Contributors

Soon Ang Head of the Human Resource and Quality Division and Deputy Director of the Information Management Research Center at Nanyang Business School, Nanyang Technological University, Singapore. She gained her PhD at the University of Minnesota, USA. She has published widely in such journals as *Information Systems Research* and *Communications of The ACM* and is on the editorial boards of ISR and JISE. Research interests include cross-national perspectives on information services outsourcing, the impact of foreign employment outsourcing and the role of guanxi networks in labour mobility.

Uday Apte Associate Professor of MIS at Southern Methodist University, Dallas, USA. He holds a PhD from The Wharton School where he previously taught for eight years. Research interests include service operations and MIS. His published research articles have won several awards. He has ten years of management experience in the financial services industry, and has consulted with several major corporations.

Paul W. Beamish Royal Bank Professor of International Business, Western Business School, The University of Western Ontario, London, Ontario, Canada N6A 3K7. Paul has published widely in leading journals and is also the Editor-in-Chief of the *Journal of International Business Studies*. His research has been focused on organizational alliances.

Myun J. Cheon College of Business Administration of University of Ulsan at Ulsan, Korea. He has a PhD from The University of South Carolina, an MBA from Indiana State University, USA, and a BBA from Keimyung University, Korea. His primary areas of interest include outsourcing of IS functions and IS security.

12
Failure in Software Outsourcing: A Case Analysis

SOON ANG AND SEE-KIAT TOH

INTRODUCTION

In recent years, outsourcing of information services has become a pervasive phenomenon. Increasingly, firms rely on external service providers for IT. Not only are firms outsourcing I/S operations, they are also outsourcing IS development (Altinkemer, Chaturvedi and Gulati, 1994; Lacity and Hirschheim, 1993; Mylott 1995).

Anecdotes from trade journals and prior research report many advantages associated with outsourcing. Through outsourcing, firms can reduce costs of developing or managing IS, gain access to specialized IS skills otherwise not found within the organization, and jettison peripheral activities to focus on core competences (see Chapters 1 and 7).

As Jaak Jurison argues in Chapter 6, outsourcing also carries risks. Perhaps the greatest risk of outsourcing is the loss of control (Lacity, Willcocks and Feeny 1995; McFarlan and Nolan 1995). Particularly vulnerable is software outsourcing. Even when software was developed in-house, organizations experienced problems. The complex, dynamic and political nature of software development often translates into time delays, cost overruns, or simply defective systems (Markus, 1983; Abdel-Hamid and Madnick 1991; Davis *et al.* 1992; Lyytinen and Hirschheim 1987). Outsourcing software exacerbates complexities of development (Whang, 1992; Richmond, Seidmann and Whinston 1992).

Strategic Sourcing of Information Systems.
Edited by L. P. Willcocks and M. C. Lacity.

Outsourcing increases the transaction costs of software development by involving external parties and incurring additional legal contractual obligations. Firms bear search costs for sourcing appropriate external software developers, negotiation costs for arriving at mutually agreed upon contract, and monitoring costs for ensuring that legal contractual obligations are fulfilled. As cautioned by Lacity, Willcocks and Feeny (1995), and McFarlan and Nolan (1995), unless properly managed, teething problems and failures in such arrangements can and do occur.

The purpose of this chapter is to demonstrate the additional complexity of developing an information systems externally. Chapter 13 will add the additional complicating factor of software development offshore. Here we focus on diagnosing a failed outsourcing case. Failed endeavours are not widely publicized. Firms restrict access to such information because failures can potentially damage their image, reputation and credibility (Sitkin, 1992). Consequently, research and the public press tend to report more of successes and less of failures. With lopsided research and reporting, organizations may unwittingly assume that failures are rare. Consequently, they forego important lessons of learning from failure. As Sitkin (1992) argues, research that over-emphasizes successes to the extent of avoiding failure is dangerous.

We believe in the control theory of negative feedback that failures provide equal, if not more diagnostic information than do successes. Studying failures is preventive because it helps firms reduce the probability of failures in future. Performance failure and implementation errors provide clear signals that something is amiss and must be changed. Accordingly, the motivation of this study is to provide a careful diagnosis of a failed outsourcing case.

