Client-Vendor Relationships in Offshore Applications Development

1 author:

Rajesh Mirani
University of Baltimore

28 PUBLICATIONS   **550** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project

(To be Decided) View project

Client-Vendor Relationships in Offshore Applications Development:
An Evolutionary Framework

Rajesh Mirani, University of Baltimore, USA

ABSTRACT

This article presents an evolutionary framework for the establishment and progression of client-vendor relationships in the context of offshore applications development. It is argued that such a relationship typically begins as a cost-reduction exercise, with the client contracting out simple, structured applications to one or more offshore vendors. Over time, the client assigns increasingly complex applications to selected vendors and cultivates loose, trust-based, networklike relationships with them. As offshore applications continue to evolve and become business-critical, the client may seek to regain control by establishing a command-based hierarchy. This may be achieved through part or full ownership of a vendor organization or by starting a captive offshore subsidiary. Thus, the initial client objective of cost reduction ultimately is displaced by one pertaining to risk control. Pertinent prior research is used to justify the proposed framework. This is followed by a case study that describes how a specialty telecommunications company is pursuing just such an evolutionary path.

Keywords: application development; business applications; case study; client-vendor relationships; contracts; evolution; hierarchies; offshoring; outsourcing of IS; stage theory; networks; risk management

INTRODUCTION

Strategic-level managers such as CEOs, CIOs, and CTOs lately have been under great pressure to seek out fresh approaches to control information technology (IT) costs and to demonstrate higher returns on technology investments. One increasingly popular response from decision makers is offshoring, or the shift of IT work to low-wage, offshore locations. Some offshore work tends to be project-oriented with specific completion criteria that were entailed in the design and development of applications. Other work comprises ongoing operations such as technology maintenance, or help desk support conducted out of call centers. This article focuses on the offshoring of applications development and how its attributes shape the evolution of relationships between client and vendor organizations.

Much has been written about the management of IT outsourcing and offshoring. Proponents of various approaches have based their

arguments on literature ranging from transaction cost economics to resource dependency, strategic choice, stakeholder theory, organizational learning, and institutional theory (Barringer & Harrison, 2000). Some of these suggested approaches also have been tested empirically. However, these management prescriptions neither have been presented nor tested in the specific context of offshore applications development. The implicit assumption underlying this omission appears to be that valid arguments in the general IT outsourcing or offshoring context also must apply to application development, regardless of task attributes. Most theories also embody static views of vendor-client relationships. They emphasize either vendor partnerships/alliances (a network-oriented, trust-based perspective) (Lander, Purvis, & McCray, 2004; Willcocks & Choi, 1995), contracts and transactions (a market-oriented, enforcement-based perspective) (Aubert, Rivard, & Patry, 2004; Richmond & Seidmann, 1993), or a mix of both (Koh, Ang, & Straub, 2004; Sabherwal, 1999). The few that offer evolutionary offshoring perspectives are focused either on client drivers and general industry trends (Carmel & Agarwal, 2002) or on vendor points of view (Rajkumar & Mani, 2001).

It is the contention of this article that for offshore application development, these ostensibly disparate management approaches in fact may represent progressive stages for the client in the evolution of its relationships with one or more vendors. While several alternative evolutionary paths may be feasible, each with its own set of antecedents, process dynamics, and consequences, the discussion here makes the case for one likely path. In this path, a given client organization evolves through successive offshoring stages, not all necessarily with the same application development vendor.

Specifically, a client's experimentation with offshoring begins with reduced development costs as the desired goal. Over time, interplay between the intrinsic task characteristics of application development and the unique attributes of offshoring broaden and evolve the client's goals. These evolving goals, in turn,

successively alter the nature of the vendor-client relationship in distinct stages. The rationale for this path is supported with logical argument, evidence from the published literature, and an original case study for illustration.

STAGES IN OFFSHORE APPLICATIONS DEVELOPMENT

Like all buyers and suppliers, offshore vendors and clients essentially interact in one of two ways: transactional/market exchanges or relational exchanges. A transactional exchange is usually a short-term contract characterized by free market price mechanisms and the need for enforcement. A relational exchange implies a longer-term relationship with ongoing interactions. Relational options include long-term contracts, networks, and hierarchies. Long-term contracts resemble the short-term variety, except that they address many more contingencies and consequences. Networks emphasize interorganizational trust, association, and solidarity. Hierarchies refer to rather rigid structural relationships based on formal authority and command, usually stemming from ownership of one organization by another.

As mentioned earlier, the published literature has focused almost exclusively on contracts and networks as mutually exclusive types of client-vendor relationships in the offshoring/outsourcing context. This article will show that applications development offshoring initiatives usually begin as transactional exchanges but end up in the relational realm. The evolutionary path that client-vendor relationships tend to follow consists of three stages. In the first stage, a client contracts out some application development to an offshore vendor with the objective of reduced costs. Success at this experimental stage may result in short-term contracts being replaced by fewer, long-term ones. Stage two arrives when the changing nature of tasks, together with the relationship attributes, makes it difficult for both client and vendor to continue in a purely contractual vein. The need for closer working ties and greater trust brings about a networklike relationship. With the increasing vendor dependence that this engenders, the importance of the

client's original cost-reduction objective gradually is displaced by a strong desire to manage vendor risk. Paradoxically, therefore, the very success of a network-style relationship causes the original success criteria to give way to new ones. This ushers in stage three. In an effort to regain tighter vendor oversight and control, the client seeks to establish a structural hierarchy with the vendor. Doing so may be impractical if there are multiple vendors or if a structural relationship is infeasible for other reasons, so one option in the third stage is for the client to establish a new captive offshore subsidiary.

