Strategic Management Journal
Strat. Mgmt. J., **26**: 25–45 (2005)
Published online 28 October 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/smj.433

WHERE DO CAPABILITIES COME FROM AND HOW DO THEY MATTER? A STUDY IN THE SOFTWARE SERVICES INDUSTRY

SENDIL K. ETHIRAJ,[1] PRASHANT KALE,[1]* M. S. KRISHNAN[1] and JITENDRA V. SINGH[2]
[1] *University of Michigan Business School, Ann Arbor, Michigan, U.S.A.*
[2] *The Wharton School of Business, University of Pennsylvania, Philadelphia, Pennsylvania, U.S.A.*

Recent years have witnessed a surge of interest in the notion of capabilities as an important source of competitive advantage. This recognition has, in turn, placed emphasis on the question of where and how these capabilities emerge and how they influence firm performance. The present paper is an attempt to address this question. Using a large sample of detailed project-level data from a leading firm in the global software services industry, we attempt to empirically study the importance of capabilities. We find that two broad classes of capabilities are significant. The first class, which we label client-specific capabilities, is a function of repeated interactions with clients over time and across different projects. This learning from repeated interactions with a given client reduces project execution costs and helps improve project contribution. The second class, termed project management capabilities, is acquired through deliberate and persistent investments in infrastructure and systems to improve the firm's software development process. Our empirical results suggest that the marginal returns to acquiring different capabilities may be different and an understanding of such trade-offs can improve firm decisions to improve and/or acquire such capabilities. We discuss the key contributions of our paper and the implications for future research on capabilities.

INTRODUCTION

In recent years strategy scholars have increasingly agreed that non-imitable and non-substitutable organizational capabilities (and resources) are a key source of inter-firm performance differences (Barney, 1991; Dosi, Nelson, and Winter, 2000; Nelson, 1991; Rumelt, 1984; Wernerfelt, 1984). This recognition has, in turn, placed emphasis on the question of where and how these capabilities emerge and how they influence firm performance. Although there are a number of theoretical arguments about the characteristics of resources or capabilities that yield competitive advantage (Barney, 1991) and what prevents their imitation (Dierickx and Cool, 1989; Peteraf, 1993), we have limited understanding of where capabilities come from or what kinds of investment in money, time, and managerial effort is required in building them.

Keywords: organizational capabilities; firm performance; software services
*Correspondence to: Prashant Kale, University of Michigan Business School, 701 Tappan Street, Room D4209, Ann Arbor, MI 48109, U.S.A. E-mail: kale@umich.edu

Received 10 December 2002
Final revision received 4 June 2004

Furthermore, if the development of capabilities requires deliberate and sustained investment of financial and managerial resources, both of which have alternative uses, it becomes important to understand the costs and benefits of such investments. In other words, different capabilities may entail different financial and managerial costs and yield dissimilar performance benefits. The systematic understanding of such trade-offs promises to enrich the theory and practice of strategy. This paper makes a modest attempt to systematically address some of these questions.

This paper investigates two interrelated research questions: one, where do capabilities come from, and two, how do capabilities affect firm performance? We combine in-depth interview data with detailed, large sample, project-level data spanning a 6-year period from one firm in the Indian software services industry to carefully address both questions. In contrast with prior research, the rich, disaggregated project-level data about resource inputs, project characteristics, capability metrics, and profitability that we collected allow us to delve deeper into the capabilities–performance link.

Several reasons make the Indian software industry an attractive context for the study of firm capabilities. First, there is a widely shared view that much of the Indian software industry's explosive growth over the last decade is accounted for by factor cost differences between India and the developed country markets (Arora *et al*., 2001; Nasscom, 2001). The implication is that performance differences are driven not by firm capability differences, but by country-level comparative cost advantages. While factor cost differences definitely exist, even a cursory examination of the data suggests that firm-level explanations cannot be discounted. The Indian software services industry accounted for $6.2 billion of export revenues in 2000–01 (Nasscom, 2001). A telling statistic is that 0.8 percent (25 firms) of the firms in the industry (nearly 3000 firms) accounted for 60 percent of export revenues. Therefore, a relatively small set of firms growing at a compounded average annual rate of about 45–65 percent over the last decade accounts for much of the activity in the Indian software services industry. Our premise is that a detailed examination of the economics of one or more of these 25 firms can afford useful insight into the micro-foundations of the capabilities that underlie such sustained and robust growth in a competitive industry.

Building on research on firm capabilities in general and on our detailed fieldwork and interviews with the project managers of several firms in the software services industry, we argue that two sets of capabilities are important in the software services industry: client-specific capabilities and project management capabilities. Client-specific capabilities are a function of repeated interaction with a given client across multiple projects over time. They largely reflect tacit knowledge of the client's business domain and operating routines acquired through repeated interaction with the client. In contrast, project management capabilities are acquired through deliberate and persistent investments in infrastructure (systems and processes) and training to improve the firms' software development processes. They reflect technical capabilities in software design, development, and execution. The development of these capabilities rests not only on implicit learning-by-doing processes but also on deliberate, proactive investments in building them. For example, drawing on industry experience, some scholars derived economic models to show that it makes economic sense for software vendors to initiate the first project with a client at low prices even if it amounts to a modest loss on the project. The first project serves as a platform for the development of client-specific and project management capabilities that help significantly reduce costs in the long run and ultimately generate positive returns (Whang, 1995). In our empirical analyses, we estimate the marginal contribution, cross-sectionally and temporally, of the two capabilities to project profitability.

The study makes two principal contributions to the extant literature on capabilities. First, we argue, and empirically demonstrate, that firm capabilities are often context-specific and fruitful research in this area might emanate from enjoining an in-depth study of the capabilities specific to a context and careful empirical estimation of their significance and value. The distinguishing feature of our work is that we conceptualize the notion of capabilities at a more micro-level within the firm, namely the project(s) which the firm executes for its clients, develop appropriate measures for these capabilities, and examine their evolution and impact on financial performance. Second, we argue, and show, that not all capabilities provide the same marginal contribution to performance.

This is significant because it suggests that if different capabilities have different costs and benefits associated with their development or acquisition, managers should pay attention to understanding these trade-offs in making investments in capability development. More broadly, our study advocates a shift in the debate from whether or not capabilities matter to 'what' capabilities matter and 'how.'

The rest of the paper is organized as follows. In the next section, we briefly outline the research literature on capabilities and how capabilities drive performance differences. We then describe the Indian software industry context in some detail and develop our hypotheses. In the following section we describe the data, the measures, and empirical estimation procedures. Finally, we present the results and discuss their implications for research on capabilities.

THEORY

What are capabilities and why are they important?

The notion of capabilities can be traced back to Penrose (1959) and Andrews (1971), among others (see also Selznick, 1957). Penrose (1959: 25) suggested that resources consist of a bundle of potential services. While these resources or factor inputs are available to all firms, the 'capability' to deploy them productively is not uniformly distributed. Analogously, Andrews (1971) argued that the 'distinctive competence' of an organization is more than what it can do; it is what it can do particularly well. Building upon this earlier work, recent literature on the resource-based view conceptualizes resources and capabilities along two lines. One set of authors (see, for example, Barney, 1991; Peteraf, 1993) tend to define resources rather broadly so as to 'include all assets, capabilities, organizational processes, firm attributes, information, knowledge, etc.' (Barney, 1991: 101). Other authors, however, have sought to clearly delineate between resources and capabilities (Amit and Schoemaker, 1993; Grant, 1991) by arguing that 'resources consist ... of know-how that can be traded, financial or physical assets, human capital etc. ... [whereas] capabilities ... refer to a firm's capacity to deploy resources' (Amit and Schoemaker, 1993: 35). In this paper, we adopt the latter conceptualization of firm capabilities in developing our arguments about where they come from and how they matter.

These definitional and conceptual differences notwithstanding, strategy researchers agree that both resources and capabilities are essentially assets with rent-generating potential. The resource-based view literature largely emphasizes two kinds of rents[1] they can generate. The first parallels the textbook notion of (Ricardian) rents from scarce resources. Ownership of a scarce resource (say land in Manhattan) enables the owner to enjoy superior rents relative to competitors who do not own the resource (they lease the land). The scarcity rents here are rooted in the inelastic supply curve for the resource. In addition, there is the implicit condition that, all else being equal, the cost of ownership is less than the lease cost. This means that at the time the scarce resource was acquired its price was less than its (future) marginal product (Peteraf, 1993).

A second type of rent is quasi-rents (Klein, Crawford, and Alchian, 1978). Quasi-rents 'are the excess of an asset's value over its salvage value or its value in its next best use' (Peteraf, 1993: 184). Quasi-rents are often associated with capabilities. The primary reason for this is that they are considered to be a product of specialized assets that are embedded within the organizational context (Rumelt, 1987) such that their optimal deployment is contingent on the presence of other complementary assets (e.g., managers, culture, technology) or even learning how to deploy the bundle of assets efficiently. Additionally, there is a cost involved in transferring this asset along with the complementary assets to another firm, thereby reducing its productive value (Langlois, 1992). This difference in value is the quasi-rent accruing to the owner of the asset. Usually, quasi-rents are a function of the uncertainty about the production function underlying the deployment of resources (Rumelt, 1984). In this paper, our notion of capabilities reflects assets that can generate quasi-rents.