To accomplish our objective, we rely on the diagnostic framework developed in Ang and Beath (1993). The framework rests on Stinchcombe's (1990) argument for the need to embed sufficient hierarchical elements in contracts to overcome shortcomings of outsourcing. The major sections of the chapter run as follows. In the next section, we present briefly the hierarchical elements framework as it applies to software outsourcing. In the next section we present the chronology of events culminating in contract failure. Then we apply the framework to the failed contract and interpret the events based on hierarchical elements analyses. We conclude in the final section with lessons learned and implications for future research and practice.

HIERARCHICAL ELEMENTS FRAMEWORK

The hierarchical elements framework (Ang and Beath 1993) is based on Stinchcombe's (1990) analysis of contracts as hierarchical elements. In his paper, Stinchcombe (1990) raised an interesting puzzle for transaction cost logic. According to transaction costs analysis, firms will refrain from outsourcing when they experience difficulty in specifying requirements in advance, when they are uncertain about prices, costs or quantities, when they require specific assets, or when they cannot control the behaviour of agents. However, in reality, we still observe firms outsourcing even under such adverse conditions. The outsourcing examples that contradict transaction cost analyses include complex R&D projects for weapons development by the government and automobile franchises in private industries (Stinchcombe 1990).

If we apply transaction cost logic to software development, we too will conclude that software should be developed internally because of the inherent uncertainties in specifying requirements determination, high investments in specific assets between the client and contractor, and performance unobservability. When we outsource software development, we should expect additional problems associated with opportunism and excessive co-ordination costs that will eventually lead to failure. In effect, software outsourcing raises the probability of failure over and above the traditional causes of failure of internally developed systems (Markus 1983; Lyytinen and Hirschheim 1987; Davis *et al.* 1992).

According to Stinchcombe, highly uncertain and high asset-specific projects survive outsourcing because firms consciously embed sufficient flexibility into the contractual relation. Flexibility is afforded by incorporating elements that are commonly found as if the activity was governed internally or hierarchically. In effect, with hierarchical elements, outsourcing will emulate hierarchical or internal governance to the extent that outsourcing contracts incorporate the flexibility and necessary control functions afforded by hierarchies. Based on Stinchcombe (1990), Ang and Beath (1993) developed a hierarchical elements framework for analysing software outsourcing. The framework comprises five major types of hierarchical controls:

1. command structures and authority systems;
2. rule-based incentive systems;
3. standard operating procedures;
4. non-market-based pricing systems; and
5. alternative dispute resolution mechanisms.

Command Structures and Authority Systems

According to Stinchcombe (1990), command structures and authority systems are information flows certified as legitimate or authoritative. Command and authority clauses substitute for complete predefinition of contingencies and contingent action. In software outsourcing, command structures assign explicitly decision rights and responsibilities to the contractual parties. Command structures in software include:

1. clauses authorizing certain parties to the contract the right to issue orders or to demand performance;
2. clauses assigning the client, contractor, or both, the power to change project scope without reneging or breaching contract;
3. clauses granting client the right to audit work-in-progress;
4. clauses granting client to choose and change contractor personnel, a privilege generally restricted to hierarchical governance; and
5. clauses granting client the right to cancel project at specified points in the outsourcing contract.

These command structures, together with sample clauses prescribed from various legal handbooks and software management publications, are elaborated in Ang and Beath (1993; 336–337).

Rule-based Incentive Systems

Rule-based incentive systems refer to systems of rewards and punishments tied to behaviour or outcomes and not to the market. Market incentives work well under conditions of certainty where all performance contingencies are considered *ex ante*. In contrast, rule-based incentive systems dissociate market-determined forces. Rule-based systems reflect locally determined inducements for desirable future performance. For example, if timely delivery is vital, penalties for delays beyond agreed completion date and rewards or bonus for early completion can be incorporated into the contract.

Standard Operating Procedures

Standard operating procedures refer to routines describing specific, well-understood actions to be followed by parties in the contract. Standard operating procedures constrain opportunistic behaviour. They facilitate monitoring and reduce uncertainties arising from performance unobservability. Standard operating procedures require contractors to produce formal progress reports to clients, and hold regular face-to-face meetings so that contractual parties can discuss potential problems arising from the project.

