, short memos, or quick and dirty solution frames which could be quickly discarded when their value degraded. Our results are consistent with earlier work that found that organizational routines in high-velocity environments tend be simple, experiential, and semi-structured (Eisenhardt and Martin 2000).

Toward a Theory of Hyper-Learning

When taken together, our results suggest a novel systemic configuration of hyper-learning organizations, as displayed in Figure 2. As shown in the figure, critical to hyper-learning is how and when organizations integrate specific capabilities into their repertoire and how they learn to quickly exploit them. The left side of the model shows rapid exploration that responds to the multiple learning stimuli for breadth, while its right side responds to the learning needs for the rapid and scalable exploitation and increasing depth.

A key element in the model is the dynamic interlacing of exploration and exploitation processes that removes the temporal and structural gaps between the two (as denoted by the two arrows that connect them). Hence, in hyper-learning systems, organizations not only mobilize particular learning mechanisms in order to separately shift both exploration and exploitation into "high gear," but they also have to accelerate cycles between exploration and exploitation through the tight coupling between them that results in ambidextrous learning. In this model, the dynamic capability of a firm in hyper-competitive environments comes from the firm's core

competency of accelerating cycles between exploration and exploitation as well as how effectively the four specific learning mechanisms are integrated, as shown by the dependencies in Figure 2.