Where do capabilities come from?

Traditionally, strategy research has devoted little attention to the issue of where capabilities

[1] Note that Winter (1995) refers also to Schumpeterian rents—entrepreneurial rents or rents from innovation—and monopoly rents—rents from output restriction under inelastic demand. Since neither of these rents is associated with resources or capabilities we do not discuss them here.

come from. Nelson and Winter (1982) made one of the early theoretical attempts to understand this question. They viewed the firm as bundles of path-dependent knowledge bases. Over time, firms' knowledge, accumulated through 'learning by doing,' is embedded in bundles of 'routines' that are likened to the genetic material of the firm. Routines, a central concept in evolutionary theory, involve repetitive patterns of activity, require investment in routine-specific human and physical capital, and are easily recognized as belonging to a class (Winter, 1990). An important element of their theory is the description of the firm as a historical entity, with productive knowledge being a result of endogenous, learning-by-doing processes. Consequently, this perspective sees firms as entities that possess heterogeneous capabilities as a function of their routines and search processes. These capabilities are rooted in the organizational skills and routines that serve as organizational memory to repetitively execute the sequence of productive activities without trouble. At their core, these organizational skills and routines embody knowledge and competence in carrying out the productive activities that the firm is engaged in. Building on the basic idea that history matters and that capabilities are rooted in contextually embedded knowledge underlying the production function, others have emphasized the significance of absorptive capacity (Cohen and Levinthal, 1990) and asset stocks and flows (Dierickx and Cool, 1989) in driving capabilities-based competition.

Some researchers have also suggested that capabilities are not merely the result of tacit accumulation of experience embedded in routines and learning by doing. They are also the result of deliberate investments in organizational structure and systems to make constant improvements in those routines and practices (Zollo and Winter, 2002). Organizations strive to adapt their operating processes through proactive actions dedicated to process improvements. These may include explicit efforts to continuously learn and capture the lessons from prior experience of self or others (Collis, 1996; Zollo and Winter, 2002) and incorporate those lessons to make improvements in prevalent practices, or create formal mechanisms to coordinate and institutionalize the improvement efforts (Kale *et al*., 2002). Although the notion of making deliberate investments to improve firm capabilities may be understood uniformly by most firms, there are idiosyncratic firm-level differences in the timing of this effort, the nature and amount of the investment and effort they undertake, and the internal organizational mind-set that supports this process. These differences may get reflected in significant heterogeneity across firms with respect to the capabilities that result from this effort.

In sum, it appears that in operationalizing the notion of capabilities it is important to show that: (1) capabilities involve the deployment of resources and there are strong theoretical reasons that undergird how and why they generate rents; (2) capabilities tend to evolve over time to reflect the joint effects of passive learning-by-doing and deliberate firm-level investments in learning and making improvements; and (3) capabilities are hard to imitate or easily acquire in factor markets and this forms the basis for rent generation.

How do capabilities matter?

The question 'How do capabilities matter?' is one that derives from the question posed earlier, 'Where do capabilities come from'? We argued above that capabilities reflect the evolutionary process of deliberate firm-specific investments and the largely tacit 'learning-by-doing' that firms engage in. This in turn results in heterogeneity of firms and the consequent differences in their performance. If we assume for the moment that 'learning-by-doing' is distributed uniformly[2] among firms, then most of the *ex post* firm performance heterogeneity is in fact a function of differences in the deliberate, firm-specific investments made (see Helfat, 1994, for an excellent account of how differential R&D investments by a sample of petroleum firms led to firm heterogeneity). Differences in the *ex post* productive value of firm-specific investments suggest that firms face significant *ex ante* trade-offs in their choices of firm-specific investments made to acquire certain capabilities. In the main, we expect that the trade-offs themselves are likely to vary between firms as well as within firms over time.

Each firm needs to make a set of strategic choices that fit together as a system (Porter, 1991). In making these choices, firms face significant trade-offs, and the nature of the trade-offs can vary

[2] The assumption of uniformity in learning-by-doing is unrealistic simply because it is likely to be endogenous with the choices of firm-specific investments. In effect, this assumption is plausible only if we can separate out learning-by-doing from the choices about firm-specific investments.

over time as well. The trade-offs are primarily a function of the interdependencies between the various strategic choices of the firm (Levinthal, 1997). When the choice to invest in acquiring a capability shares positive interactions with other choices within the firm, the marginal benefit of acquiring the capability is likely to be higher than in the case when the choice shares negative interactions with other choices made by the firm. Simply put, different capabilities are likely to yield different marginal benefits to the firms making the investments in such capabilities. If indeed money and managerial time and effort are scarce and firms need to allocate these scarce resources among competing initiatives to acquire relevant capabilities, then it is of significant theoretical and practical importance to understand the trade-offs they face in doing so. Theoretically speaking, the answer to this puzzle is rather obvious—managers should invest in acquiring capabilities that yield the greatest marginal returns to the investment. In practice, however, such a cost–benefit calculus is far from simple. We face formidable theoretical and empirical challenges because the interdependencies between the various strategic choices that a firm makes are often unknown and the impact on the firm of altering one choice can be unpredictable (Ethiraj and Levinthal, 2004). Moreover, as a firm attains critical levels on a given capability, its marginal returns to investing in the same capability might decline. Alternatively, the marginal returns to building other complementary capabilities might increase. In this paper, we make a modest attempt to examine the marginal returns to different capabilities, both cross-sectionally and temporally.

Empirical research on capabilities

The empirical literature on capabilities is fast growing and we can partition this body of work along two broad lines. The first stream of work includes detailed historical accounts that track alternative choices made by firms and their performance outcomes. Several excellent papers in this tradition provide useful insight into the historical development and evolution of capabilities (e.g., Iansiti and Khanna, 1995; Rosenbloom, 2000). Unfortunately, such methods do not permit the estimation of the significance and value of capabilities. The second stream of research that includes large-sample empirical work has attempted to do this. Even in this tradition, however, few studies have managed to adequately capture the spirit of the idea. They usually fall short either in the choice of the independent variables employed to measure capabilities or the dependent variable used to measure performance. Many studies have measured capabilities using aggregate indicators such as R&D intensity (e.g., Silverman, 1999) at the firm level. But if capabilities critically reside at the operational level within firms, aggregate firm-level measures may tend to mask much of the variance within firms.

Some recent studies have identified and measured more disaggregated capabilities to address this limitation. For instance, Henderson and Cockburn (1994), using survey data attempted to get at more disaggregated measures of R&D capability at the program level. They measured architectural competence (ability to integrate knowledge within the firm) and component competence (locally embedded knowledge) at the R&D program level to predict patenting productivity. Similarly, McGrath, MacMillan, and Venkataraman (1995) measured firm competence at pursuing new initiatives and identified that deftness and comprehensiveness are important precursors to competence acquisition and ultimately to the emergence of competitive advantage. These studies and others in this tradition (e.g., Schroeder, Bates, and Junttila, 2002), however, have been limited to assessing firm performance with disaggregated, nonfinancial measures of performance such as patenting or survey-based self-reports of performance. This is not surprising given that it is extremely difficult to relate disaggregated measures of capabilities to aggregate, financial measures of firm performance. Makadok and Walker (2000), in a notable exception, examine the financial returns to forecasting ability in the money fund industry. Along similar lines, Brush and Artz (1999) explore the trade-offs in various services offered by different firms and their impact on revenue per transaction in the veterinary medicine industry.

Our study adopts an approach that is similar to this latter set of studies. We construct disaggregated, context-specific capability measures that are as close to the operational level as is practically possible and we construct disaggregated financial measures of performance at the same operational level. We also track the evolution of these capability measures over time. Most importantly, our study is distinctive from these prior studies in that

we examine the choices made by a single firm over time and evaluate the performance trade-offs or the marginal returns to different capabilities that it sought to build over a 6-year period. The following section develops the key hypotheses advanced in this study.

OVERVIEW OF THE INDIAN SOFTWARE SERVICES INDUSTRY

In this section we provide a brief overview of the Indian software services industry. We focus on the demand- and supply-side economics of the industry and attempt to draw out the nature and type of capabilities that might have the potential to generate rents.

The Indian software services industry is relatively young, with many of its most mature companies incorporated in the late-1970s and early 1980s. The domestic market for IT services was always small and continues, even today, to account for only about 25–30 percent of the industry's sales (Nasscom, 2001). The Indian industry received a big boost in the early 1990s when the demand for IT services in the developed world outstripped the available supply of skilled labor. India, which at that time was graduating 150,000 English-speaking engineers a year with only a limited demand for their services within the country, was well placed to take advantage of this opportunity. Companies in the developed world began focusing on India to leverage its low-cost, English-speaking IT manpower. The low levels of initial investment required to enter the software services business and the minimal regulatory intervention from the Indian government, created few, if any, entry barriers or constraints in these early years. Several hundred Indian firms were founded to exploit this opportunity and the industry witnessed very rapid growth. Overall, there was a general consensus that macro factors such as the access to large, low-cost, English-speaking technical manpower in India and improvements in IT-related infrastructure (e.g., high-bandwidth communication lines) set up by private and state-owned enterprises positively influenced the competitive advantage and growth of Indian companies (Arora *et al*., 2001; Nasscom, 2001).