Non-market-based Pricing Systems

A non-market-based pricing system works on the principle of cost recovery or a combination of cost recovery and market prices. Generally, clients prefer market-based prices while contractors prefer cost recovery pricing. Market-based pricing is fixed at the onset of the contract. In contrast, cost-recovery pricing is not determined until the project is completed. Market-based pricing therefore reduces uncertainty on the cost of software outsourcing. When development cost is difficult to estimate, a cost recovery system removes risks of uncertainty from the contractor. To mitigate price uncertainty, a client may insist on fixing part of the price for systems delivery at the outset of the contract with a cost recovery-based system allowing clients to modify requirements midstream without necessarily shifting the consequences of change, (i.e. additional costs) onto the contractor. Clauses that mix fixed pricing together with cost recovery attempt to strike a reasonable balance between price risk for the client and compensation risk for the contractor.

Alternative Dispute Resolution Mechanisms

Alternative dispute resolution mechanisms refer to procedures used in resolving conflicts without having recourse to direct court sanctions. When companies resort to court sanctions, conflicts are resolved but relationships between contractual parties are severely impaired. Alternative mechanisms for dispute resolution serve to resolve conflicts with the objective of allowing contractual parties to survive conflicts and even complete the project in hand.

Alternative mechanisms are embedded in contracts, either in the form of private grievance procedures or third-party mediation or arbitration. Private grievance procedures comprise two levels of management. First, project managers from the client and contractor firms meet, discuss and resolve conflicts arising from the transaction. In the event that a resolution is not achieved at that level, senior management teams from both firms may be asked to intervene and negotiate the dispute directly.

In the event that private grievance procedures fail, parties may agree to third-party mediation. In such cases, contractual parties submit the dispute to non-binding mediation by a mutually agreed-upon computer professional. On the other hand, the parties may agree on arbitra-

tion to reach a final and binding solution. Under circumstances when both private grievance procedures and third-party mediation or arbitration fail, parties then pursue remedies available to them in formal legal litigation.

The five major categories of elements described above characterize hierarchical controls that have been prescribed to mitigate the risks of outsourcing. In the next section, we describe the chronology of events leading to a failure in software outsourcing.

SOFTWARE OUTSOURCING FAILURE: A CASE STUDY

We gathered facts of the case from secondary sources. The contract, together with notes of meetings, faxes and letters of correspondence between the two parties, and project documentation, form primary sources of evidence.

Table 12.1 gives an overview of the major milestones and events relating to the case. Dates and names of the companies have been disguised to ensure anonymity and confidentiality of the case.

Parties Involved

In June 19x1, Alpha, a company in the business services industry, approached Omega, a software consultancy firm, for assistance in software development. Omega was a reputable firm. In Omega's own words, they offered:

> "consultancy services to numerous companies for the computerization of their information. These companies are from a wide spectrum of services."

employed competent consultants:

> "Our consultants were well versed in the tasks of developing, evaluating, and implementing computer systems."

regarded highly their professional attitude and efforts:

> "Our approach to successfully assisting you is primarily based on providing experienced consultants who are familiar with your industry, and have a wide range of EDP systems development experience; and who provide an independent and objective appraisal of your requirements and suitable alternatives."

and took pride in their tried and tested systems development methodology:

> "Method X (disguised), is a comprehensive guide to computer planning, development, implementation, and operations. Standard project planning and control forms, checklists, and documentations ensure a thorough, consistent and effective approach to system development. Method was developed by Omega based on worldwide experience gained while conducting hundreds of thousands of hours of systems projects."

Based on Omega's reputation, Alpha negotiated directly with Omega to undertake the project, without open tender.

The Project

In the late 1980s, Alpha recognized the need to computerize. The company possessed no knowledge about information technology, and had been operating in a manual environment. Due to an unprecedented surge in business, Alpha became severely handicapped because of this. Management felt it imperative that they sought automation to relieve mounting paperwork. The firm faced difficulty keeping track of accounts receivables. In some cases, customers were issued invoices later than due payment dates. The bottleneck of writing out individual invoices manually to debtors triggered off a need for a computerized billing system that would remove the mundane tasks of generating invoices by hand.

Alpha approached Omega with a request for proposal 1 June 19x1. In their request for proposal to Omega, Alpha sought assistance from Omega for the following two objectives:

1. to identify detailed requirements for a computerized management system; and
2. to develop a billing system which should be simple to operate. The system should generate debit and credit notes based on the billing cycle and automatically compute the interests based on given rates.