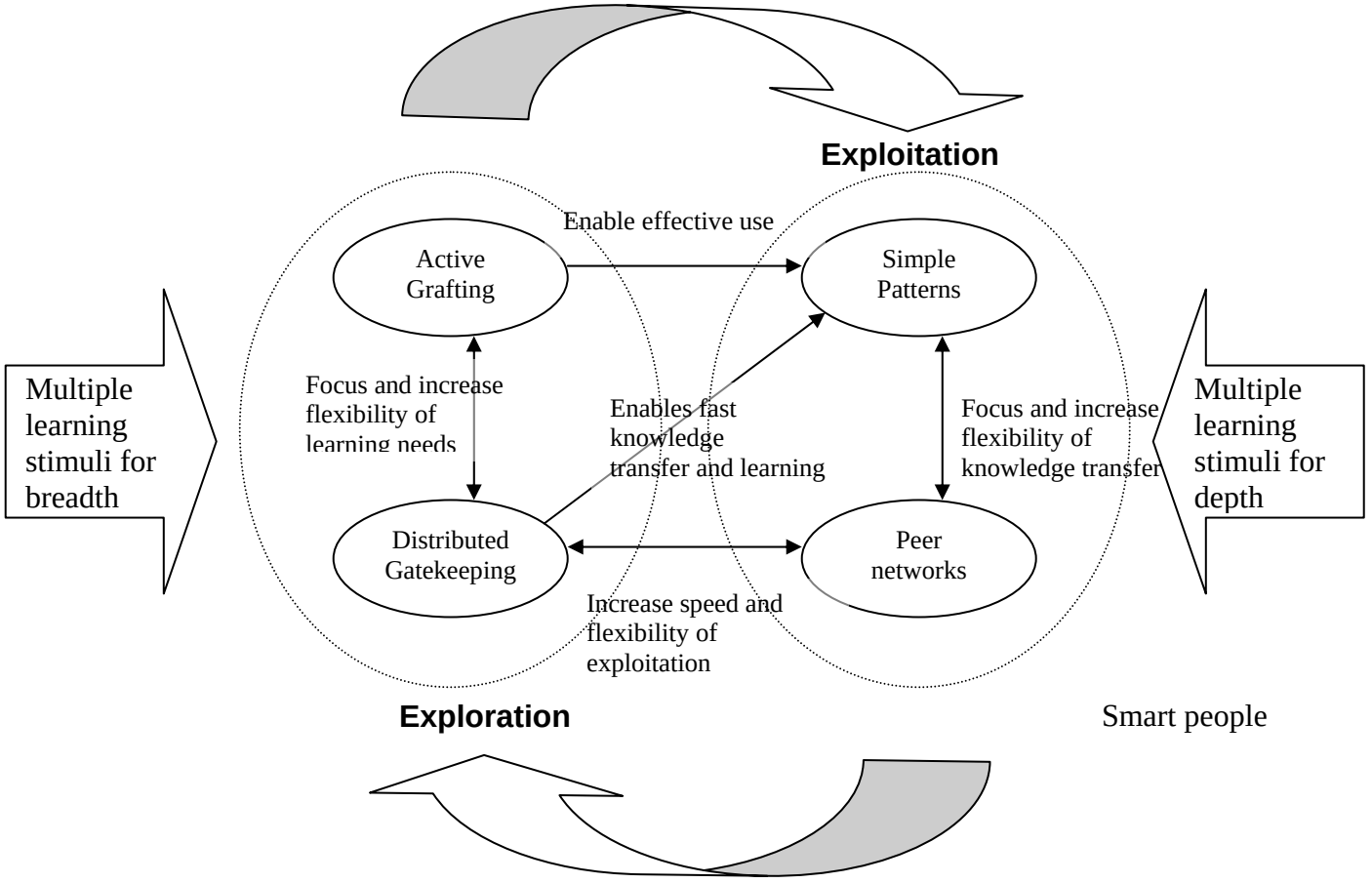

Figure 2. Hyper-Learning System

When new learning stimuli arise, hyper-learning firms need to swiftly decide whether they require a new capability or whether it needs to deepen its existing sets of expertise. If it observes a need for the new set of capabilities, then the firm needs to make swift decisions as to whether it will rely on their gatekeepers to explore new capabilities further, or whether it can import them from external sources. The outcome of exploration is made available to the rest of the organization through radically simplified routines and a tight peer expert network. This has significant implications for the processes that follow and how the organization will continually configure its resources. The resultant tacking around dynamic capability resembles the finessed dance of decisions observed in high velocity environments (Eisenhardt and Tabrizi 1995).

In hyper-learning systems, specific learning mechanisms for exploitation and exploration do not exist on their own. Rather, they exist as components of the duality that plays out the dialectics of hyper-learning. For example, active grafting and distributed gate-keeping cannot be

effective learning mechanisms for rapid exploration unless they shape and are shaped by simple patterns and peer networks for rapid exploitation. Likewise, the effectiveness of the rapid exploitations depends on the success of rapid explorations.

The hyper-learning system, as depicted above, is a dramatic departure from traditional organizational learning models both temporally and structurally. In traditional learning models, exploration functions (R&D and market research) are largely centralized in order to optimize the accuracy and scale of knowledge transfer from exploration to exploitation. On the other hand, the exploitation function is streamlined for long term and incremental learning in order to support formalized knowledge transfer mechanisms. The logic of the system is largely dictated by the need to find a balanced allocation between exploration and exploitation over time (March 1991). Thus, incremental learning and exploitation of current technology is prolonged until executives of the company realize the changes in technology and/or environments and make decisions to focus more on explorations. From the knowledge management perspective, traditional organization units involved explorations that focus primarily on knowledge creation, while others focus on knowledge re-use and transfer. To the contrary, knowledge is constantly created and re-created through the dialectic interlacing of explorations and exploitations in hyper-learning systems.

The shift from a traditional learning system based on punctuated equilibrium to a hyper-learning system based on hyper-competition is thus a disruptive organizational transformation that requires cognitive, behavioral and structural changes. In this sense, one element of the dynamic capability is the organization's capability to transform itself from a traditional learning system to the one depicted in Figure 2 -- a kind of dynamic meta-capability.