Demand for Indian software services

Faced with a small and undeveloped domestic software services market, Indian software firms focused primarily on the export market. Their early work, however, was neither technologically very sophisticated nor critical to clients' businesses. Clients usually did the high-end work such as requirement analysis and top-level design either in-house or through U.S.-based consultants. They outsourced the low-end, labor-intensive work such as low-level design, coding, testing, support and maintenance to Indian companies to leverage their low-cost activity base. Clients retained the high-end work in the early years either because their Indian vendors did not possess the requisite skills to undertake these activities or, even if they did, clients did not have sufficient confidence to entrust such activities to them. Thus, the origin of the Indian software industry was firmly rooted in performing low-end, technically less demanding and labor-intensive work for the global IT industry and exploiting labor cost arbitrage opportunities between India and developed country markets (Nasscom, 2001).

Over time, however, there was a change in this trend. The low entry barriers in the Indian software services industry triggered abnormally high levels of entry by new firms. Between 1989 and 1998, over 3000 software services firms were founded that aspired to serve export markets. Consequently, domestic competition in the labor market (i.e., for trained engineers) shot up. It was not enough any more to just possess low-cost labor resources and exploit arbitrage opportunities. Firms also needed to improve the productivity of labor to compete effectively in the market. Thus, some firms began a systematic push to build high-end software capabilities, move up the value chain, and improve revenue per employee figures. Consequently, the leading Indian firms today are also the leading firms in the global market (see *Market-Guide*, 2002).

Since the mid-1990s there has been a distinct shift in the nature of software projects executed by the leading Indian software firms. They gradually shifted their role from that of merely implementing a design provided by their overseas clients to becoming active participants in the design of the complete application product. As a consequence, they now span the full spectrum of jobs from highly labor-intensive code migration work such as the integration of old mainframe-based systems

into new e-commerce platforms, or developing new code for pre-designed applications and software tools, to projects that involve both conceptual design and implementation of customer relationship applications and supply-chain management systems. The capacity to execute such a range of jobs enabled these firms to deliver end-to-end solutions and, thereby, lay claim to a larger share of the client's IT budget and compete for it with leading firms in the United States such as IBM and Accenture.

Overall, while comparative cost advantages do exist for Indian software firms, they are by no means sufficient or even sustainable. First, the nature of services provided by these firms involves specialized work (e.g., designing supply chain management systems or setting up trading exchanges) that requires not only technical knowledge of software design but also an in-depth understanding of the client's industry and business process. Second, even firms such as IBM, Sapient, and Accenture have set up subsidiaries in India exploiting the same cost advantages. These subsidiaries undertake software projects involving both design and development on latest-technology platforms and business applications. This development not only supports the argument that Indian software teams are capable of executing these high-value projects but also erodes much of the Indian firms' traditional cost advantages.

Supply of Indian software services[3]

Two aspects of the Indian software services industry are critical to understanding its supply-side economics: (a) where and how the software development process is managed and organized; and (b) what type of contracts are used to provide the services. We elaborate below.

Indian software firms have traditionally executed two types of projects: onsite and offshore. In onsite projects the Indian firm supplies software professionals who possess the requisite technical skills that the clients demand. The entire project is then developed and executed at the clients' site. In offshore projects, in contrast, the Indian firm typically sends a few software professionals to the client site to understand its requirements and specifications, but thereafter the entire software is developed in India. The post-development support and maintenance of the software is also carried out largely from India. In some cases, a hybrid of the two types is also observed. Obviously, the offshore development model is more cost effective due to labor market arbitrage. From a cost standpoint, the greater the proportion of work completed offshore, the lower the cost of project execution (Gopal *et al*., 2003). This is primarily because in onsite projects employees need to be paid in accordance with host country norms,[4] which erodes a significant proportion of the cost-based advantages that Indian firms enjoy.

In the early days of the Indian software industry, Indian companies executed a majority of the projects onsite. This happened because, first, the overseas clients had limited confidence in the Indian firms' ability to execute projects in conformance to their needs. Second, the Indian firms also had only a limited understanding of clients' needs and often required close and regular interaction with the client. But, over time, as overseas clients developed confidence in the software capabilities of their Indian vendors and the vendors, in turn, developed a better understanding of clients' needs, it was possible to relocate the bulk of the project development activities to India to take full advantage of its low-cost development base. Improvements in the infrastructure for long-distance communication and data transfer facilitated this process as well.

The second significant feature of software services, both in India and worldwide, involves the type of contract adopted in software outsourcing arrangements. Contracts can be broadly classified into two categories: fixed price and time & material (Banerjee and Duflo, 2000). In fixed price contracts the vendor charges a fixed fee for its services, which is usually negotiated before the start of the project. Although the vendor bears most of the risk in this case, efficient project management can yield potentially higher margins. In a

[3] The material here draws heavily from an excellent article on the Indian software industry by Arora *et al*. (2001).

[4] In the United States, issue of H1B visas to overseas software professionals requires the sponsoring firm to certify that the professionals are being paid wages commensurate with what a U.S.-based employee with similar qualifications would obtain. This provision was introduced to discourage firms from substituting domestic labor with low-cost foreign labor, while allowing them to access specialized labor not easily available domestically.

time & material (T&M) contract, the vendor provides services at a pre-negotiated rate for every person-hour of effort expended on the project and receives payment either at the end of the project or at periodic intervals when project milestones are reached. Here, although a vendor is usually protected against any cost and schedule overruns that may arise due to changes in client specifications, there is a concomitant reduction in incentives for the vendor to execute more efficiently.

In the early years, most Indian software firms preferred T&M contracts. Since these contracts were behavioral, the client needed a safeguard that the Indian firm was not billing them more person-hours than was necessary to execute the project. So clients typically reduced their risk by negotiating hard on the price per person-hour. From the Indian vendor's standpoint, reduced risk came at the cost of reduced margins. Therefore, for firms that had the capability to manage their projects efficiently and assume the risks, it made economic sense to move to a fixed-price contract that potentially yielded higher margins. In fixed-price contracts, clients often agreed to higher prices because such contracts reduced the need for behavioral monitoring and had strong penalties associated with delays or defects in project completion. Thus a combination of improved project execution and management capabilities and the Indian firms' impetus to improve profit margins led to an increasing preference for fixed-price contracts.

Overall, on the supply side the steady transition from onsite to offshore projects and from T&M to fixed-price contracts meant that an increasing share of project management risk had to be borne by the Indian firms. This meant that the firms had to acquire the requisite capabilities to compete effectively. The following section elaborates what these capabilities were and how they played a critical role in Indian firms' successful evolution.

Capabilities of Indian software services firms

Against the backdrop of the demand- and supply-side economics of the Indian software services industry, two broad sets of capabilities are critical. The first is what we term *client-specific capabilities*. As software firms work with their clients over time they develop several client-specific patterns of interaction that become cost-effective over repeated interactions. For instance, one of the firms we interviewed mentioned that clients tend to have fairly idiosyncratic ways of doing things and it takes some time to understand and appreciate this. One of this firm's clients wanted a team of its employees to be stationed at the client site for 3 months after completion of the project. Initially the Indian firm resisted such a demand since it led to increased costs. However, over a period of time, the firm was able to convince the client that it could provide the same quality of after-sales service with the support team based in India. This was possible because with repeated interaction with the same client over time software vendors not only develop a better understanding of the information infrastructure at the client site but also develop better clarity of how the software relates to the client's business environment. Such knowledge is acquired through repeated interactions with the client over various stages of the development cycle such as requirements specification, business process design, data preparation, software installation, debugging, and testing. Hence long-term relationships and repeated interactions with clients resulted in client-specific learning that had a positive effect on both revenues and costs.