Stage 1: Contracts

Labor costs constitute the single major source of applications development expenses. The abundant availability of qualified software engineers in low-wage countries typically drive a client organization's exploratory efforts to seek out an overseas vendor, with the objective of vastly reduced development costs (Matloff, 2004). Preliminary contact with a prospective offshore vendor has the attributes of a highly efficient, classic free market exchange. Both parties, unconstrained by prior contact experiences or future relationship expectations, are focused fully on negotiating the current transaction, with a view to maximizing their respective short-term utilities. Such an exchange tends to yield the best possible outcomes for both client and vendor but only in the presence of perfect information (i.e., complete knowledge about each other's objectives, strengths, and weaknesses).

All information services markets, however, are beset by the presence of information monopoly or asymmetry (Park, 1996), which is the exact opposite of perfect information or complete knowledge. Geographical, cultural, business, and regulatory differences that are inherent to the offshoring context exacerbate this asymmetry (Ramarapu, Parzinger, & Lado, 1997). The vendor faces a rather limited amount of risk from this asymmetry. Depending on its past experience with other clients, the vendor may or may not possess an understanding of this client's application context or its industry's business processes. The vendor also may have

concerns regarding its interactions with the client or the latter's financial stability. However, the client's lack of familiarity with the vendor's business environment exposes it to many more risks, including shoddy development processes and practices, poor product quality, poor documentation, incomplete or inappropriate solutions, and business process integration issues. The lack of information transparency thus affects the client more severely than the vendor.

The larger and more complex the application is, the higher the development risks are (Barros, Werner, & Travassos, 2004). Therefore, in order to reduce these risks and to minimize the impacts of any detrimental outcomes, the client organization follows a cautious approach in identifying initial work to send offshore. This implies small applications or components of low complexity, for which specifications can be communicated completely and whose development process is highly structured. Such applications call for little vendor supervision and no need for a window into the vendor's internal processes. A transaction-oriented, contractual relationship between client and vendor suffices as the control mechanism, thereby keeping interorganizational coordination costs incurred by the client to a minimum.

Unpleasant experiences with these initial applications quickly can end the client's experimentation with offshore development. On the other hand, successful initiatives not only result in reduced costs but also make the client aware that offshoring can sustain or even raise software quality (Khan & Fitzgerald, 2004). This emboldens the client to outsource more applications to the same vendor or even to new vendors (Benamati & Rajkumar, 2002). Heightened client awareness, though, is accompanied by the expectation that vendors will put the effort into acquiring a deeper understanding of its business processes and informational needs. In turn, vendors rationalize the extra effort required from these fresh expectations with their own long-term goals of being assigned progressively higher-level work or serving larger markets. Successful outcomes that result

from these mutual commitments serve to supplant the client's original objective of reduced costs with the realization that it may be able to reap collaborative synergies with vendors by moving beyond the scope of simple applications into higher-end development (King, 2005). Collaborative efforts on more complex tasks can yield innovative outcomes, including new solutions; new approaches to problems; unique value added propositions; and, ultimately, original sources of competitive advantage for the client (Dyer & Singh, 1998). Accordingly, the client entrusts more complex applications to selected vendors, with less structure and more room for creativity and innovation. As enhanced complexity necessitates a closer working relationship, the client may move to the middle by replacing several short-term contracts with a handful of long-term contracts (Clemons, Reddi, & Row, 1993), essentially substituting transactional interactions with fewer relational mechanisms.

However, long-term contracts increase interorganizational coordination costs by placing greater collaborative and enforcement burdens on the client. They also hinder collaborative innovation, because both client and vendor may be reluctant to share valuable, firm-specific resources with unfettered, autonomous, distant partners (Goes & Park, 1997). Their mutual independence and freedom of action outside the limited terms of the contract, even a long-term one, causes both organizations to continue to be wary of each other's intentions and actions. Thus, beyond the simplest applications, the continued use of purely contractual arrangements is ineffective, because contracts merely replace the traditional in-house project risks with a different set of more perilous vendor risks. Despite initial successes, therefore, the probability of eventual failure remains significantly high (Natovich, 2003).

Ongoing offshore relationships also bring into play asset specificity, dependence, measurement difficulty, and uncertainty. Asset specificity refers to investments by relationship partners in informational or physical assets that have little value outside the context of that specific

relationship. With higher application complexity, both vendor and client invest more time and energy into understanding each other's unique processes, and increasing asset specificity is inevitable. Asset specificity and the resultant dependence give rise to opportunism, the tendency of one partner (usually the vendor) to take advantage of the fact that the other partner has invested too much in the relationship to walk away from it (Lonsdale, 2001). The awareness of such an exit barrier for the client may tempt the vendor to exhibit opportunism in the form of cost escalation, unreasonably high charges for services not explicitly mentioned in the contract, inexperienced staff assigned to the project, reduced quality and service levels, or holding the client captive to obsolete or inappropriate technologies.

Measurement difficulty translates into inscrutability of the vendor's work processes. Since both short-term and long-term contracts are much more conducive to the measurement of outcomes than processes, the vendor's internal practices remain largely hidden from the client. This may trigger not just opportunism of the kinds already discussed but also vendor manipulation of internal project management data to deflect culpability in problem situations. Finally, uncertainty refers to the prospect of unanticipated developments in the technological, business, or political environments, which are of particular concern, given the global nature of offshoring relationships.

Client organizations are aware of these pitfalls. In efforts to preempt problems that might stem from asset specificity, dependence, and measurement difficulty, some attempt to anticipate as many contingencies as possible and to incorporate them into increasingly intricate contracts. However, the preemptive use of contingent contractual clauses is a futile exercise. With consultants and legal experts in tow, such organizations eventually can get to the point where the sheer overheads of administering these contracts make them untenable. Alternatively, the mistrust engendered by minutiae can vitiate their relationships with offshore vendors. The intensification of these dysfunctional dy-

namics provides a window for network-style relationships to replace contracts.