Conclusion

Our study contributes to the growing stream of research on dynamic capability as a strategic asset during hyper-competition. We explored in depth how organizations became ambidextrous in the *simultaneous* exploration and exploitation of new capabilities. We painted a picture of organizational learning that is radically different from the dominating models of organizational learning that lean on a punctuated model of strategic capability development, which recognizes distinct and separate exploitation and exploration phases. Our study complements and challenges many of the assumptions in the existing body of the literature in organizational learning. Within the organizational learning literature, our study contributes to the growing body of knowledge about dynamic capability. In particular, it provides important insights into how organizations quickly develop new capabilities. This is linked to their ability to reduce the increasing gap between exploration and exploitation both in time and structure, which is achieved through their systemic interlacing. Furthermore, we found that micro-level behaviors related to exploration and exploitation exhibit unique characteristics. This finding offers further support to the viewpoint that dynamic capabilities are not vague or tautological (Eisenhardt and Martin 2000). We can expect that there are also other mechanisms by which firms hyper-learn and different ways by which they reduce the gap between exploration and exploitation. Therefore, future research on dynamic capabilities needs to focus on specific micro-level learning behaviors as well as on organizational mechanisms and technologies that connect learning processes in order to accelerate the innovation.

For the knowledge management literature, our study challenges the prevalent dichotomy

of knowledge creation and knowledge reuse. While some scholars in this area have emphasized knowledge creation (Leonard-Barton 1995, Nonaka 1991, Nonaka and Konno 1998), many others focused on the reuse and transfer of knowledge (Argote and Ingram 2000, Davenport and Prusak 1998, Dixon 2000, Hansen 1999, Szulanski 2000). We believe that while such a dichotomy makes sense in stable environments where firms primarily exploit knowledge that is created by another part of the organization, it may of limited value to analyze knowledge management needs in high-velocity environments.

In terms of the software engineering and information systems development research, our findings point out the importance of understanding the dynamics of exploration and exploitation in software development innovation. This finding is in stark contrast with the current focus on software engineering research, which has almost solely focused on improving the exploitation speeds and quality in software development. The nearly religious focus on rigid methodologies (Lyytinen and Rose 2003, Pressman 1998), systematic process improvement programs (Humphrey 1989), and the "silver bullet" of automating software development (Brooks 1987) testify to this. Such approaches seem to be applicable only when development contingencies fall within cell II, where both the technologies and the problem-solution mappings are relatively stable. In our case, such a contingency could not be supported so organizations had to give up their investments in process technologies since their learning modalities did not meet the needs of hyper-competition. This suggests that software engineering and information systems development research should be more open to analyzing the implications of different learning modalities and environments for the development of processes, tools and methods.

Our study has several limitations. First of all, we observed only seven firms in a particular industry at a very peculiar point in time. The specific influence of software development and its techniques need to be considered before drawing general conclusions from our study. It is important to note that software development generally contains the following unique features: it is abstract in nature and therefore difficult to scale up; it is very error prone; and it is extremely learning intensive. Second, the frantic period of the "New Economy" certainly had an impact on the ways the firms operated. Thus, one might argue that these patterns of hyper-learning were in fact part of the Internet hype of that period, though they still are unique to the specific period of hyper-competition. One needs to examine other high-velocity industries to find out whether what we saw in these seven firms are indeed more general patterns that can be found in other high-velocity environments. We are cautiously positive that what we report here can be more broadly applied based on similar findings from previous studies (Brown and Eisenhardt 1997, Eisenhardt and Tabrizi 1995).

Despite these limitations, our study provides important insights for companies competing in high-velocity environments. Managers need to pay closer attention to the temporal and structural gaps between explorations and exploitations. Their ability to manage this gap significantly shapes their firm's dynamic capabilities. Furthermore, our results suggest that organizations competing in high-velocity environments may need to revisit their human resource practices. Incentive structure and personnel development practices might have originally been designed for stable environments.

Our research also provides several directions for future research. First, future research needs to examine other hyper-competitive environments in order to validate similar transformations in learning patterns. Furthermore, we might be able to find specific forms of hyper-learning that are idiosyncratic to specific industries.