On the revenue side, clients were more willing to agree to higher prices on repeat projects as they developed confidence in the capability of the Indian firm to execute projects per their specifications. We reckon that Indian firms were able to capture at least some of the switching costs faced by the client in finding a new vendor and building a new working relationship. On the cost side, there was tangible cost reduction associated with working with a client over time. For instance, one of our interviewees pointed to a client who never specified requirements very clearly at the outset and tended to repeatedly come back and ask for new features after the project was delivered. While this tended to be disruptive initially and created problems for the Indian firm, over time it learned to work around this. Rather than deliver very finished projects, the vendor firm began to involve the client firm at the prototype stage itself and then started building in the client firm's needs as the project went along. In this manner, they were able to avoid the post-delivery negotiations and completion delays and the associated costs. Such client-specific tailoring of projects also enhances the software firm's understanding of the client's business domain. Thus, repeat projects for clients helped develop important client-specific capabilities that contributed to higher profits by reducing

costs. Therefore, we hypothesize:

> *Hypothesis 1: Development of client-specific capabilities based on repeated interaction with clients is positively related to project performance.*

A second capability that is more fungible across clients and industry domains is the software development and project management capability (Humphrey, 1989; Jalote, 1997). The following three capabilities are particularly important. (i) *Software design and building capabilities*: First, vendors must have the capability to understand the requirements of the client and design an appropriate system or architecture to address them. Second, they must possess the capability to efficiently and effectively build the code in conformance to the design and coordinate the entire code development process that is usually distributed across many teams and/or sites. These capabilities are usually reflected in the defects identified in the product/software during the design and development process. (ii) *Effort estimation and management capabilities*: Vendors have to be skilled not only in accurately assessing the requirements of the client but also in assessing the resource inputs or effort required to build and execute the project. They need to be able to identify appropriate resources (for instance, people with the necessary skill, experience, availability, etc.) and create and use prior experience/data to arrive at accurate estimates of the resource/effort requirements. Further, they also require skills to ensure the effective management and deployment of the required resources. Poor capabilities in effort estimation and management are usually reflected in increased manpower cost and/or effort overrun. (iii) *Schedule estimation and management capabilities*: Once companies have a tentative idea of the resource inputs necessary to build and implement the project, they must be able to correctly estimate the duration and schedule for completing the project. They also need to possess the management skills to ensure that the project resources are garnered, deployed and managed to complete the project within the planned schedule. Again, poor capabilities on this dimension are reflected in project completion delays and schedule slippages.

Given the importance of the above capabilities, in recent years firms have placed a great deal of emphasis on software engineering and project management and have been making investments in improving their processes and capabilities. The Capability Maturity Model (CMM) developed by the Software Engineering Institute (SEI) at Carnegie Mellon University is a widely adopted framework to improve software capabilities. Built on theories of quality and continuous process improvement, the CMM was first initiated to provide the Department of Defense a standard means for measuring contractor capability via its definition of process maturity (Humphrey, 1989). As the CMM became widely adopted as a standard quality process model in the defense sector, commercial organizations also began to investigate whether they could benefit from the approach to software process improvement expressed in the CMM. In the last few years software process improvement based on the CMM has emerged as an integrated solution to the software problems in various corporations and empirical evidence in support of the same has been reported (Herbsleb *et al*., 1997; Krishnan, 1996).

Recognizing the importance of project management capabilities, several Indian firms have been the leaders in adopting CMM guidelines to improve their software development processes. According to SEI, in 2001, of the 42 companies worldwide certified as having attained a level-5 capability, 25 are based out of India. Meeting these guidelines is not a trivial task. Firms need to make substantial investments in firm infrastructure, systems, and human capital. The CMM specifies five maturity levels, each consisting of several key process areas (KPAs), to assess an organization's process capability by measuring the degree to which processes are defined and managed (see Paulk *et al*., 1993: Figure 1). Firms first need to do a detailed comparison of their development and project management processes with the CMM process quality framework and identify specific aspects that need to be improved. After making these organizational and process changes, software firms need to set up a rigorous metrics program to collect data to assess various aspects of their development process and institute audit systems to track non-conformance of best practices, process deviations, and exceptions. Measurement-based feedback to improve process capability is achieved through monitoring and review forums to track improvements and make required changes on a regular basis.

In addition, software firms need to invest in training programs to improve their process maturity under the CMM framework. The training activities mandated by CMM include ongoing training in both new technology and software process and project management skills for software developers and managers. To achieve higher levels of process maturity, firms need to organize rigorous training of all their employees not only with a view to improve their development/project management practices but also to institutionalize the entire capability improvement initiative (Paulk *et al*., 1993).

The CMM framework offers generic guidelines for institutionalizing disciplined practices across various activities in software development. Since the nature of software development is such that software teams can quickly turn to ad hoc practices, institutionalization of disciplined practices and continuous feedback-based improvements can evolve as a core project management capability of the firm. Overall, firms can eventually develop their software development and project management capabilities as a result of cumulative and integrated effort across the many process areas of CMM as described above. Eventually the possession of these capabilities should lead to better project-level performance for the firm. Past studies document the relationship between software and project management capabilities and processes and the quality of the products/services provided as well as the efficiency in providing them (Herbsleb *et al*., 1997; Krishnan, 1996). Eventually, we believe that these benefits should result in improved project performance in terms of profitability. Therefore, we hypothesize:

Hypothesis 2: Higher levels of project management capabilities will lead to higher levels of project performance

While client-specific capabilities and project management capabilities are both positively related to project performance, they might differ in the marginal returns they generate. Even within the set of project management capabilities, it is possible to observe some differences in their marginal returns. As argued earlier, differences in marginal returns may arise due to differences in the costs associated with developing these respective capabilities. Our empirical analysis can help shed light on whether that is indeed the case in our setting.

To summarize the paper so far: We have sought to understand the origin, significance, and value of firm capabilities. We first argued that capabilities reflect the distinctive deployment of resources and also that they are contextually grounded. We then elaborated the specific capabilities that are important in the software services industry, namely, *client-specific capabilities* that arise in the context of repeated interactions with clients over time and project *management capabilities* that develop through deliberate and persistent investments in the effort and infrastructure to create them. While client-specific capabilities help reduce project execution costs, project management capabilities help maintain low cost and high quality all along the software development and management value chain. These capabilities, in turn, provide opportunities for rent generation for firms. We next turn our attention to the significance and value of these capabilities as a matter for empirical investigation.

METHODS

Data

We obtained detailed quantitative data at the project level from a leading world-class software services firm headquartered in India.[5] Over 90 percent of its revenues are export-based, of which more than 60 percent comes from North American clients. The dataset includes information on revenues, cost, factor inputs, capability measures, and various project characteristics such as size, client industry, development platform, etc., measured at the project level.

We have data on 227 projects executed by the firm over a 6-year period from 1996 to 2001. However, after dropping single projects and projects with missing data, our sample was reduced to 138 projects.[6] The firm has executed these 138 projects for 57 different clients over the study period. Our dataset has 22 new clients for whom the firm executed its first project during the study period and

[5] Since the dataset reveals the economics of the firm and has competitive implications, we are unable to reveal the identity of the firm.

[6] We found no statistically significant differences between projects with complete data and those with missing data for those variables for which we had complete information. This increased our confidence that the analyzed data were not systematically different from the data on all the projects. The results reported in the paper are estimated using the sample of 138 projects.

35 repeat clients for whom the first project was executed prior to the study period. The average number of projects executed per client is 2.385; the range is from a minimum of 2 to a maximum of 12 projects. Thus, our dataset comprises a panel of 57 clients for whom the firm has executed two or more projects during the study period.

Model specification and estimation

We assume that the firm produces a single output, software services, using skilled labor. Since the software services business is highly labor-intensive, the scale of the firm's operations and its costs depend almost exclusively on manpower. The firm's profit function from each project is given by

$$\pi_{ijt} = F(P, C(\delta, \kappa))$$

where, π_{ijt} are the profits from project i for client j in year t. This recognizes that profitability can vary as a function of time, client characteristics, and project characteristics. P are the prices and C are the costs respectively of project i for client j in year t. Both prices and costs depend on project-specific characteristics, δ, such as size, complexity, duration, and type of contract. The last term in the equation, κ, captures the capabilities measures, which vary with client and time and also depend on project-specific characteristics δ.

This reflects our hypotheses that capabilities tend to influence both the prices and costs of software services. As argued in the previous section, capabilities allow the firm to command a price premium (shift the demand curve), and also reduce costs (shift the supply curve).

We estimated the following equation using project-level data:

$$\log(\pi_{ijt}) = \beta \log(w_{ijt}) + \chi z_{ijt} + \gamma \kappa_{ijt} + \varepsilon$$

where the dependent variable, π_{ijt}, is project-level contribution (revenue minus cost), w_{ijt} is the vector of input characteristics that determine costs, z_{ijt} is a vector of project specific controls, and κ_{ijt} reflect the capability measures. The coefficients on the logged variables are directly interpreted as elasticities, while the coefficients on variables measured in levels vary with the magnitude of the variables.[7] We only logged independent variables that did not have zero values and retained variables with zero values as levels to avoid estimation difficulties.

The equation above cannot be estimated using simple OLS since there are multiple projects per client. Any unobserved relationships among projects for a given client and the consequent heterogeneity across clients can contribute to heteroskedasticity and bias the standard errors of the coefficient estimates. The standard solution to this problem is either to use a GLS estimator and correct for panel heteroskedasticity or employ OLS and account for the correlations within each panel (Wooldridge, 2002). In using OLS to estimate equations with panel data such as that employed in this paper, the usual practice is to employ a fixed-effects specification[8] (Greene, 1997). The fixed-effects model involves parameterizing the client-specific effects by including dummies for each client. We report the results of the estimation using, primarily, a fixed-effects specification. As a robustness check, we also report GLS with panel heteroskedasticity-corrected standard errors.