Initial Negotiation

On 18 June, Omega presented the first proposal for a billing system. Omega estimated the project would take eight weeks to complete. Omega assigned four consultants to the project: V1, a senior consultant overall responsible for the project; V2, a mid-ranked consultant, and V3 and V4, two junior-ranked, associate consultants. The estimated cost of the system was between US$4000–$5000.

After hearing the first proposal, Alpha began to appreciate the potential power of computerization. Intense negotiations continued for four months. The continual dialogue and negotiation ultimately culminated

Table 12.1 Chronology of Events Culminating in a Failure

19x1	Major Events
1 June	• Alpha, a company in the services industry approached Omega, a management consultancy firm for consultancy assistance and software development for a billing system
18 June	• Omega offered first proposal: a system analysis and software development of billing system with user manual • Estimated time frame: 8 weeks • Estimated price: US$4–5K • Identified key personnel: V1–V4 (V1 senior consultant; V2 consultant; V3–V4 associate consultants)
6 July	• Omega offered second proposal • Scope increased to include additional two systems providing MIS exception reporting facilities to two other departments. In total three systems including data conversion from manual to computerized system • Time frame: 10 weeks • Estimated price: US$8–9K with out of pocket expenses capped at US$250
8 Aug	• Omega offered third proposal • Scope increased to include three more MIS systems for three other departments. In total six systems including data conversion from manual to computerized system • Estimated price: US$10–11K • No change in time frame
15 Aug	• Omega offered fourth proposal • No change in scope, but the five MIS systems plus the original billing systems were broken down to 10 separate modules • Time frame: 18 weeks • Estimated price remained at US$10–11K • V4 left, V5 joined
23 Oct	• Alpha confirmed acceptance of terms of Omega's proposals dated 6 July–15 Aug. 19x1 • Payment schedule: 70 per cent over five equal instalments on monthly basis effective Sept. 19x1. 20 per cent payable upon completion of assignment; 10 per cent payable after three months warranty period • Estimated time frame silent; estimated pricing unchanged at $20–22K
30 Nov	• Omega revised delivery schedule
19x2	
1 Jan	• V2 left; V6 joined
23 Jan	• Omega reported cost overruns amounting to US$65K • Suggested ex-gratia compensation
7 Feb	• Alpha clarifies position on direct compensation: will consider ex-gratia compensation only at end of contract
June–Aug	• Two trainees, undergraduate students on industrial attachment, were assigned to develop project

Table 12.1 (cont.)

19x2	Major Events
15 Nov	• Omega waiting for Alpha to sign off two modules
1 Dec	• Omega waiting for Alpha to sign off two more modules
28 Dec	• Alpha informed Omega of problems in some of the modules • Alpha refused sign-off
31 Dec	• V1 left. V6 took charge
19x3	
2 Jan	• Omega brought to Alpha attention outstanding sign-offs and deliverables
1 Mar	• V6 left, and V7 took charge
15 Mar	• V7 left and V8 took charge • Omega identified a total of 19 modules, 10 more than the original number • Omega delivered 14 modules all awaiting sign-off (commissioning) • Five modules yet to be delivered
31 Mar	• Omega issued stern warning letter demanding payment • Omega billed Alpha for about US$120 000, more than ten times the original contracted price

in a fourth proposal which expanded the original scope considerably. Added to a simple billing system were five MIS offering exception reporting facilities in each of the five major divisions of the firm. Omega then divided the six systems into ten separate software modules.

By October 19x1, four months after the initial proposal, the two parties signed an agreement for the project. Omega estimated that the project based on the fourth proposal would be completed within an 18 week time frame (i.e., around mid-February 19x2). Alpha was pleased with the time frame. It meant that they would be able to automate their internal operations by the first quarter of 19x2.

The estimated cost of the fourth proposal was around US$11 000. Alpha agreed to a staggered payment schedule. Alpha would pay 70 per cent of the cost over five equal instalments on a monthly basis with effect from September 19x1; 20 per cent payable upon completion of the assignment, and 10 per cent payable after three months warranty period.

Major Events Leading to Estrangement of Relationship

Three major events occurred between October 19x1 and March 19x3, the period between the original agreement and the final estrangement of the contractual relation.