Second, future research needs to study the longitudinal impacts of hyper-learning. Not

surprisingly, hyper-learning is intense and stressful. It is not clear whether organizations can sustain such a high-level of intensity for an extended period time and what cognitive overload at the organizational level does for organizations. In the future, it would therefore be necessary to examine in more detail the evolutionary patterns of hyper-learning and the organizations that pursue it- for example, there is a question of whether there are t shifts toward the punctuated learning model and when such shifts occur, do such organizations break down easily from stress. Our sample included firms working at the cutting edge of innovation in the software industry at the early phase of diffusion. One can ask the question that, while these companies acquire more capabilities and markets grow, do firms become different types of players, or do they continue to stay at the edge of the innovation curve? One implication of our study is the size of units engaged in hyper-learning. The dependency on smart people, informal networks and simple rules of thumb does not easily scale up. We could also see in our organization that they emphasized the need to find the right size for their organizational units and they were constantly struggling with the need to scale up or down, depending on their strategic position.

Third, we also noticed three dilemmas that hyper-learning poses to organizations. First, when organizations simultaneously pursue rapid exploration and exploitation, they have to sacrifice the depth of knowledge when they confront resource constraints (i.e., time, resource pool available). Hence, during hyper-competition, exploration will deal with newer, more radical and larger sets of possibilities while the exploitation deals with less rigid certainties. This implies higher risk and strategic vulnerability for their task execution and reduced effectiveness in executing their tasks. Second, hyper-learning organizations have more limited ability to quickly scale up their knowledge assets by using formal transfer knowledge mechanisms. Thus, knowledge in hyper-learning organizations will be increasingly tacit, which is more difficult to transfer on a large scale, thus making their growth potential smaller. Third, hyper-learning organizations must trade-off redundancy in their learning outcomes and competencies. By maximizing both exploration and exploitation, organizations soon face soon the limit of their learning capabilities. One mechanism for freeing capacity is to reduce redundancy and have less people know more. This strategy, however, will make them vulnerable for easily losing critical knowledge and skills through hostile maneuvers by their competitors. It is not clear how organizations can face these dilemmas of hyper-learning. Future research needs to examine these dilemmas and organizations' varying responses to them more carefully.

Finally, we have not addressed the issue of organizational performance- how did these organizational strategies and changes translate into organizational effectiveness? There are clearly other factors that influence organizational performance during hyper-competition, but absorptive capacity and its increase is surely a critical factor.

Epilogue

The recent gloom in the economy has had its impact on the firms in the study. Two firms went bankrupt even though the divisions included in the interviews had been financially successful. Of the remaining five, two smaller firms have experienced a sharp drop in customer demand as the economy shrank drastically in 2001. They are still engaged in Internet-computing and see this as their future. They also feel that the fast change is endemic to their environment. The remaining three firms are each part of a large parent organization and they have been able to continue their Internet development operations and increase them. For each of the five surviving

firms, the direction of learning has not deviated from our observations, though the pace of change in the market (but not in technology) has slowed.

The ongoing recession of 2001 and 2002 indicates that the interviews conducted in mid- and late 2000 occurred at a truly exceptional point in time. In essence, we were able to capture a synopsis of the entire hyper-competitive phenomenon by asking interviewers to reflect on its impact on systems development. While the business context now has changed, the interviews offer a window into understanding the hyper-competitive environment, which is triggered by technological change. Certainly, disruptive innovations are not singular events but more extensive processes, even though the particular period of our study was in many ways unique. Critical parameters of the technology have changed before (e.g., mainframes and client/server) and other periods have seen infusions of capital and research and development (e.g., electronics in the 1960s). So it is likely that hyper-competition will take place again in IT industries.