In the fixed-effects models the coefficient estimates are interpreted as the amount of within-panel variation in the dependent variable (project performance) that is explained by the within-panel variation in the independent variables. Thus, the regression analysis relates changes in project contribution across different projects for a given client to changes in the independent variables after controlling for unobserved time-invariant effects, and the included controls.

Measures

Dependent variable

Project contribution. The dependent variable is project contribution (revenues minus costs) measured in Indian rupees (INR) recognized on the date of completion of the project. We chose to

[7] We explain our rationale for logging the dependent variable in the following section. Entering the variables in levels does not change the results. We report the results on the logged variables for ease of interpretation.

[8] We performed a Hausman test to choose between the random-effects and fixed-effects models. The Hausman test on the full model yielded a chi-squared test statistic equal to 36.22 ($p \leq 0.03$), suggesting that the fixed-effects model is preferred over the random-effects model.

focus on the project level because the firm capabilities that we study primarily exist and evolve at the project level within firms. However, firm profitability is essentially an aggregate of project-level contribution so we expect the correspondence between project and firm profitability to be high. Project profitability is an imperfect indicator of firm profitability to the extent that it does not account for firm-level cost overheads. To be able to use this measure meaningfully we need to adjust the INR values both for inflation and INR–USD (U.S. dollar) exchange rate depreciation over time.[9] As we discovered, this is a far from simple problem, since identifying an appropriate price index specific to an export-based software services firm is quite difficult. Fortunately, we found that a simple solution is to take logs of the dependent variable and include year dummies in the regression estimation. This helps remove the effect of the adjustment factor from the parameters being estimated.[10]

Independent variables

Client-specific capabilities. As outlined earlier, client-specific capabilities are a function of repeated interactions with a given client. These repeated interactions may be a function of time (on a long project) or spread over several projects. We measured it in two ways. For each project in our dataset, we have information on whether the client is new (i.e., the first project executed for that client) or a repeat client (i.e., the firm has executed projects for that client in the past). For each project in the dataset, we have a dummy variable, *customer type*, coded 0 if the firm has executed projects for the client in the past and coded 1 if it is the first project executed for the client. Repeat clients are a proxy for client-specific capabilities developed by the firm. We expect the sign on this coefficient will be negative. Since we also have panel data on each client in our sample (i.e., data on multiple projects done for each client), we used it to estimate a *client fixed effect by including a dummy variable for each client*. Though both measures are largely substitutes, the first measure captures some information about past projects not included in our dataset, and the second produces a within-sample estimate of the client-specific effect. Since we cannot estimate a single parameter for the client fixed effects, we only examine whether they are jointly significant after controlling for input and project characteristics and time.

Project management capabilities. Project capabilities are not client-specific. They are capabilities that can, by definition, be leveraged across clients, industry domains, and development platforms. We used three metric variables to measure project management capabilities. The first variable measures the *number of in-process defects* identified during the project execution phase. Since in-process defects can vary by size of the project, we normalized it by a project size measure (FP) described below.[11] In-process defects, which measure the defects detected in the product, reflect a firm's software development and management capabilities. Further, there is evidence that the cost of fixing defects in the early phases of software development can be substantially lower than the cost of fixing them in the final stages of software development (Jones, 1997). Hence we expect that lower in-process defects will lead to higher project contribution.

A second variable measures *effort overrun*, i.e., difference between actual person-months required to complete the project and person-months that were initially estimated. Effort overrun can affect project profitability since project costs go up as effort increases. This measure reflects effort estimation and management capability. Effective project management involves minimizing such overruns. Since effort overrun is likely to vary directly with budgeted person-months, we normalized effort overrun by the budgeted person-months to estimate its main effect. The expected sign on this coefficient is negative, i.e., higher effort overrun will lead to lower project contribution. It was

[9] The exchange rate depreciation over time is important since the firm incurs most of its costs (e.g., wages) in Indian rupees whereas its revenues are in U.S. dollars. Therefore, changes in the rupee–dollar exchange rates will also change the coefficient estimates in the firm profit function.

[10] For example, C2000 (contribution in year 2000) needs to be adjusted to the same units as C1996 (contribution in 1996). Usually, C2000 is adjusted as C2000/P1996, where the denominator (P1996) is the appropriate index for adjustment. Taking logs, we obtain log(C2000)—log(P1996). Thus if we take logs on the dependent variable and include year dummies, the parameters are free of the adjustment factor. The year dummies absorb the adjustment factor, and since the year dummies are only controls the main results and our interpretation remain unaffected.

[11] The measure of project size, called function points (FP), is a composite measure of project size and complexity and is described in greater detail in the subsection below on the control variables included in the estimation.

entered as levels (rather than logs) since zero or negative values are possible.

A third variable measures the *extent of schedule slippage*, i.e., delay in project completion date. It reflects schedule estimation and management capability. Delays in project completion can adversely affect profitability since the firm can incur contractual penalties for delays and also bear increased labor costs. We expect the magnitude of schedule slippage to vary directly with expected project duration. To control for this, we normalized schedule slippage measured in days by project duration (also measured in days). The expected sign on this coefficient is negative, i.e., higher schedule slippage will lead to lower project contribution.

In sum, the capabilities measures employed here reflect our view that capabilities are distinct from either inputs or resources. While inputs or resources such as capital or labor are accessible by all firms at prevailing factor prices, capabilities reflect the deployment of resources (Makadok, 2001). Therefore, capability differences between firms are reflected in productivity differences between them, and within firms in productivity improvement over time. Thus, changes in the capability measures reflect changes in the productivity of resources over time.

Control variables

We controlled for a variety of other variables that might impact project profitability. We briefly describe these variables below.

Contract type. There are two types of contracts commonly used in the Indian software services industry: time & material (T&M) and fixed price. As explained earlier, the role of project management capabilities may differ in the two types of contracts. Also, these contracts may vary in terms of their influence on project profitability. To control for the obvious effect of contract type on profitability we included a dummy variable, where T&M contract was coded 0 and fixed price was coded 1. In terms of the risk–return trade-off fixed price projects are expected to yield superior project profitability, suggesting an expectation of a positive sign on the coefficient.

Project size and complexity. We expect project size to affect profitability. Though we have no priors on this, we might reasonably expect that the size–profitability relationship might be increasing below the minimum efficient scale and decreasing above the maximum efficient scale. We employed a measure of project size, called function points (FP), to reflect a composite measure of project size and complexity.[12] We logged the FP measure in our estimation and, as a result, the coefficient can be interpreted as elasticity with respect to project profitability.

Team size. We also expect team size to have an effect on project profitability. As above, we expect project profitability will increase with team size when the team is too small and overworked and decrease when the team size is large enough to create coordination problems. Team size is logged in our model and can be interpreted as elasticity with respect to profitability.

Person-months. The team size measure is imperfect since there may be attrition[13] during the project or some members may work only part-time. This will cause the team size measure to be overstated. A more precise measure of project size is person-months of labor. This accounts for both team member attrition and part-time manpower usage. Person-months are entered as logs and can be interpreted as elasticity with respect to profitability.

Project duration. Duration is an important control variable since longer projects are more prone to cost overruns, either due to forecasting difficulties or employee attrition. We measure project duration as the actual months taken for project completion. This measure is entered in logs and can also be interpreted as elasticity with respect to project profitability.

Industry domains. The vendor firm we studied executed projects for clients in multiple industries. Since competition and appropriability conditions can vary by industry and the vendor's capabilities may not be entirely fungible across industry

[12] Earlier measures of size and complexity tended to just count the number of lines of code in software and use it as a proxy. The problem with this measure is that the number of lines of code for a given application varies directly by software platform. For example, for a given application, the number of lines of code in Cobol is several times greater than the number of lines of code in Java. The FP measure is designed to be independent of programming language (see Albrecht and Gaffney, 1983).

[13] Employee attrition is historically very high in this industry, ranging from 10 to 30 percent per year.

domains, we include industry dummies to control for industry-specific differences. We included three dummy variables for financial services, manufacturing, and marketing industries. The omitted category was 'other.'

Development platforms. Profitability can also depend on the software platform on which the software is coded. The vendor is likely to face lower costs on platforms on which it has executed a number of prior projects and is likely to incur substantial learning and start-up costs on newer platforms. To account for this, we included four dummy variables to distinguish between Windows NT, Mainframe, Unix, and Web-based platforms. The omitted category was 'other.'

Time. We control for time in all our models using year dummies. The year dummies reflect the particular year in which each project was started. (We also tested models using year dummies based on project completion year and found that the results did not change substantively.) This control is necessary to capture the variance in the dependent variable due to exchange rate fluctuations and inflation. The time dummies are also important to account for any variation in the proportion of onsite and offshore projects that the firm has done over time. No projects in our dataset were exclusively onsite or offshore. Unfortunately, the firm did not have data at the project level on the proportion of work done onsite and offshore. The firm confirmed that there were no dramatic differences across projects on the mix of onsite and offshore work. However, there has been some change in this mix over time. Thus including the year dummies also helps control for the change in the proportion of onsite and offshore work over time.