Stage 2: Networks

As application complexity increases, the efficacy of contractual mechanisms breaks down. The client's success with unstructured, intricate projects is more dependent on the vendor's internal practices and methods than before. This creates the need for better scrutiny of the vendor's processes, which contracts alone are unable to provide, given their enforcement-oriented nature. Higher task complexity necessitates joint coordination and a closer, more cooperative working relationship. Pronounced differences in location, culture, business processes, and regulatory practices stemming from the very nature of offshoring magnify these imperatives. A networklike linkage based on trust, solidarity, shared values, and open communication, therefore, represents a better option for the client, because it engenders procedural coordination, vendor self-enforcement, and reciprocity (collaboration and cooperation). The establishment of such a relationship tends to foster product and process innovation (Ritter & Gemunden, 2003).

Interorganizational networks such as joint ventures, alliances, associations, and consortia typically consist of three or more organizations. However, a client-vendor dyad with soft interorganizational boundaries can exhibit working interfaces and other properties similar to those of classic networks. The major advantage of networklike relationships over purely contractual mechanisms for offshoring is that they better facilitate the codification and communication of technological skills and organizational knowledge (Park, 1996). In particular, underlying tacit knowledge is more successfully exchanged due to the mutual trust, strong social ties, and shared values/systems in such relationships (Dhanaraj, Lyles, Steensma, & Tihanyi, 2004). The successful codification and transfer of tacit undocumented knowledge is crucial for more complex applications, as it provides the offshore vendor a better grasp of the client's general business context and unstructured aspects of task requirements. Although such knowledge exchanges increase coordination costs even more (Sobrero & Roberts, 2001), they are accompanied by the joint creation of new knowledge, a higher order benefit (Sharma, 1997). Both tacit knowledge transfer and new knowledge creation are invaluable in the development of more complex applications and necessary in order for synergistic innovations to emerge (Hardy, Phillips, & Lawrence, 2003).

Procedural coordination, a key attribute of network-style relationships, enhances the effectiveness of existing contractual client-vendor exchanges when used as a secondary linkage mechanism (Sobrero & Schrader, 1998). The complementary advantages that it provides are highly effective, even if the original contractual exchanges were highly customized (Poppo & Zenger, 2002). Studies of IT outsourcing have shown that trust, another critical attribute of networks, is preferred over contracts, particularly when contractual hazards are perceived as high (Barthelemy, 2003). In an empirical study, IT outsourcing projects often were found to begin with simple, outcome-oriented controls in place, but performance problems usually led to the introduction of additional controls oriented toward trust, vendor behavior, and clan-like affiliation (Choudhury & Sabherwal, 2003). Another empirical study that focused on software offshoring by small firms found that clients attempted to mitigate the high inherent transaction costs of contracts through embedded network ties in the form of mutually trusted individuals or liaisons of knowledge flows (Carmel & Nicholson, 2005). The importance of trust in interorganizational collaboration is well-documented elsewhere, as well, both for IT outsourcing (Jennex & Adelakun, 2003; Lanfield-Smith & Smith, 2003; Sabherwal, 1999; Willcocks & Choi, 1995) and in general (Vangen & Huxham, 2003; Zaheer, McEvily, & Perrone, 1998).

Stage 3: Hierarchies

The introduction of trust and clan-like affiliation represents a step up from an exclusive

reliance on contracts. However, network-style client-vendor relationships by themselves or in conjunction with long-term contracts by no means represent the stable state. The very network attributes that serve to bring the client and vendor together in a closer working relationship (i.e., trust, reciprocity, effective knowledge transfer, and vendor self-enforcement) ultimately render it ineffective. Eventually, more authoritative mechanisms are needed, and the appropriate relationship is a command-oriented hierarchy. This may be achieved in one of two ways. The first entails the client acquiring a formal stake in a vendor organization through part or full ownership (i.e., vertical integration). In the second approach, the client sets up a captive offshore subsidiary of its own (Preston, 2004).

While neither of these two approaches may be feasible for every client organization, the following discussion justifies the eventual need for authoritative mechanisms. The primary impetus for the transition from Stage 2 to Stage 3 comes from continuing changes to the attributes of offshore applications to the point where higher complexity is accompanied increasingly by criticality to the client's business processes (Jensen, 2004). This is but a natural outcome of a close working relationship in which client-vendor synergies constantly are being explored. Business criticality poses enhanced risk and necessitates greater checks and balances on the vendor than are feasible through the network. Trust and reciprocity alone become insufficient to maintain a healthy relationship, and institutional control mechanisms are needed (Miles & Snow, 1984). The resulting swing toward monitoring and formalization, however, threatens the autonomy of the partner organizations, even as they are highly interdependent (Van de Ven & Walker, 1984). The implementation of new checks, balances, and controls also imposes even higher coordination costs, which destabilizes and ultimately breaks the network (Park, 1996). Thus, networks, like contractual markets in the previous stage, eventually unravel in the face of ever-changing attributes of offshore applications. Key empirical evidence for this comes

from a study that found that partner commitment in outsourcing relationships actually declines with the age of the relationship, increasing the chances of conflict (Lee & Kim, 1999).

Trust, a fundamental basis of network-like arrangements, is known to be particularly fleeting in knowledge-sharing relationships. The progression of such relationships is affected more importantly by procedural justice than by trust (Daellenbach & Davenport, 2004) and by structures that propagate goal congruence, particularly in the presence of opportunism (Jap & Anderson, 2003). In a study, opportunism was found to be reduced more by formalizing policy than by encouraging cooperation (Dahlstrom & Nygaard, 1999). Procedural justice, goal congruence, and formalization are all attributes associated more closely with hierarchical relationships than with networks.