FIRM	Firm 1	Firm 2	Firm 3	Firm 4	Firm 5	Firm 6	Firm 7
Division Focus	Custom software development. Primarily e-business applications	B2B e-Business consulting solutions	Small spin-off of parent. Is Web-based ASP for parent company's customers	E-Business consulting specializes in mobile computing	Systems integrators to upgrade legacy systems to include Web and mobile solutions	E-Business solutions Specializes in mobile computing Need-based assembling of components and applications	Management consulting, development, IS products, networking and hosting services
Interviewee Details	Six senior employees including an executive, managers, and software architects	A senior manager of an IS development group and one of his key developers	The CIO, and the five key senior technologists who were responsible for the creation of the spin-off	Five senior employees including ISD project managers, developers, and the senior technology architect	One of the founding executives who was responsible for development of business processes	Four senior employees including a systems architect, manager, and software engineer	One senior manager of IT development services
History	15 year old firm-had been mainframe and client server shop with 500 employees in 4 locations	Part of a large, multinational business consulting company	Part of a large financial company with several thousand employees	Multinational e-Business consulting firm founded in 1995 with several thousand employees	e-Commerce development firm founded in 1996 starting with 6 employees	Large multinational e-Business consulting and software development firm	Mature, large, multinational development and IT service firm
# Employees in Division	Several hundred	Several hundred	70	100+	200+	700+	Several hundred
Typical work week	40 hours	50 hours	50 hours	60 hours	37.5 hours	Varies	37.5 hours
Employee turnover / year	18-30%	15-30%	< 10%	3%	3%	Uncertain	Uncertain

Table 2. Firm Characteristics

FIRM	Firm 1	Firm 2	Firm 3	Firm 4	Firm 5	Firm 6	Firm 7
Organizational Structure	-President -Branch manager -Field manager -Project manager	-Partner -Director -Project and technical managers	CIO, then flat	-Client manager -Project manager	Entirely flat except for salary issues.	Rigid vertical hierarchy with formalized methodologies for all aspects of business	Company is divided into autonomous units based on market sector of client
Project Team Characteristics	15-20 people including: business analysts, architects, lead developer, other developers, QA person	Architects, analysts, expert developers, rookie developers	Informal	Flat with the following roles: project assistant, technical lead, designer, information architect	Informal	Rigid vertical hierarchy	Broken down by customers (approximately 50/customer) and subsequently by teams (of 10 each)
Miscellaneous	80% of their business is repeat	Roughly $750 million budget	Spun off to make decisions at faster pace to meet needs of market	Company expands through vertical and horizontal acquisition	20-30 projects ongoing at any time	Mostly analysts. Only 30 java coders as primary job. 200 doing HTML construction and Scriptlets, etc. Remaining development done by partners.	Offices in 13 countries

Table 2. Firm Characteristics (continued)

	Rapid Expansion of Breadth of Knowledge		Rapid Expansion of Depth of Knowledge	
	Active and Strategic Gatekeeping	Active Grafting of External Knowledge	Reuse of Simple Rules	Building Peer Expertise Network
Firm 1	Lead developers: • The firm only develops systems that the lead developers want to develop. • The firm invests heavily in off-site training of lead developers.	• Active purchasing of software component libraries from external sources.	• Web page templates are reused.	• Place people with appropriate skills to spread knowledge around through mentoring • Mentors review code written by others.
Firm 2	Technology gurus: • Technology geeks who are deeply plugged into external technology communities. • The firm piggybacks on them for new technologies.	• Import external software code library from open-source community. • Outsource contractors for specific areas.	• Design and analysis patterns are reused. • Loosened up rigid development methodology.	• The firm increasingly relies on informal on-the-job mentoring. • Gatekeepers provide about 75% of technical inquiries.
Firm 3	New CIO: • He was brought in because of his substantial experiences in Web-based systems development.	• Outsourcing programmers. • Hiring a new CIO.	• A simple informal 3-step methodology is used.	• The firm is planning to implement a mentoring program.
Firm 4	Skill leaders: • There are skill leaders in eight key technology areas. • They are responsible for environmental scanning.	• Outsourcing web hosting and ASP. • Purchasing components and platforms.	• A simple 3-step methodology consisting of the envision, shape, and realize phases.	• Each new employee is assigned to a mentor and a regional training program.
Firm 5	Experts: • They are 5-10% of employees who are expected to learn new technology and share their knowledge with the rest of firm.	• Minimum purchase of software components. • Limited contracting for the newest technologies.	• The development methods for web-based systems are much simpler than the traditional methods.	• Experts are expected to teach others through mentoring and informal workshops. • Projects maintain nearly "flat" structures in order