First, on 30 November 19x1, about a month after the original agreement, Omega informed Alpha that they would not be able to deliver within the 18 week time frame. No concrete deadline was negotiated between the two parties.

Second, on 23 January 19x2, three months into the analysis and design of the system, Omega reported to Alpha substantial cost overruns amounting to US$65 000, about six times the cost stated in the original agreement. Omega demanded *ex-gratia* compensation. On 7 February 19x2, Alpha replied that it would consider *ex-gratia* compensation only at the end of the contract when Omega delivered a workable system.

Third, two key personnel on the project, V1 and V2, resigned from Omega leaving a vacuum in the leadership for the project. They were replaced by less experienced staff. During the summer months of June to August, two undergraduate students were employed on a temporary basis to expedite the development of the system.

Towards the end of 19x2, Omega finally delivered a number of completed modules to Alpha for commissioning or sign-offs. However, Alpha refused to sign off on any of the modules because none of them were fully operational.

The Estrangement

In March 19x3, some 18 months later, Omega issued a stern notice to Alpha demanding compensation for work completed. The bill amounted to around US$120 000, more than ten times the original agreed-upon amount of US$11 000.

As part of its justification for the revised fee, Omega claimed that Alpha continued to expand project scope beyond the original agreed upon specifications. Omega claimed to have developed a total of 19 system modules, nine more than the ten modules originally agreed upon. The notice also blamed Alpha for the colossal time slippage. According to Omega, Alpha was extremely non-committal and very tardy in signing off or commissioning modules that had been completed.

INTERPRETATION OF THE CASE

The case brought to the surface critical hierarchical elements missing in the contract that led to the demise of the outsourcing project. Table 12.2 applies the hierarchical elements framework to the case. Below we discuss in greater detail missing elements and their implications.

Table 12.2 *Case Diagnosis*

Hierarchical Elements	Case Diagnosis
Authority Relations	
(a) Explicit assignment of responsibilities of both Alpha, the client, and Omega, the contractor	None
(b) Explicit assignment of authority for authorizing scope changes	None. Alpha users continue to demand more modules. By March 19x3, Omega claimed a total of 19 system modules requested by Alpha, as opposed to the 10 agreed upon in the original contract
(c) Authority over price adjustments in projects	Alpha agreed verbally to *ex-gratia* payments after delivery to compensate for severe cost-overruns
(d) Authority over assignment of specific personnel and change in personnel	None. Project suffered serious turnover problems from Omega consultants. Three of the four original team members left within the 21 month period (see Figure 12.1)
(e) Right to audit work-in-progress	None. Alpha had no knowledge of right to audit work-in-progress
(f) Right to cancel project	None
Rule-based Incentive Systems	
Rules for punishing delays or giving bonuses for early completion	None
Standard Operating Procedures	
(a) Progress reports	Sporadic
(b) Regular meetings to discuss problems	Rare. A major meeting was convened between senior members of Omega and Alpha on 23 Jan 19x2. Omega reported cost overruns amounting to five times the original cost
Non-market Based Pricing Systems	
Pricing based on cost recovery considerations	The major fee for the major systems developed was computed on a fixed price basis. Cost recovery only for a small sum of incidental expenses such as travel. A cap of US$250 was placed for incidental expenses
Dispute Resolution Mechanism	
Informal (private grievances and third-party mediation/arbitration)	Omega raised attention of Alpha to non-payment, but Alpha continued to withhold payment for incomplete and unworkable system. Omega then served formal notice as first step towards legal recourse

Authority Relations

Based on the case, three major authority elements surfaced as being highly critical:

1. *Assignment of duties and responsibilities.* Explicit delineation of roles, duties, and responsibilities between Alpha and Omega was glaringly missing. Given that Alpha had had no prior working relationship with Omega, it is paramount that parties lay down clear lines of authority, accountability and responsibilities.
2. *Project scope.* Given the unwieldy and inevitable evolution of any system, a care and systematic authority system for approving scope changes is important. Changing scope generates rippling effects on cost structure and time schedules. In outsourcing contracts, clients should monitor scope changes carefully, otherwise they face the consequences of delayed schedules and escalating costs.
3. *Authority over personnel changes.* Contractors in outsourcing services often adopt a strategy of putting forward the best, most experienced people for contract proposal. As the project progresses, the experienced people are pulled out of the projects to pursue or deliver new business. In their place less experienced staff are assigned to complete existing projects.