In a seminal research study, equity-based mechanisms (e.g., part or full ownership of one organization by the other, etc.) were found to benefit partner organizations in a relationship more than nonequity mechanisms (Zollo, Reuer, & Singh, 2002). Likewise, in a study of strategic alliances, equity-based alliances were found to promote greater interfirm knowledge transfer (Mowery, Oxley, & Silverman, 1996). As has been noted, applications development offshoring entails a high degree of knowledge sharing and transfer. A study of vertical partnerships found that linkages that support the extensive integration of the structures and management of both organizations were more successful from the perspectives of both parties (Donada, 2002). Another study that focused on asymmetric partnerships, in particular, found that effectiveness and competitive advantage were higher when the leader organization utilized authoritative coordination mechanisms (Hernandez-Espallardo & Arcas-Lario, 2003).

In other evidence, transaction cost economics theory (Williamson, 1975, 1985) predicts that a preponderance of transactions in which utilized assets have relatively little value outside of the transaction (high asset specificity) will lead to an integration of assets. The firm that stands the most to lose from a

holdup (i.e., the client) by the other firm will tend to want to acquire the assets of the other. It also has been shown that in an interorganizational relationship in which the two firms hold complementary assets, the sum total of these assets leads to synergies or increasing returns to scale, and therefore, there is a natural tendency for integration or for both sets of assets to be acquired by one firm (Hart, 1995).

Last, but not least, the introduction of a hierarchy alleviates internal political concerns in the client organization regarding loss of valued in-house technical expertise and skills to offshore vendors. The establishment of a hierarchy brings these skills back into the client's fold and under its control, albeit possibly at arm's length. This regained expertise now may be leveraged by the client and sold to other organizations, thus providing an added source of business revenue.

OFFSHORE DEVELOPMENT FOR ETRUS: A CASE STUDY

This section provides an illustration of the proposed framework for offshore applications development in the form of a case study. The narrative describes how the interactions of Etrus, a real-world client organization, with various offshore development vendors, successively has mirrored the three stages of contracts, networks, and hierarchies. It highlights the interplay among various factors that influenced this evolution, including constantly changing objectives, internal imperatives, and external compulsions on both sides. While the specific circumstances of this case are unique, the broader lessons it offers are consistent with the proposed evolutionary framework.

Etrus Corp.[1] is a 13-year-old specialty telecommunications company whose expertise lies in identifying and relieving critical stress points and bottlenecks that impede its business customers' network performance. Using a mix of systems, software, and services, it helps its customers to make maximum use of their applications that drive their businesses, while minimizing the total cost of their network own-

ership. Originally established as a Delaware corporation in the high-end, IP-based optical niche market, the company since has grown tremendously to approximately $400 million in annual sales, largely through a series of acquisitions and mergers, including 11 in the past eight years. It also has carefully crafted technology and business partnerships with major telecom players such as Cisco in order to offer a fuller range of solutions for its customers. Comprised of 1,500 employees, it now occupies specific niches across the breadth of telecommunications and serves customers on every major continent. These include cable service providers, telecommunications companies (telcos) such as Verizon, British Telecom, MCI, and AT&T; government agencies; and private corporations with their own networks. Its products and services include multiplexers, aggregators, switches, integrated platforms/systems, and specialized network layer services/enhancements, some of which have won prestigious industry awards. Essentially a design and engineering firm, Etrus envisions and builds new off-the-shelf products for both the horizontal and vertical markets, with customized variations for individual customers. Its contemporary rivals include Cisco (its business partner) as well as Lucent and Nortel. Along the way, Etrus also has acquired ISO 9001 certification.

Etrus' foray into applications development offshoring began with two conversion-oriented projects. These were precipitated by Etrus' series of acquisitions of other specialty telecommunications companies in 2001 and 2002. The new product development processes of both Etrus and the acquired companies had depended extensively on an industry standard product management software tool from Agile, which enabled them to manage the cost of their products and to track various aspects of their supply chain (e.g., tracking parts and materials for manufacturing). While this platform compatibility between the acquiring and acquired companies definitely was a positive feature, all these companies were using different versions of the Agile software and, therefore, essentially employing somewhat different

design, documentation, and manufacturing standards. Thus, an important objective was to get the data of the acquired companies into the structure and standards followed by Etrus so that the consolidated data could be managed centrally in an integrated manner. One of the conversion projects, therefore, entailed upgrading the software used by an acquired partner to a more modern and recent version by converting its design and database structure and content so that it was compatible with that used by Etrus. The other project actually entailed rolling a more recent version back to an older version, in the case of a different acquired partner that was using a more modern version of the software than Etrus was. While these conversion tasks were highly structured and fairly straightforward, they could not be fully automated, as the conversion process consisted of some decision points, choices, and new data creation. In addition, since Etrus had decided to leverage the opportunity created by both conversion efforts by adding new functionality to the off-the-shelf product, this called for some custom programming.

Both conversion projects were outsourced to a Silicon Valley-based vendor called Blue Ridge, which had technology and business relationships with Agile and entrusted the projects to its offshore subsidiary in India. Etrus' selection of Blue Ridge was based on the latter's expertise with Agile software and on its apparent application development process maturity. This maturity was evidenced by its ISO-9001 certification, and its achievements related to the Capability Maturity Model (CMM), developed by Carnegie-Mellon University's Software Engineering Institute (SEI).[2] It used a proprietary extended framework that focused on key processes, and its unique internal governance mechanisms periodically monitored ongoing projects to provide directional guidance as well as to issue resolution. Blue Ridge previously had employed its skills and methodology successfully with Fortune 100 customers. It prided itself on being able to provide clients with a window into their internal processes and with total visibility of up-to-the-minute performance

metrics and execution status of ongoing projects, regardless of the physical location of the actual application development efforts.