	• For each key area, the firm had 1-3 experts.			to leverage individual expertise.
Firm 6	Boundary Scanners: • They specialized in particular technology areas to keep up with changes in technology. • They work closely with selected business partners in order to experiment with emerging technologies.	• Outsourcing for custom code developments. • Extensive use of components.	• Analysis and overall solution design patterns are heavily reused.	• Each new hire is mentored for three months. After the initial mentoring period, a local mentor will be assigned for 2-4 years.
Firm 7	Mentors: • The top 1% of technical support group who assist frontline consultants is expected to become experts in certain areas. • The mentors were assigned to key strategic projects.	• Outsourcing certain special areas. • Acquire companies instead of outsourcing, if necessary.	• Most aggressively pursuing code reuse and formal development methodology. • However, it focuses on "bare essentials" of the methodology.	• The firm rotates "mentors" around different projects to transfer their expertise.

Table 3. Summary of the Findings

References

Argote, L., P. Ingram. 2000. Knowledge transfer: A basis for competitive advantage in firms. *Organizational Behavior and Human Decision Processes* **82** (1) 150-169.

Argyris, C., D. A. Schön. 1978. *Organizational Learning: A Theory of Action Perspective*, Addisson-Wesley, Reading.

Barney, J. B. 1996. The resource-based theory of the firm. *Organization Science* **7** (5) 469.

Basu, A., A. Kumar. 2002. Workflow management issues in e-business. *Information Systems Journal* **13** (1) 1-14.

Boyatzis, R. 1998. *Transforming Qualitative Information: Thematic Analysis and Code Development*, Sage, Thousand Oaks, CA.

Brooks, C. 1987. No silver bullet--The essence and the accidents of software engineering. *IEEE Computer* **14** (2)

Brown, S. L., K. M. Eisenhardt. 1997. The art of continuous change: Linking complexity theory and time-paced evolution in relentlessly shifting organizations. *Administrative Science Quaraterly* **42** (1-34)

Chandler, A. D. 1962. *Strategy and Structure: Chapters in the History of the American Industrial Enterprise*, MIT Press, Cambridge, MA.

Christensen, C. M. 1997. *The innovator's dilemma: When new technologies cause great firms to fail*, Harvard Business School Press, Boston, MA.

Cohen, W. M., D. A. Levinthal. 1990. Absorptive capacity: A new perspective on learning and innovation. *Administrative Science Quarterly* **35** 128-152.

D'Aveni, R. A. 1994. *Hypercompetition: Managing the Dynamics of Strategic Maneuvering*, The Free Press, New York.

Davenport, T. H., L. Prusak. 1998. *Working Knowledge: How Organizations Manage What They Know*, Harvard Business School Press, Boston.

Dierickx, I., K. Cool. 1989. Asset stock accumulation and sustainability of competitive advantage. *Management Science* **35** (12) 1504-1511.

Dixon, N. M. 2000. *Common Knowledge: How Companies Thrive by Sharing What They Know*, Harvard Business School, Boston, MA.

Eisenhardt, K. M. 1989. Building theories from case study research. *Academy of Management Review* **14** (4) 532-550.

Eisenhardt, K. M., J. A. Martin. 2000. Dynamic capabilities: What are they? *Strategic Management Journal* **21** 1105-1121.

Eisenhardt, K. M., C. B. Schoonhoven. 1996. Resource-based view of strategic alliance formation: Strategic and social effects in entrepreneurial firms. *Organization Science* **7** (2) 136-150.

Eisenhardt, K. M., B. N. Tabrizi. 1995. Accelerating adaptive processes: Product innovation in the global computer industry. *Administrative Science Quarterly* **40** 84-110.