In software outsourcing, a client's right to monitor and authorize personnel changes to the outsourcing team is vital because productivity of good and bad IT professionals can vary in the order of magnitude of 1:25 (Jones, 1985).

In this case, turnover was especially high—Figure 12.1 illustrates the excessive staff changes. Except for V3, a junior associate consultant, all original team members, including the senior consultant V1, left the project. At the peak of development, Omega resorted to deploying two student trainees to develop the system.

Alpha needed to ensure that Omega continued to provide the best people in the project. If the primary objective of outsourcing is to tap the distinctive competences of Omega, Alpha must ensure the quality of personnel on the project, and scrutinize any changes made to the project team. This requirement was also noted by Lacity and Hirschheim (1993).

Rule-based Incentive System

When successfully implemented, rule-based incentives act either to reward or punish the contractor to ensure timely delivery of system. In

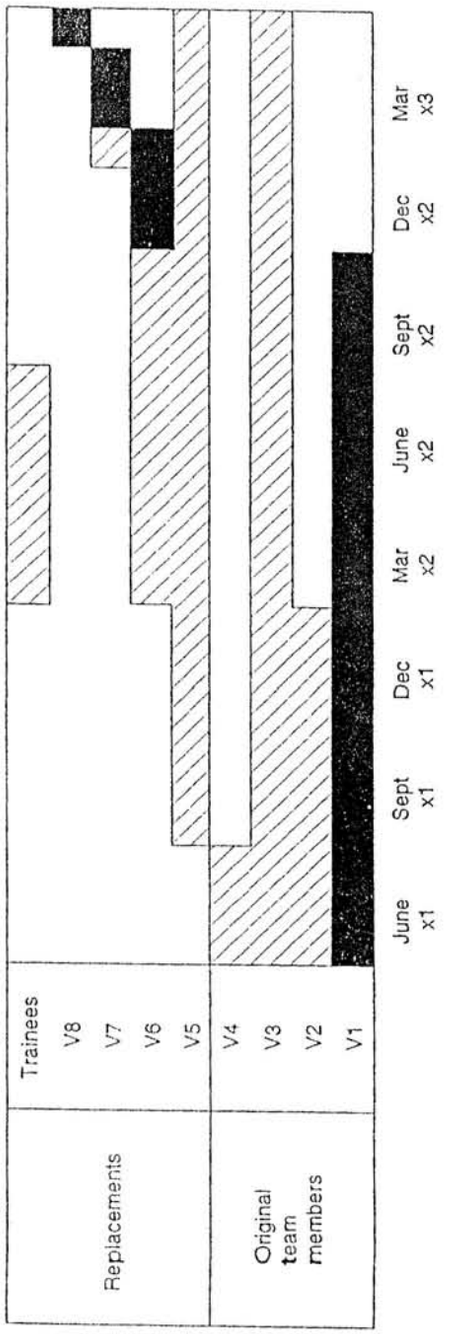

Figure 12.1 *Omega Staff Turnover Profile*

this case, Alpha and Omega had originally agreed upon a rule-based system based on instalment payment. However, the instalment plan was ineffective. First, the incentive plan was tied more to behaviour (rewarding Omega for time put into the project) rather than to outcome (rewarding for delivering modules). Based on the plan, Omega would be paid 70 per cent of the total cost over five equal instalments effective September 19x1, regardless of delivery of completed modules. Second, the instalment plan lost its effectiveness as an incentive when project costs soared and time schedules slipped badly. Even if Alpha had paid according to schedule, the payment amounting to 70 per cent of the original cost were minuscule compared to the colossal cost overruns. Alpha had wanted the cut-over of the computerization effort by the beginning of 19x2. If the cut-over deadline was critical, sufficient rewards and punishments should have been put in place to tie directly to outcome—the timely delivery of completed systems. In an extreme case, Omega may be asked to reduce the purchase price of the system at a certain rate as liquidated damages to Alpha.

Standard Operating Procedures

Outsourcing projects do not enjoy the luxury of hierarchies and bureaucracies. Conscious effort must be made to document all decisions so that someone else can examine the documents and reconstruct these decisions, especially when those working on the project leave.