Blue Ridge completed each project in about four weeks at a cost of approximately $20,000. Both development projects were deemed to be resounding successes. In the words of Etrus' Vice President of Engineering Services, the conversion work, together with the custom-built features from Blue Ridge, "dramatically reduces the time to take a product from design to large-scale production, giving Etrus vastly more revenue and market share. This enables us to achieve aggressive new product launch objectives through supply chain collaboration, so critical to our success. We couldn't have asked for better support from Blue Ridge."

These initial small experiments with application development offshoring represented the first offshoring stage for Etrus. The successful culmination of contracts with Blue Ridge set a positive internal tone within Etrus and bolstered the confidence of its IT managers in offshore development. As the company continued to grow rapidly through acquisitions, the IT group decided that it would be best to focus in-house resources on tasks associated with new product development and to offshore more of the other routine application development work. However, the fast pace of corporate change and the quick adaptation expected from IT meant that the internal IT staff had very little time available to engage collaboratively with vendors in order to jointly generate specifications that were clear and detailed enough for the latter to take over and to complete the development process. This constraint prevented many new offshoring projects from being initiated in the first place. The solution that was agreed upon was to look for parcels of work that were both complex enough to leverage the cost and quality benefits of offshore development and, at the same time, structured enough to not set the internal IT group back in terms of time spent working with the vendor to generate excessively detailed specifications. This approach also would require the careful selection of vendors whose strengths matched the attributes

of specific projects. The first such opportunity came about in the shape of a need to customize and implement an internal business application, an off-the-shelf engineering and design information system oriented around engineering resources and portfolio management and built on a popular technology platform. The rest of the Etrus case study describes the characteristics of this particular development project, how the offshore vendor was selected and managed, the outcome of the development effort, and a discussion of its implications.

This project represented a moderately complex application, whose deliverables consisted of the system setup, the standard template setup and custom template design, the standard reports setup and custom report design, and some custom workflow programming logic. It entailed a different technological platform than the other two projects, and Blue Ridge was deemed not to possess the ideal expertise for this job. Consistent with the aforementioned need to carefully select a vendor whose strengths matched the attributes of the application to be developed, the project instead was awarded directly to Fiore, a leading offshore-based global IT firm. Fiore's strengths ranged across the full technology life cycle spectrum, including business and technology consulting, research and development, implementation, and process ownership/operations. For its applications development business line, Fiore owned and operated sophisticated, mature, offshore development facilities. Aside from Fiore's excellent reputation, Etrus had had prior contacts with Fiore's R&D group, which had designed and developed some components of their standard product offerings in the past. Since Fiore was well-placed in the high-end, engineering software niche, Etrus' IT managers believed that Fiore possessed the background and skills to successfully deliver this application, which was more complex relative to the first two that they had offshored earlier.

The application, when complete, would provide Etrus with the resource loading and skill prediction requirements across their entire engineering portfolio. It would enable them to load all their engineering projects that they had in the pipeline and track their status, their project plans, and the skills that it was going to take to get them done. This project portfolio information then could be rolled up and mapped onto the skill sets of individual product designers and engineers in order to yield accurate estimates of how many people of each skill set they would need over the next couple of years and the personnel budgets that all of this would entail. It also would enable Etrus to build the background or foundation to support the product development commitments that their salespeople were making to their customers and to reconcile market realities with their planning processes.

At a higher level of complexity relative to the prior offshore development projects, the development of this application called for closer communication and interaction between client and offshore vendor. However, even as this project was underway, Etrus' internal resources, including IT, were under severe strain due to other pending corporate acquisitions. It was decided, therefore, to focus as much of the iterative dialog with the vendor as possible in the initial stages of the development life cycle and then to hand over the process entirely to Fiore to manage until the end product was ready. Fiore's project leads, who were technical specialists, worked closely with Etrus' internal IT staff in the initial stages. This minimized the number of contact points between client and vendor in the later stages of development, thereby enabling Etrus to function within their time constraints, but it also ensured that each contact point served as an intensive knowledge transfer linkage. Once the leads had internalized the specifications and the deliverables, the client then would be ready for them to move the actual development work offshore to Fiore's premises and to manage it from there.

Thus, in the requirements analysis stage, Etrus' internal IT staff provided the vendor not only with documents containing detailed requirements specifications but also with some graphical mockups of key deliverables such as input/output screens and reports. These

prototypical screens and reports then were subjected to iterative changes based on short, rapid feedback cycles between Fiore and Etrus. While the technical context was well-documented and communicated to the vendor in this manner, the transfer of business domain knowledge or the organizational context (i.e., knowledge of the client's business processes) was not given as much attention. Both client and vendor agreed that the application, while moderately complex in effort, represented a fairly structured task with few unknowns; for instance, the hardware, software, and technical architecture for the project already had been decided by Etrus. Aside from some examples of how things were done at Etrus, it wasn't really necessary to get into details of the business knowledge in order for Fiore to complete the project, despite the fact that the two organizations had no prior interactions. This also was the reason that the project leads from Fiore were technical specialists rather than business or process specialists.

Since the moderate level of application complexity also called for some degree of vendor control, Etrus chose to anchor this control to outcome-oriented measurable standards for the vendor to follow. This was consistent with the fact that the technical architecture already had been set by the client. The contract, therefore, spelled out both timelines and quality standards. An example of the latter was specifications pertaining to the expected number of rework cycles and other measures of operational efficiencies such as software bugs. These standards were based on Etrus' objective assessment of the project's complexity and its own historical data from other internally developed applications. Also added to the contract were specific deliverables associated with checkpoints. One example of a deliverable at a checkpoint entailed running the set of custom workflows through a series of test processes to ensure that the set touched all the right spots in the workflow.