Fredickson, J. W. 1984. The comprehensiveness of strategic decision processes: Extension, observation, future directions. *Academy of Management Journal* **27** (3) 445-467.

Galunic, D. C., K. M. Eisenhardt. 1996. The evolution of intracorporate domains: Divisional charter losses in high-technology, multidivisional corporations. *Organization Science* **7** (3) 255-282.

---. 2001. Architectural innovation and modular corporate forms. *Academy of Management Journal* **44** (6) 1229-1250.

Glasser, B. G., A. L. Strauss. 1967. *The Discovery of Grouped Theory: Strategies for Qualitative Research*, Aldine Publishing Company, New York, NY.

Grant, R. M. 1996. Prospering in dynamically-competitive environments: Organizational capability as knowledge integration. *Organization Science* **7** (4) 375-387.

Hansen, M. T. 1999. The search-transfer problem: The role of weak ties in sharing knowledge across organization subunits. *Administrative Science Quarterly* **44** 82-111.

Henderson, R., I. Cockburn. 1994. Measuring competence? Exploring firm effects in pharmaceutical research. *Strategic Management Journal* **15** 63-84.

Henderson, R. M., K. B. Clark. 1990. Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly* **35** (1) 9-30.

Humphrey, W. S. 1989. *Managing the Software Process*, Addison-Wesley, Reading, MA:.

Ilinirtch, A. Y., R. A. D'Aveni, A. Y. Lewin. 1996. New organizational forms and strategies for managing in hypercompetitive environments. *Organization Science* **7** (3) 211-220.

Imai, K., N. Ikujiro, H. Takeuchi. 1985. Managing the new product development process: How Japanese companies learn to unlearn. R. H. Hayes, K. B. Clark and J. Lorens, eds. *The Uneasy Alliance: Managing the Productivity-Technology Dilemma*, Harvard Business School Press, Boston, MA, 337-375.

Kim, L. 1998. Crisis construction and organizational learning: Capability building in catching-up at Hyundai Motor. *Organization Science* **9** (4) 506-521.

Lant, T. K., S. J. Mezias. 1992. An Organizational Learning Model of Convergence and Reorientation. *Organization Science* **3** (1) 47-71.

Leonard, D. 1995. *Wellsprings of Knowledge: Building and Sustaining the Sources of Innovation*, Harvard Business School, Boston.

Leonard-Barton, D. 1995. *Wellsprings of knowledge: Building and sustaining the sources of innovation*, Harvard Business School Press, Boston, MA.

Lippleman, S. A., R. P. Rumelt. 1982. Uncertain imitability: An analysis of interfirm differences in efficiency under competition. *Bell Journal of Economics* **13** 418-438.

Lyytinen, K., G. Rose. 2003. Disruptive information systems innovation: The case off internet computing. *Information Systems Journal*

Lyytinen, K., G. Rose, R. Welke. 1998. The brave new world of development in the internetwork computer architecture (InterNCA): or how distributed computing platforms will change systems development. *Information Systems Journal* **8** (3) 241-253.

March, J. G. 1991. Exploration and Exploitation in Organizational Learning. *Organization Science* **2** (1) 71-87.

Montealegre, R. 2002. A process model of capability development: Lessons from the electronic commerce strategy at Bolsa de Valores de Guayaquil. *Organization Science* **13** (5) 514-531.

Nelson, R. R., S. G. Winter. 1982. *An Evolutionary Theory of Economic Change*, Belknap Press, Cambridge, MA.

Nonaka, I. 1991. The Knowledge-Creating Company. *Harvard Business Review* **69** (6) 96-104.

---. 1994. A dynamic theory of organizational knowledge creation. *Organization Science* **5** (1) 14-37.

Nonaka, I., N. Konno. 1998. The Concept of "Ba": Building a foundation for Knowledge Creation. *California Management Review* **40** (3) 40-55.

Nonaka, I., H. Takeuchi. 1995. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*, Oxford University Press, New York.