RESULTS

Table 1 presents the descriptive statistics and correlation matrix of the variables employed in the estimation. From the correlation matrix it is clear that all the control variables are significantly positively related to project contribution. We also note that the relationship between the capability measures and project contribution is negative as expected, though only the process defects variable is statistically significant.

Table 2 presents the coefficient estimates from the fixed-effects panel regression analysis. The second column lists the predicted sign on the key independent variables in the model. The column labeled Model 1 presents results with just the control variables in the model. The R^2 for this model is highly significant and the control variables account for about 66 percent of the variance in the data.[14] The model also includes the client fixed effects. As expected, the client fixed effect is strongly significant, providing some prima facie evidence for client-specific learning and/or switching cost associated with repeat projects for a given client.[15]

In Model 2A we added the customer type variable to the analysis. The model is again highly significant and its adjusted R^2 increases to about 0.67. The customer type measure sought to capture aspects of client-specific capabilities, such as specialized investments and learning from repeated interactions with the client.[16] Though the sign on the coefficient was in the expected direction, it was not statistically significant. Note, however, that the client fixed effect continues to be highly significant even in Model 2A.

To investigate the possibility that the client fixed effect might be capturing variance in the customer type variable, we re-estimated the model by dropping the client fixed effect. Since dropping the fixed effect can bias the estimates, we employed a generalized least squares (GLS) model that allowed us to correct for heteroskedasticity within projects executed for a given client. These results, presented in Model 2B, confirm our conjecture. The coefficient on the customer type variable is negative and statistically significant, suggesting that project contribution, on average, is lower for new clients as compared with repeat clients. This provides some confidence that client-specific capabilities are related to project contribution.

[14] We tested for decreasing returns to project and team size by introducing their quadratic powers. The coefficients were non-significant, suggesting that the constant returns to scale assumption might be reasonable for this dataset.

[15] To check for multicollinearity, we computed the variance inflation factor (VIF) indices and found that the person–months variable had a VIF of 4.42, followed by team size (2.97), size FP (2.46), and duration (2.36). The mean VIF for all independent variables was 2.51, alleviating multicollinearity concerns (Chatterjee, Hadi, and Price, 2000).

[16] In the dataset used for empirical estimation there is no evidence that revenues from repeat clients were systematically higher than that of first-time projects after controlling for project characteristics. Revenues tended to remain constant after adjusting for inflation and exchange rate changes.

Table 1. Descriptive statistics and correlation matrix of variables ($N = 138$)

Variables	Mean	S.D.	1	2	3	4	5	6	7	8	9	10	11	12
Log (Contribution)	4.875	1.314												
Log (Size)	6.458	1.491	0.489*											
Log (Team size)	2.117	0.668	0.609*	0.679*										
Log (Person-months)	3.077	1.041	0.675*	0.670*	0.741*									
Log (Duration)	4.884	0.601	0.459*	0.536*	0.470*	0.639*								
Customer type	0.159	0.367	0.100	0.169*	0.177*	0.235*	0.195*							
Process defects	0.630	1.980	−0.319*	−0.638*	−0.302*	−0.411*	−0.276*	−0.117						
Schedule slippage	0.018	0.093	−0.138	0.026	0.034	−0.052	−0.045	0.087	−0.028					
Effort overrun	0.050	1.174	−0.077	−0.037	0.033	0.043	−0.002	0.148*	0.006	−0.142				
Contract type	0.609	0.490	−0.254*	−0.080	−0.117	−0.059	−0.038	0.023	0.115	−0.102	−0.138			
Year	1999.050	1.479	0.120	−0.081	0.013	−0.156*	−0.249*	−0.375*	0.113	−0.140*	0.043	−0.085		
Effort overrun ∗ Y2001	—	—	0.089	0.009	0.056	0.081	0.031	0.028	0.007	−0.195*	0.544*	−0.083	−0.026	
Schedule slippage ∗ Y1996	—	—	0.027	−0.044	−0.006	0.077	−0.031	0.136*	0.034	0.353*	0.052	−0.080	−0.235*	0.006

Asterisks indicate coefficients significant at 0.05 level or lower.

Table 2. Regression estimates ($N = 138$)

Independent variables	Pred. sign	Dependent variable: log (Contribution)					
		Model 1	Model 2A	Model 2B	Model 3A	Model 3B	Model 4
Customer type	—		−0.841	−0.154**	−0.687	−0.118†	−0.108
			(0.538)	(0.055)	(0.531)	(0.07)	(0.191)
Process defects	—				−0.037	−0.049**	−0.038
					(−0.046)	(−0.014)	(−0.042)
Schedule slippage	—				−3.493*	−2.734**	−4.790***
					(1.582)	(0.486)	−0.823
Effort overrun	—				−0.197*	−0.166*	−0.422***
					(0.087)	(0.037)	(0.094)
Schedule slippage ∗ Y1996	—						7.355***
							(1.873)
Effort overrun ∗ Y2001	+						0.297**
							(0.118)
Controls							
Contract type		−0.309	−0.289	−0.463***	−0.321†	−0.538***	−0.512***
		(0.202)	(0.201)	(0.071)	(0.193)	(0.071)	(0.14)
Project size (FP)[a]		0.108	0.103	0.059	0.092	0.012	0.119
		(0.096)	(0.095)	(0.037)	(0.111)	(0.037)	(0.083)
Team size[a]		0.057	0.074	0.041	0.055	0.237	0.118
		(0.272)	(0.282)	(0.088)	(0.281)	(0.085)	(0.189)
Person months[a]		0.675***	0.602***	0.731***	0.517***	0.598***	0.477***
		(0.175)	(0.179)	(0.071)	(0.179)	(0.062)	(0.137)
Duration[a]		0.233	0.258	0.198*	0.363†	0.250**	0.278†
		(0.23)	(0.228)	(0.099)	(0.221)	(0.079)	(0.167)
Domain controls		Sig.	Sig.	Sig.	Sig.	Sig.	Sig.
Platform controls		Sig.	Sig.	Sig.	Sig.	Sig.	Sig.
Year effects		Sig.	Sig.	Sig.	Sig.	Sig.	Sig.
Constant		1.459	1.033	1.985***	0.976	1.712***	1.550*
		(1.063)	(0.982)	(0.366)	(0.981)	(0.291)	(0.728)
Adjusted R^2		0.66	0.672	—	0.704	—	0.719
N		138	138	138	138	138	138
F-value		11.04***	10.51***	—	9.50***	—	8.82***
Wald Chi-Sq.		—	—	1641.72***	—	2062.54***	—
F-test for client fixed effects		2.06***	1.82***	—	1.78**	—	1.72**

[a] Variables are logged. *** $p \leq 0.001$; ** $p \leq 0.01$; * $p \leq 0.05$; † $p \leq 0.10$. Standard errors are reported in parentheses.

In Model 3A, we included variables for the three project management capabilities. The overall model continues to be significant and accounts for about 70 percent of variance in the data. We find that schedule slippage and effort overrun respectively are significantly negatively related to project contribution, suggesting that improvement in project management capabilities is rent-generating at the project level. The coefficient for in-process defects is not statistically significant though its sign is in the expected direction.[17] We explored this result in discussions with project managers. They reasoned that the identification of defects during project execution is indeed a capability since defects can be fixed at lower cost during execution than if they were to be fixed after the project was completed. In other words, higher values on the in-process defects measure might indicate better project management skills. On the flip side, they also pointed out that an increase in the number of in-process defects tends to disrupt delivery schedules and negatively impact project profitability. In sum, it seems that while in-process defects do reflect detection capability, they also expose some weakness in software design and execution capabilities; namely why these defects arose in the first place. In the long run, we expect that defect detection capability might be positively

[17] One reviewer suggested that we enter in-process defects as a binary variable given that the modal value is zero. When entered as a binary (1 or 0) variable, in-process defects was marginally significant (at the 10% level).

related to project contribution, especially with repeat projects. However, the need to fix defects during project execution can disrupt project schedules and increase project costs, leading to the observed result.

In Model 3A, we also find that the customer type variable remains non-significant when the project management capabilities measures are included along with the client fixed effects. Therefore, we again re-estimated the model by dropping the client fixed effects. Model 3B reports the results of the GLS estimation with the correction for panel heteroskedasticity. Not surprisingly, customer type turns significant, though only marginally ($p \leq 0.09$). The project management capabilities variables, however, continue to be robustly significant in the predicted direction. This result seems to suggest that the project management capabilities variables share some common variance with the client-specific capabilities measure. This runs counter to our hypothesis that they are each distinct sets of capabilities. We discussed this result with industry experts, who suggested that by working with a given client over time (i.e., repeat projects) the vendor firm gains a better understanding of the client's needs and expectations. This, in turn, can help reduce effort overrun and schedule slippage accounting for the observed result. The managers in the software firm we studied tended to agree with this possible interpretation. Finally, we also performed a joint significance test of the project management capabilities. We found that the three variables were jointly significant ($F = 3.23$; $p \leq 0.01$), reconfirming the independent, additional significance of these variables in influencing project contribution.