Enforcement of standards, while not strictly followed through formal audits, was built into these checkpoints as threshold gates. If deliverables were determined not to have met certain standard levels, exit clauses would kick in, enabling Etrus to revisit the contract or to stop the project all together. Essentially, the client then could choose to take all the work that had been done to date, together with documentation, and to finish the project itself or hand it over to another vendor. Fiore then would be paid for work done to that point and no further. The intent of these clauses, though, was not based on an adversarial premise; the clauses were placed in the contract largely as mechanisms to document the expectations of quality to which both client and vendor were committed, essentially as vendor guidelines. Further, although the contract focused on deliverables rather than on process, it did not contain any provisions for special rewards or incentives for work that exceeded standards or that was completed ahead of schedule. The message to Fiore was, "We don't have the time to accomplish this ourselves, so as long as you can manage the project and demonstrate the deliverables and the milestones, we'll be perfectly happy."

Due to various reasons, including time constraints and the nature of the development work, the client maintained a clearly outcome-oriented stance; the vendor, on the other hand, was CMM-certified at a high level and took pride in following strictly established, documented, and publicized processes. Despite the fact that Fiore was a big player for whom this project represented, at best, a small foot in the door, it proceeded to apply high process standards for itself, including rigorous testing and documentation. This was over and above its self-enforcement of the deliverable-oriented contractual terms that had been set by Etrus. These respective outcome- and process-oriented approaches of client and vendor had a decidedly synergistic effect; the end result was a development project that came in at only slightly higher than the funds budgeted ($45,000 vs. $41,000), took less time than expected (six weeks vs. two months), and was regarded as successful by the client, despite an unplanned switch of offshore project managers in the middle of the development effort. The vendor's process orientation

ensured the continuity that was needed at the point of this switch, and the carefully crafted deliverables-oriented measures that had been set by Etrus helped Fiore to carve the path to an unambiguous end point. When the project was complete, the in-house IT staff at Etrus was unanimous in its opinion that, although the work could have been done internally, it would have been more unpredictable, it would have entailed a longer learning curve, it would have cost approximately 50% more, and it probably would not have been as high of a technical quality.

Etrus' successful experience with Fiore represents the second stage in its evolutionary experiences with application development offshoring. This stage was characterized by higher complexity work, a closer relationship with the vendor, and a greater focus on process. Ironically, though, as the IT staff at Etrus now considers a full-fledged role for applications development offshoring, some of the key factors that worked to its advantage in its recent project are the ones that it thinks will work against it in the future. Specifically, the dynamics of the telecommunications industry are changing, and the acquisition and consolidation mania is beginning to slow down. This implies that the IT staff will be less engaged in putting out acquisition-related fires and will be more available to leverage offshoring by sending out increasingly complex applications for development. While the benefits of offshoring more complex applications likely will be of a higher order, the need for closer working relationships between client and vendor IT personnel also is expected to be greater. The higher the application complexity, the greater is the need for joint, iterative activities in any development effort. While the internal IT staff finally may have more time available for such collaborative client-vendor activities, the uncertainty embedded in complex applications also usually results in several rounds of changes to initial specifications as possibilities are explored, as knowledge is exchanged between client and vendor, and as implications continually are clarified. With vendors who flaunt high-level CMM certifications, this creates two problems.

One has to do with the enormous level of detail needed in initially documented specifications, because CMM processes require extensive documentation. The second has to do with the communications, documentation, testing, and recertification costs of reworking these specifications with each round of changes. In other words, while vendor CMM certification provides reliability and confidence in the end product, it also adds a tremendous overhead burden to the cost structure. The net result can be that any incremental gains or values from initial offshoring work are offset by the high coordination costs of adhering to process-oriented standards such as CMM as the work gets more complex.

For these reasons, Etrus has decided to disengage itself from Fiore. It has committed itself to establishing a captive (subsidiary) offshore application development facility of its own in the near future so that it can better leverage the higher-order benefits of offshoring and, at the same time, maintain some control over what otherwise would be exorbitantly high vendor coordination costs. Doing so will take Etrus into the third and final evolutionary stage for offshore development.

DISCUSSION AND IMPLICATIONS

Using a theoretical framework and a supporting case study, it has been argued that client-vendor relationships in application development offshoring contexts evolve first from transaction-oriented, price-based contracts to loose, informal, but fewer networks based on trust and vendor self-enforcement. Subsequently, the growing complexity and criticality of applications introduces the need for greater checks and balances, which necessitates a hierarchical arrangement: part- or full-client ownership of a vendor. While this final stage actually may not be feasible for every client, essentially, a cost reduction game is replaced by one of risk control.

There is somewhat of an irony in the position taken by this article that a formal hierarchy ought to succeed a networklike stage in a cli-

ent-vendor relationship. Vertical integration as a control-oriented, preferred organizational response to external uncertainty first was proposed in a classic work a long time ago (Thompson, 1967). Many years later, however, it was argued that creating a hierarchy entails mechanisms that are unnecessary, that are overkill, and that are difficult to establish or too costly to sustain. In its place, the network, with its freedom of exit attributes, was propounded as a superior response to dynamic, unpredictable environments (Achrol, 1997).