Peach, R. W. 2002. The ISO 9000 Handbook Fourth Edition. Fairfax, VA,

Penrose, E. T. 1959. *The Theory of the Growth of the Firm*, Wiley, New York.

Peteraf, M. 1993. The cornerstones of competitive advantage: A resource-based view. *Strategic Management Journal* **14** (3) 179-191.

Pisano, G. P. 1994. Knowledge, integration, and the locus of learning: An empirical analysis of process development. *Strategic Management Journal* **15** 85-100.

Porter, M. E. 1980. *Competitive Strategy: Techniques for Analyzing Industries and Competitors*, The Free Press, New York.

Prahalad, C. K., G. Hamel. 1990. The Core Competence of the Corporation. *Harvard Business Review* **68** (3) 79-91.

Pressman, R. 1998. Can internet-based applications be engineered? *IEEE Software* (Sept-Oct) 104-110.

Sambamurthy, V., R. W. Zmud. 2000. The organizing logic for an enterprise's IT activities in the digital era-A prognosis of practice and a call for research. *Information Systems Research* **11** (2) 105-114.

Schumpeter, J. A. 1934. *The Theory of Economic Development*, Harvard University Press, Cambridge, MA.

Straub, D. W., R. T. Watson. 2001. Transformational issues in researching IS and Net-enabled organizations. *Information Systems Research* **12** (4) 337-345.

Strauss, A., J. Corbin. 1990. *Basics of Qualitative Research: Grounded Theory Procedure and Techniques*, Sage, Newsbury Park, CA.

Szulanski, G. 2000. The process of knowledge transfer: A diachronic analysis of stickiness. *Organizational Behavior and Human Decision Processes* **82** (1) 9-27.

Teece, D. 1987. Profiting from technological innovation: Implications for integration, collaboration, licensing, and public policy. D. J. Teece, eds. *The Competitive Challenge*, Ballinger, Cambridge, MA,

Teece, D. J., G. Pisano, A. Shuen. 1997. Dynamic capabilities and strategic management. *Strategic Management Journal* **18** (7) 509-533.

Tuomi, I. 2002. *Networks of Innovation: Change and Meaning in the Age of the Internet*, Oxford University Press, New York.

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

Volberda, H. W. 1996. Toward the flexible form: How to remain vital in hypercompetitive environments. *Organization Science* **7** (4) 359-374.

Wernerfelt, B. 1984. A resource-based view of the firm. *Strategic Management Journal* **5** (2) 171-180.

Winter, S. G. 1995. Four r's of profitability: Rents, resources, routines, and replication. C. A. Montgomery, eds. *Resource-based and Evolutionary Theories of the Firm: Towards a Synthesis*, Kluwer Academic Publishers, Norwell, MA, 147-178.

Winter, S. G., G. Szulanski. 2001. Replication as strategy. *Organization Science* **12** (6) 730-743.

---. 2002. Replication of organizational routines: Conceptualizing the exploitation of knowledge assets. N. Bontis and C. W. Choo, eds. *The Strategic Management of Intellectual Capital and Organizational Knowledge: A Collection of Readings*, Oxford University Press, New York, 207-222.

Yin, R. K. 1994. *Case Study Research: Design and Methods*, Sage, Thousand Oaks, CA.

Young, G., K. G. Smith, C. M. Grimm. 1996. "Austrian" and industrial organization perspectives on firm-level competitive activity and performance. *Organization Science* **7** (3) 243-254.

Zahra, S. A., G. George. 2002. Absorptive capacity: A review, reconceptualization, and extension. *Academy of Management Review* **27** (2) 185-203.

Zhang, A., K. Lyytinen. 2001. A framework for component reuse in a metamodelling based software development. *Requirements Engineering Journal* **6** 116-131.

芽|Sprouts

Working Papers on Information Systems | ISSN 1535-6078

Office:
Sprouts
University of Amsterdam
Roetersstraat 11, Room E 2.74
1018 WB Amsterdam, Netherlands
Email: admin@sprouts.aisnet.org

芽|Sprouts