Omega was confident that its proven methodology would offer routines necessary for conducting, monitoring and regulating the projects. However, Alpha did not understand the role they played in the methodology. Especially in the case where neither party has had prior mutual working relations, it is important that routines be institutionalized and made explicit. Procedures embedded in the methodology would possess no disciplining muscle if Alpha neglected their responsibilities out of ignorance of the methodology. Ignorance of the purpose and procedures of the methodology also meant that Alpha could not gauge whether Omega had breached procedures laid out by the methodology.

Non-market-based Pricing System

This case demonstrates how a poorly estimated project can go awry. Omega grossly underestimated the cost of the project. Although project scope increased at least fivefold (from a billing system to six major MIS systems), the fees increased only slightly over twofold (from an original US$4000–5000 to US$11 000).

In retrospect, Alpha should have ensured a better way of estimating the cost of the project by requesting for more than one bid from various vendors. Bids from various vendors provide valuable benchmarking data. For example, some companies use bids from leading consulting firms as industry benchmarks. They then adjust the fees proportionately in line with those signalled by the benchmark firm.

Parties to the contract must also adopt non-market pricing or cost recovery mechanisms so that the contractor can recoup its costs as project scope expands. To mitigate price uncertainty, Alpha and Omega could have specified an upper limit for costs exceeding the original fixed price of the system. An example of such a clause reads as follows:

> "If the verifiable actual cost of developing the System exceeds $25 000, GTC shall invoice MMRF for half of such cost exceeding $25 000, up to a maximum of $18 000, upon MMRF's accepting the system. ..." (American Bar Association 1987, p. 982).

In the context of expected scope changes, the above clause would strike a reasonable balance between price risk for Alpha and compensation risk for Omega. A mixed fixed price and cost recovery mechanism would provide sufficient inducement for both Alpha and Omega to enter into and to survive the outsourcing contract.

Alternative Dispute Resolution Mechanisms

Omega and Alpha had resorted to private grievance redress before the final estrangement of their business relationship. Sporadic formal meetings were called by the leader of the project team from Omega to discuss with Alpha issues concerning non-payment and escalating costs.

Unfortunately, private grievance procedures were not very effective because the frequent and abrupt changes of Omega personnel in authority impeded the development of mutual understanding. Generally, mutual understanding and adaptation occur through the social interaction that accompanies sustained joint work activities by the same members of an interorganizational team. As social interaction intensifies, members develop implicit standards of expected behaviour and mutual understanding. This overlay of social relations on a purely contractual relationship plays a crucial role in promoting alternative dispute resolution mechanisms.

Omega leadership changed hands four times during the course of the outsourcing contract (see Figure 12.1). Consequently, the contractual relationship between Omega and Alpha did not have the opportunity

to develop into the good, trustworthy, socially embedded relationship necessary to fend off potential misunderstandings and disagreements. In fact, the final stern notice originated from a new senior consultant of Omega (V8) who was assigned primary responsibility for the Alpha project two weeks before the notice was served. V8 had had no prior contact with Alpha and did not empathize with the complexities of the project.

Team rebuilding and sustained effort on recommitment in the light of turnover cannot be over emphasized. In the landmark outsourcing contract between Kodak and IBM, the partnership had appointed two new CIOs and experienced an almost total turnover of the co-ordination team from both Kodak and IBM. According to McFarlan and Nolan (1995), the key success factor of the Kodak-IBM partnership is its emphasis on team building. The team interacts frequently and attends team-building retreats to ensure that turnover of personnel does not erode the stable relations of trust, obligations, customs and values of the outsourcing team (Willcocks and Kern, 1997). The issue of partnering was pursued in more detail by Robert Klepper in Chapter 10.

CONCLUSION

The purpose of this chapter was to describe and diagnose a failed contract to illuminate the added complexities of software outsourcing. To accomplish our objective, we relied on a diagnostic framework developed by Ang and Beath (1993) which prescribes embedding hierarchical elements in outsourcing contracts to overcome the shortcomings of outsourcing.

The analysis of the Alpha-Omega case shows that the outsourcing failure can be attributed to the lack of attention by the outsourcing parties to critical hierarchical elements. Major oversights included lack of clear lines of authority over sanctioning changes in project scope; lack of client authority over selection and changes in team members from the consulting firm; lack of punitive incentive systems for delays; lack of communication of the importance of standard operating procedures such as formal sign-offs and client audit; and unrealistic market-based, fixed pricing for a project of uncertain and uncontrollable scope.