Models 3A and 3B suggest that decreases in schedule slippage and effort overrun respectively contribute to increases in project contribution. We further analyzed whether and how project management capabilities evolved over time and how that affects contribution.[18] We included an interaction term of schedule slippage with 1996 (the first year of data) and effort overrun with 2001 (the last year of data).[19] These results are reported in Model 4. The overall model continues to be highly significant, accounting for about 72 percent of the variance in the data. As expected, we find that the extent of effort overrun on projects is declining over the 1996–2001 period and this decline is positively related to project performance. Contrary to expectation, we found that schedule slippage increased marginally during the 1996–2001 period, which adversely affected project contribution in the later years.

To understand this result, we examined the data more closely and found three possible reasons that might explain it. First, the vendor steadily increased the number of fixed price projects over the years and did not manage this transition well. As observed in our data, fixed price projects exhibit greater schedule slippage than T&M projects. This is also corroborated in the regression results, where we find T&M projects, contrary to expectations, to be more profitable than fixed price projects. Second, in the fixed price projects we found a greater difference between the team size and person-months measures. While team size reflects the maximum number of persons who worked on a given project, the person-months measure captures actual labor input in the project. At one extreme, if all members of the team worked full time on a given project the team size and person-months would be identical. The divergence in the two measures appears when there is high turnover in the project teams. We found that the difference between the two measures increased over the years, suggesting that an increase in project team turnover might account for the increase in schedule slippage. During the 1996–2001 period, the attrition due to employee resignations fluctuated between 9 and 16 percent. Increase in turnover directly contributes to schedule slippages since there is a set-up cost when new employees enter a project mid-way. Our discussions with company executives revealed a third reason for increase in schedule slippage:

[18] Ideally we would have liked to examine the evolution of client-specific capability over time. It was not possible since client-specific capability is measured as a dummy variable, thus precluding an interaction term with time.

[19] We did not include an interaction term with capabilities and time as a continuous measure since this implies the assumption that capabilities change linearly over time. We found no theoretical or empirical basis for such an assumption. Our results, however, are robust to including year as a continuous variable. However, we were unable to include both interactions in the same model. The high correlations between the two interaction terms created estimation difficulties. Similarly, including an interaction effect of both capability measures with the same year (i.e., 1996 or 2001) created estimation problems since they were highly correlated. We switched the years to avoid this problem. For completeness, an interaction of schedule slippage with 2001 is negative and significant, which is consistent with our interpretation.

an increase in e-commerce-related projects in the post-1996 period. The e-commerce clients, in their effort to speedily set up their Web infrastructure, set aggressive project completion schedules that were difficult for the company to meet. One or more of these three explanations account for the observed increase in schedule slippage that in turn contributed to reduced project profitability.

DISCUSSION

We sought to examine two interrelated research questions in this paper: Where do capabilities come from and how do they affect firm performance? We made three main arguments in addressing these questions. First, we suggested that capabilities involve the deployment of resources and they evolve over time through the joint effects of deliberate, persistent firm-specific investments and learning-by-doing. Second, we proposed that capabilities are context-specific and, therefore, we need to conceptualize and study them accordingly. In this paper, we looked at the software services industry and identified two important capabilities at the project level: client-specific capabilities and project management capabilities. Third, we argued that improvement in capabilities will result in improved project profitability and that different capabilities yield different marginal benefits.

Our results are broadly supportive of our hypotheses. In the software services industry we find that capabilities contribute positively to project performance. The empirical evidence for the importance of client-specific capabilities was only modest. In models that included client fixed effects we did not find support for the effect of client-specific capabilities on contribution. However, in models that did not include the client fixed effect, we found that projects for new clients, on average, yield approximately 2 percent lower contribution than projects for repeat clients. In models including the client fixed effect, it appears that the dummy variable capturing repeat or new clients is too coarse a measure to adequately reflect client-specific capabilities. An ideal measure of client-specific capabilities is a historical count of all the projects executed for a client (not just a count of projects within the sample period). Unfortunately, the firm was unable to provide these data. Nevertheless, the significance of the client-specific capabilities measure in the GLS model is indicative and might be useful to examine carefully in future research. In addition, we believe that the client fixed effect might be also picking up some variance in the learning associated with repeated interaction with clients (Anand and Khanna, 2000).

Project management capabilities were generally predictive of higher project contribution. Two of the three measures of project management capabilities were statistically significant. A 1 percent increase in schedule slippage resulted in a 0.6 percent decline in project contribution.[20] A 1 percent increase in effort overrun causes a 0.1 percent decline in project contribution. It seems that the effect of effort overruns just increases labor cost and causes less damage to project contribution, whereas increases in schedule slippage triggers two independent and additive increases in cost: labor costs and contractual penalties for late completion. Our results suggest that although both types of capabilities, namely schedule estimation and management, and effort estimation are significantly related to performance, the former makes a higher marginal contribution to performance. Therefore, our results suggest that firms may be better off erring on the side of caution and overstaffing their project teams rather than facing the prospect of a schedule slippage.

Finally, we found some evidence for the evolution of capabilities over time and its impact on project contribution. We found that a tighter control of effort overrun in projects in 2001 resulted in better project contribution as compared with projects in 1996–2000. On the other hand, we found that an increase in schedule slippage from 1997 to 2001 has had a significant negative effect on project contribution.[21] This raises the value of improving the capability to better forecast schedules and stick to them.

We believe our study of firm capabilities and their effects on performance in the software services industry raises several important issues. First, our study has sought to make the case that identifying the capabilities that are the sources of performance differences need to be contextually

[20] The semi-elasticities for the coefficients on the capabilities measures were computed as $m_i = \partial F(X, \beta)/\partial X_i \beta_i$ with all the variables fixed at their means.

[21] We avoid calculating semi-elasticities for the interaction terms. The typically high correlations between the main effect and the interaction effect result in imprecisely estimated coefficients. As a result, computing accurate economic significance of these coefficients becomes difficult.

grounded. Each industry is driven by its own demand- and supply-side economics, which also changes over time. It is important to take this into account while identifying and measuring capabilities. It seems that there are indeed significant differences in the way the same firm deploys its resources, both across projects and over time. Holding project inputs and characteristics constant we found that an improvement in project management capabilities resulted in increases in project contribution. Put simply, our results demonstrate that project profitability differences could be a function of differences in the way the same resources are deployed by the firm. Also, we found some evidence that an improvement in the productive deployment of resources over time yielded increases in project profitability. This provides reason to take more seriously Penrose's (1959) key insight that differential firm capabilities in productively deploying resources lie at the heart of firm performance differences.

Second, it appears that measuring capabilities at more micro levels within a firm is quite promising. It not only helps better estimate their economic significance but also provides clear guidelines to the firm on where and how it needs to improve its capabilities. Measures of capabilities at the aggregate firm level, while useful in identifying between-firm differences, provide little understanding of the micro-foundations of such inter-firm differences. As our study demonstrates, measurement of firm capabilities at the micro-level holds promise of enhancing our understanding 'how' and 'why' some firms perform better than others.

Lastly, our analyses also show that the marginal returns to different capabilities are not uniform. Given scarce managerial resources, it is useful for firms first to identify the capabilities that provide the highest marginal returns to performance and then direct the bulk of its resources to acquiring them. Building capabilities requires significant and, often, irreversible commitment of real resources, both financial and managerial. Decisions about which capabilities to acquire or build require due diligence in the analysis of costs and benefits. Moreover, it is likely that different firms will face different costs and benefits of acquiring the same capabilities given the interdependencies with various other organizational choices (Ethiraj and Levinthal, 2004). For instance, for a firm that engages in relatively few repeat projects, the marginal benefit of client-specific capabilities is likely to be significantly less than the marginal benefit of project management capabilities (this would generally be the case for smaller or newer firms who are likely to have fewer repeat clients in their early years). The converse is likely to be true in the case of a firm that engages in a large number of repeat projects. In fact, the latter set of firms could also explore the feasibility of investing in creating deliberate and institutionalized mechanisms to build their client-specific capabilities than leaving it to tacit learning-by-doing. Our interviews seem to support this contention. Finally, in our dataset, if we assume that the marginal cost of acquiring different project management capabilities is the same, our results suggest that the firm would be well advised to expend resources to improve schedule estimation and management capabilities so as to tightly manage schedule slippages. Improvements in this capability promise to yield higher marginal improvement in project contribution performance.

LIMITATIONS AND DIRECTIONS FOR FUTURE RESEARCH

In this paper we attempted to uncover the micro-foundations of capabilities and how they affect performance. Our study, like any study, suffers from some limitations. First, it is based on a single service industry with its own peculiar characteristics. It is not clear to what extent the substantive results of this paper are generalizable across industries. At the same time, as we have asserted above, capabilities are usually context-specific, i.e., industry-specific, and capabilities that are generalizable across industries are likely to be overly abstract and less useful as a guide to managerial action. A second limitation is that our study is based on data from a single firm. Ideally, we would have liked to include data from a few more firms. However, getting access to such detailed data of great competitive significance is a difficult challenge. In our case, this involved several years of data collection, ongoing negotiations with the firm concerned, and the signing of non-disclosure agreements. Third, since our analysis is based on data from only one firm we could not make any explicit inter-firm comparisons of competitive advantage. However, since the firm is among the top five firms in the industry on several dimensions such as growth

and total revenues it gives us some confidence for drawing the capabilities–performance link.