The knowledge of a natural relationship progression can help decision makers on either side of an offshoring relationship. On the client side, senior business and technology executives can make more informed judgments regarding the viability of offshoring beyond the short term, given the kind of commitments that such vendor relationships ultimately will entail. The resultant better decision process will prevent organizations that are ill-equipped to deal with the concomitant costs from embarking upon complex or critical offshore projects. It will also enable organizations that do possess such a capability to be better prepared for the internal changes that will accompany changing relationship structures with vendor organizations. At the least, the clarity resulting from understanding the sequential stages of interorganizational processes will encourage executives to make candid assessments of why they might be interested in offshoring in the first place.

On the vendor side, the implications of a relationship progression perhaps are more profound. With the realization that clients will seek to regain control as applications evolve to become complex and critical, decision makers in vendor organizations are faced with fundamental questions pertaining to organizational mission, culture, and employee work autonomy. For example, should they even let client-vendor relationships progress to such points, or should they plan to stay with lower- and mid-level work? The greatest dilemma here is for the creative, decentralized, organic types of offshore vendor organizations. On the one hand, they may value the empowerment that their internal environment provides them. On the other hand, this empowerment may be threatened when put to good use for higher-level work.

For researchers interested in pursuing further the progression of offshoring relationships, this article offers a couple of different directions for more rigorous empirical inquiry. The most obvious one is the implicit proposition that as a vendor-client offshoring relationship is established and as it ages and matures, vendor oversight first is conducted through possibly increasingly complicated contractual mechanisms; then through loose, trust-based, affiliation-solidarity networks; and finally, through a client-controlled hierarchy. An alternative investigation scheme would avoid using relationship age as the progression benchmark and, instead, would look for evidence of the three stages by employing application complexity as a surrogate for relationship maturity. If either of these two methods reveals the existence of such stages, then the next logical step would be to test whether hierarchies as the interorganizational structure of choice in mature relationships are really more effective than other alternative mechanisms.

REFERENCES

Achrol, R.S. (1997). Changes in the theory of interorganizational relations in marketing: Toward a network paradigm. *Journal of the Academy of Marketing Science*, *25*(1), 56–71.

Aubert, B.A., Rivard, S., & Patry, M. (2004). A transaction cost model of IT outsourcing. *Information & Management*, *41*(7), 921–932.

Barringer, B.R., & Harrison, J.S. (2000). Walking a tightrope: Creating value through interorganizational relationships. *Journal of Management*, *26*(3), 367–403.

Barros, M.O., Werner, C.M.L., & Travassos, G.H. (2004). Supporting risks in software project management. *The Journal of Systems and Software*, *20*(1-2), 21–35.

Barthelemy, J. (2003). The hard and soft sides of IT outsourcing management. *European Management Journal*, *21*(5), 539–548.

Benamati, J., & Rajkumar, T.M. (2002). The application development outsourcing decision: An application of the technology acceptance model. *The Journal of Computer Information Systems*, *42*(4), 35–43.

Carmel, E., & Agarwal, R. (2002). The maturation of offshore sourcing of IT work. *Management Information Systems Quarterly-Executive*, *1*(2), 65–76.

Carmel, E., & Nicholson, B. (2005). Small firms and offshore software outsourcing: High transaction costs and their mitigation. *Journal of Global Information Management*, *13*(3), 33–54.

Choudhury, V., & Sabherwal, R. (2003). Portfolios of control in outsourced software development projects. *Information Systems Research*, *14*(3), 291–314.

Clemons, E.K., Reddi, S.P., & Row, M.C. (1993). The impact of information technology on the organization of economic activity: The "move to the middle" hypothesis. *Journal of Management Information Systems*, *10*(2), 9–35.

Daellenbach, U.S., & Davenport, S.J. (2004). Establishing trust during the formation of technology alliances. *Journal of Technology Transfer*, *29*(2), 187–202.

Dahlstrom, R., & Nygaard, A. (1999). An empirical investigation of ex post transaction costs in franchised distribution channels. *Journal of Marketing Research*, *36*(2), 160–170.

Dhanaraj, C., Lyles, M.A., Steensma, H.K., & Tihanyi, L. (2004). Managing tacit and explicit knowledge transfer in IJVs: The role of relational embeddedness and the impact on performance. *Journal of International Business Studies*, *35*(5), 428–442.

Donada, C. (2002). Generating cooperative gain in a vertical partnership: A supplier's perspective. *Canadian Journal of Administrative Sciences*, *19*(2), 173–183.

Dyer, J.H., & Singh, H. (1998). The relational view: Cooperative strategy and sources of interorganizational competitive advantage. *Academy of Management Review*, *23*(4), 660–679.

Goes, J.B., & Park, S.H. (1997). Interorganizational links and innovation: The case of hospital services. *Academy of Management Journal*, *40*(3), 673–696.

Hardy, C., Phillips, N., & Lawrence, T.B. (2003). Resources, knowledge and influence: The organizational effects of interorganizational collaboration. *The Journal of Management Studies*, *40*(2), 321–347.

Hart, S.L. (1995). A natural-resource-based view of the firm. *Academy of Management Review*, *20*(4), 986–1014.

Hernandez-Espallardo, M., & Arcas-Lario, N. (2003). The effects of authoritative mechanisms of coordination on market orientation in asymmetrical channel partnerships. *International Journal of Research in Marketing*, *20*(2), 133–152.

Jap, S.D., & Anderson, E. (2003). Safeguarding interorganizational performance and continuity under ex post opportunism. *Management Science*, *49*(12), 1684–1701.

Jennex, M.E., & Adelakun, O. 2003. Success factors for offshore information system development. *Journal of Information Technology Cases and Applications*, *5*(3), 12–31.

Jensen, B.K. (2004). The expert opinion. *Journal of Information Technology Cases and Applications*, *6*(4), 51–54.

Khan, N., & Fitzgerald, G. (2004). Dimensions of offshore outsourcing business models. *Journal of Information Technology Cases and Applications*, *6*(3), 35–50.