The chapter contributes to outsourcing practice by demonstrating how one can conduct a diagnosis, analysis and postmortem on failed external software contracts. However, it behoves practitioners to recognize that although hierarchical or control elements mitigate the risk of outsourcing failure, they do not of themselves necessarily lead to outsourcing success.

In terms of future research, the application of the framework to the case surfaced critical contingencies that may derail outsourcing projects. One significant contingency is the high turnover rate experienced in the software industry. This makes it even more important to carry out the sort of extended vendor analysis detailed in Michell and Fitzgerald (1997). Future research should also examine the impact of personnel turnover on contractual parties ability to learn and improve outsourcing performance. The challenge is to identify and put in place mechanisms that foster inter-organizational learning, that is, the ability to translate largely tacit knowledge in software development to explicit routines and procedures available to both parties in the outsourcing relation. This, of course, gives an additional twist to the argument detailed earlier in Chapter 4.

REFERENCES

Abdel-Hamid, T. and Madnick, S. E. (1991) *Software Project Dynamics: An Integrated Approach*, Englewood, NJ: Prentice Hall.

Altinkemer, J., Chaturvedi, A. and Gulati, R. (1994) Information Systems Outsourcing: Issues and Evidence, *International Journal of Information Management*, 14, 252–268.

American Bar Association (1987) Software Contract Forms, American Bar Association, Section of Science and Technology, Chicago.

Ang, S. and Beath, C. M. (1993) Hierarchical Elements in Software Contracts, *Journal of Organizational Computing*, 3, 3, 329–362.

Davis, G. B. *et al.* (1992) Diagnosis of an Information Systems Failure: A Framework and Interpretive Process, *Information and Management*, 23, 293–318.

Jones, C. (1985) *Programmer Productivity*, New York: McGraw-Hill.

Lacity, M. and Hirschheim, R. (1993) *Information Systems Outsourcing: Myths, Metaphors, and Realities*, Chichester: John Wiley and Sons.

Lacity, M., Willcocks, L. and Feeny, D. (1995) IT Outsourcing: Maximize Flexibility and Control, *Harvard Business Review*, May–June, 84–93.

Lyytinen, J. and Hirschheim, R. (1987) Information Systems Failures—A Survey and Classification of the Empirical Literature, in Zorkoczy, P. I. (ed.), *Oxford Surveys in Information Technology*, 4, Oxford University Press, Oxford, 257–309.

Markus, M. L. (1983) Power, Politics, and MIS Implementation, *Communications of the ACM*, 26, 6, 430–444.

McFarlan, F. W. and Nolan, R. L. (1995) How to Manage an IT Outsourcing Alliance, *Sloan Management Review*, Winter, 9–23.

Michell, V. and Fitzgerald, G. (1997) The IT Outsourcing Marketplace: Vendors and Their Selection, *Journal of Information Technology*, **12**, 3, 130–148.

Mylott, T. R., III (1995) *Computer Outsourcing: Managing the Transfer of Informa-*

tion Systems, Englewood Cliffs: Prentice Hall.
Richmond, W. B., Seidmann, A. and Whinston, A. (1992) Contract Theory and Information Technology Outsourcing, *Decision Support Systems*, 8, 5, 459–477.
Sitkin, S. B. (1992) Learning Through Failure: The Strategy of Small Losses, in Cummings, L. L. and Staw, B.M. (eds), *Research in Organizational Behavior*, Greenwich, CT: JAI Press, Inc.
Stinchcombe, A. L. (1990) Contracts as Hierarchical Documents, in Stinchcombe, A. and Heimer, C. (eds), *Organizational Theory and Project Management*, Oslo, Norway: Norwegian University Press, 1985; reprinted in Stinchcombe, A. (1990) *Information and Organizations*, Berkeley, CA: University of California Press, Chapter 6.
Whang, S. (1992) Contracting for Software Development, *Management Science*, 38, 3, March, 307–324.
Willcocks, L. and Kern, T. (1997) IT Outsourcing as Strategic Partnering: The Case of The Inland Revenue, Proceedings of The Fifth European Conference in Information Systems, Cork, Ireland, June.