In spite of some of the data limitations outlined above, some of the unique aspects of our dataset outweigh some of these disadvantages. One, the data on resource inputs (team size, experience, etc.) and capabilities allow us to empirically measure Penrose's (1959) key distinction between available resources (factor inputs such as labor) and how they are deployed (i.e., productivity of resources). Two, the data on project-level contribution suffer from relatively fewer problems associated with aggregate accounting data. Finally, our design allowed us to combine the depth and richness of longitudinal, single-firm case studies with the rigor of large-sample empirical estimation.

In conclusion, the paper attempted to take an initial step in teasing out the importance of capabilities and estimating their impact on performance. We hope that the spirit of this paper in advocating the importance of contextually grounded studies of firm capabilities will spur further research along these lines in other industries. The shift in research focus from whether or not capabilities matter to 'what' capabilities matter and 'how' they matter promises to enrich our understanding of firm profitability differences.

ACKNOWLEDGEMENTS

We thank Rich Makadok, Phanish Puranam, and an anonymous referee for helpful comments on an earlier version of this paper. Errors and omissions remain our responsibility. Research funding for this project from the Mack Center for Technological Innovation at the Wharton School and the William Davidson Institute and Mike and Mary Kay Hallman Fellowship at the Michigan Business School is gratefully acknowledged.

REFERENCES

Albrecht AJ, Gaffney JE. 1983. Software function, source lines of code, and development effort prediction: a software science validation. *IEEE Transactions on Software Engineering* **SE-9**(6): 639–648.

Amit R, Schoemaker PJH. 1993. Strategic assets and organizational rent. *Strategic Management Journal* **14**(1): 33–46.

Anand BN, Khanna T. 2000. Do firms learn to create value? The case of alliances. *Strategic Management Journal* **21**(3): 295–315.

Andrews K. 1971. *The concept of Corporate Strategy*. Richard D. Irwin: Homewood, IL.

Arora A, Arunachalam VS, Asundi J, Fernandes R. 2001. The Indian software services industry. *Research Policy* **30**(8): 1267–1287.

Banerjee AV, Duflo E. 2000. Reputation effects and the limits of contracting: a study of the Indian software industry. *Quarterly Journal of Economics* **115**(3): 989–1017.

Barney J. 1991. Firm resources and sustained competitive advantage. *Journal of Management* **17**(1): 99–120.

Brush TH, Artz KW. 1999. Toward a contingent resource-based theory: the impact of information asymmetry on the value of capabilities in veterinary medicine. *Strategic Management Journal* **20**(3): 223–250.

Chatterjee S, Hadi AS, Price B. 2000. *Regression Analysis by Example* (3rd edn). Wiley: New York.

Cohen WM, Levinthal DA. 1990. Absorptive capacity: a new perspective on learning and innovation. *Administrative Science Quarterly* **35**(1): 128–152.

Collis DJ. 1996. Organizational capability as a source of profit. In *Organizational Learning and Competitive Advantage*, Moingeon B, Edmondson A (eds). Sage: London; 139–163.

Dierickx I, Cool K. 1989. Asset stock accumulation and sustainability of competitive advantage. *Management Science* **35**(12): 1504–1512.

Dosi G, Nelson RR, Winter SG. 2000. Introduction. In *The Nature and Dynamics of Organizational Capabilities*, Dosi G, Nelson RR, Winter SG (eds). Oxford University Press: New York; 1–22.

Ethiraj S, Levinthal DA. 2004. Modularity and innovation in complex systems. *Management Science* **50**(2): 159–173.

Gopal A, Sivaramakrishnan K, Krishnan MS, Mukhopadhyay T. 2003. Determinants of contract choice in offshore software development. *Management Science* **49**(12): 1671–1683.

Grant RM. 1991. The resource-based theory of competitive advantage: implications for strategy formulation. *California Management Review* **33**(3): 114–135.

Greene WH. 1997. *Econometric Analysis* (3rd edn). Prentice-Hall: Upper Saddle River, NJ.

Helfat CE. 1994. Evolutionary trajectories in petroleum firm R&D. *Management Science* **40**(12): 1720–1747.

Henderson RM, Cockburn I. 1994. Measuring competence? Exploring firm effects in pharmaceutical research. *Strategic Management Journal*, Winter Special Issue **15**: 63–84.

Herbsleb J, Zubrow D, Goldenson D, Hayes W, Paulk M. 1997. Software quality and the capability maturity model. *Communications of the ACM* **40**(6): 30–40.

Humphrey WS. 1989. *Managing the Software Process*. Addison-Wesley: Cambridge, MA.

Iansiti M, Khanna T. 1995. Technological evolution, system architecture and the obsolescence of firm capabilities. *Industrial and Corporate Change* **4**(2): 333–361.

Jalote P. 1997. *An Integrated Approach to Software Engineering*. Springer-Verlag: New York.

Jones C. 1997. *Software Quality: Analysis and Guidelines for Success*. International Thomson Computer Press: Boston, MA.

Kale P, Dyer JH, Singh H. 2002. Alliance capability, stock market returns and long-term alliance function: the role of the alliance function. *Strategic Management Journal* **23**(8): 747–767.

Klein B, Crawford RG, Alchian AA. 1978. Vertical integration, appropriable rents, and the competitive contracting process. *Journal of Law and Economics* **21**(2): 297–326.

Krishnan MS. 1996. Cost and quality considerations in software product management. Doctoral dissertation, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA.

Langlois RN. 1992. Transaction cost economics in real time. *Industrial and Corporate Change* **1**(1): 99–127.

Levinthal DA. 1997. Adaptation on rugged landscapes. *Management Science* **43**(7): 934–950.

Makadok R. 2001. Toward a synthesis of the resource-based and dynamic-capability views of rent creation. *Strategic Management Journal* **22**(5): 387–401.

Makadok R, Walker G. 2000. Identifying a distinctive competence: forecasting ability in the money fund industry. *Strategic Management Journal* **21**(8): 853–864.

Market-Guide. 2002. Software and programming companies list. http://biz.yahoo.com/p/softwrttmd.html.

McGrath RG, MacMillan IC, Venkataraman S. 1995. Defining and developing a competence: a strategic process paradigm. *Strategic Management Journal* **16**(4): 251–275.

Nasscom. 2001. *The Indian Software Industry Report*. Nasscom: New Delhi.

Nelson RR. 1991. Why do firms differ, and how does it matter? *Strategic Management Journal*, Winter Special Issue **12**: 61–74.

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

Paulk MC, Curtis B, Chrissin MB, Weber CV. 1993. Capability Maturity Model, Version 1.1. *IEEE Software* **10**(4): 18–27.

Penrose ET. 1959. *The Theory of the Growth of the Firm*. Wiley: New York.

Peteraf MA. 1993. The cornerstones of competitive advantage: a resource-based view. *Strategic Management Journal* **14**(3): 179–191.

Porter ME. 1991. Towards a dynamic theory of strategy. *Strategic Management Journal*, Winter Special Issue **12**: 95–117.

Rosenbloom RS. 2000. Leadership, capabilities, and technological change: the transformation of NCR in the electronic era. *Strategic Management Journal* **21**(10–11): 1083–1103.

Rumelt RP. 1984. Toward a strategic theory of the firm. In *Competitive Strategic Management*, Lamb RB (ed.). Prentice-Hall: Englewood Cliffs, NJ; 556–570.

Rumelt RP. 1987. Theory, strategy, and entrepreneurship. In *The Competitive Challenge*, Teece D (ed.). Ballinger: Cambridge, MA; 137–158.

Schroeder RG, Bates KA, Junttila MA. 2002. A resource-based view of manufacturing strategy and the relationship to manufacturing performance. *Strategic Management Journal* **23**(2): 105–117.

Selznick P. 1957. *Leadership in Administration: A Sociological Interpretation*. Harper & Row: New York.

Silverman BS. 1999. Technological resources and the direction of corporate diversification: toward an integration of the resource-based view and transaction cost economics. *Management Science* **45**(8): 1109–1124.

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

Whang S. 1995. Market provision of custom software: learning effects and low balling. *Management Science* **41**(8): 1343–1352.

Winter S. 1990. Survival, selection, and inheritance in evolutionary theories of organization. In *Organizational Evolution: New Directions*, Singh JV (ed). Sage: New York; 269–297.

Winter SG. 1995. The four Rs of profitability: rents, resources, routines, and replication. In *Resource-Based and Evolutionary Theories of the Firm: Towards a Synthesis*, Montgomery CA (ed). Kluwer Academic: Boston, MA; 147–178.

Wooldridge JM. 2002. *Econometric Analysis of Panel Data*. MIT Press: Cambridge, MA.

Zollo M, Winter S. 2002. Deliberate learning and the evolution of dynamic capabilities. *Organization Science* **13**(3): 339–351.