King, W. (2005). Outsourcing becomes more complex. *Information Systems Management*, *22*(2), 89–90.

Koh, C., Ang, S., & Straub, D.W. (2004). IT outsourcing success: A psychological contract perspective. *Information Systems Research*, *15*(4), 356–373.

Lander, M.C., Purvis, R.L., & McCray, G.E. (2004). Trust-building mechanisms utilized in outsourced IS development projects: A case study. *Information & Management*, *41*(4), 509–528.

Lanfield-Smith, K., & Smith, D. (2003). Management control systems and trust in outsourcing relationships. *Management Accounting Research*, *14*(3), 281–307.

Lee, J.-N., & Kim, Y.-G. (1999). Effect of partnership quality on IS outsourcing success: Conceptual framework and empirical validation. *Journal of Management Information Systems*, *15*(4), 29–61.

Lonsdale, C. (2001). Locked-in to supplier dominance: On the dangers of asset specificity for the outsourcing decision. *The Journal of Supply Chain Management*, *37*(2), 22–27.

Matloff, N. (2004). Globalization and the American IT worker. *Communications of the ACM*, *47*(11), 27–29.

Miles, R.E., & Snow, C.C. (1984). Fit, failure and the hall of fame. *California Management Review*, *26*(3), 10–28.

Mowery, D.C., Oxley, J.E., & Silverman, B.S. (1996, Winter). Strategic alliances and interfirm knowledge transfer. *Strategic Management Journal*, *17*, 77–91.

Natovich, J. (2003). Vendor related risks in IT development: A chronology of an outsourced project failure. *Technology Analysis & Strategic Management*, *15*(4), 409–419.

Park, S.H. (1996). Managing an interorganizational network: A framework of the institutional mechanism for network control. *Organization Studies*, *17*(5), 795–824.

Poppo, L., & Zenger, T. (2002). Do formal contracts and relational governance function as substitutes or complements? *Strategic Management Journal*, *23*(8), 707–725.

Preston, S. (2004). Lost in migration: Offshore need not mean outsourced. *Strategy & Leadership*, *32*(6), 32–36.

Rajkumar, T.M., & Mani, R.V.S. (2001). Offshore software development: The view from Indian suppliers. *Information Systems Management*, *18*(2), 63–73.

Ramarapu, N., Parzinger, M.J., & Lado, A.A. (1997). Issues in foreign outsourcing. *Information Systems Management*, *14*(2),

27–31.

Richmond, W.B., & Seidmann, A. (1993). Software development outsourcing contract: Structure and business value. *Journal of Management Information Systems*, *10*(1), 57–72.

Ritter, T., & Gemunden, H.G. (2003). Network competence: Its impact on innovation success and its antecedents. *Journal of Business Research*, *56*(9), 745–755.

Sabherwal, R. (1999). The role of trust in outsourced IS development projects. *Communications of the ACM*, *42*(2), 80–86.

Sharma, A. (1997). Professional as agent: Knowledge asymmetry in agency exchange. *Academy of Management Review*, *22*(3), 758–798.

Sobrero, M., & Roberts, E.B. (2001). The trade-off between efficiency and learning in interorganizational relationships for product development. *Management Science*, *47*(4), 493–511.

Sobrero, M., & Schrader, S. (1998). Structuring inter-firm relationships: A meta-analytic approach. *Organization Studies*, *19*(4), 585–615.

Thompson, J.D. (1967). *Organizations in action*. New York: McGraw-Hill.

Van de Ven, A.H., & Walker, G. (1984). The dynamics of interorganizational coordination. *Administrative Science Quarterly*, *29*(4), 598–621.

Vangen, S., & Huxham, C. (2003). Nurturing collaborative relations: Building trust in interorganizational collaboration. *The Journal of Applied Behavioral Science*, *39*(1), 5–31.

Willcocks, L., & Choi, C.J. (1995). Cooperative partnership and "total" IT outsourcing: From contractual obligation to strategic alliance. *European Management Journal*, *13*(1), 67–78.

Williamson, O.E. (1975). *Markets and hierarchies: Analysis and antitrust implications*. New York: Free Press.

Williamson, O.E. (1985). *The economic institutions of capitalism*. New York: Free Press.

Zaheer, A., McEvily, B., & Perrone, V. (1998). Does trust matter? Exploring the effects of interorganizational and interpersonal trust on performance. *Organization Science*, *9*(2), 141–159.

Zollo, M., Reuer, J.J., & Singh, H. (2002). Interorganizational routines and performance in strategic alliances. *Organization Science*, *13*(6), 701–713.

ENDNOTES

[1]   The identities of the client and key vendor organizations have been disguised, and other descriptive statistics and facts carefully distorted in order to maintain confidentiality and yet preserve the essence of the case.

[2]   While the principles of the Software Capability Maturity Model (SW-CMM) continue to be followed by organizations worldwide, the model itself has been superseded by a newer one called Capability Maturity Model Integration (CMMI), also from SEI. The original SW-CMM, now in sunset, focused on software development processes. Its structure was characterized by five levels of increasing maturity: Level 1–Initial, Level 2–Repeatable, Level 3–Defined, Level 4–Managed, and Level 5–Optimizing. Organizations at Level 5 epitomize highly mature, efficient, productive, innovative, and proactive development processes.

Rajesh Mirani is associate professor of information systems at the Merrick School of Business, University of Baltimore. He holds a PhD from the University of Pittsburgh. His research interests are generally oriented around the broad issue of business and information systems alignment. His present focus is on how organizations manage their offshoring endeavors. He has previously published in Decision Sciences*,* Journal of MIS*, and* MIS Quarterly*, among other journals.*