13

THE PROCESS OF OFFSHORE SOFTWARE DEVELOPMENT
Preliminary Studies of UK Companies in Malaysia

Aini Aman and Brian Nicholson
School of Accounting and Finance, University of Manchester, UK

Abstract: There are many studies of information technology outsourcing but very few discuss in depth the process of offshore software development outsourcing. The aim of this research is to understand the important issues and strategies in managing activities and relationships among software development teams involved in offshore outsourcing. The study involves UK companies who are engaged in offshore outsourcing of software development to Malaysia either through joint ventures or fully owned subsidiaries. There are an increasing number of software firms choosing Malaysia as a venue for software development. The study contributes to the growing body of literature on offshore software development outsourcing by building a synthesized conceptual framework for global software outsourcing. This is derived from concepts found to be important in previous studies. The framework is illustrated using data from two ongoing case studies.

Keywords: Communication, Control, Culture, Globalization.

1   INTRODUCTION

Offshore software development outsourcing takes place when a customer uses a contractor in another to perform a software-related task, such as programming or data entry (Phillips, 2002). Amoribieta et. al, (2001) explains that the development of custom software is different from most other business activities since it requires a detailed understanding of business processes and the way information technology (IT) supports them. Managing offshore software development is highly complex because distance creates difficulties in coordination and control, knowledge transfer and communication (Carmel and Agarwal, 2001). Potential problems related to

offshore software development include cross-cultural issues, language barriers, time zone differences, political instability and unreliable telecommunications infrastructure (Gao et. al, 2002; Walsham, 2002). Building an offshore partnership requires much effort and delicate handling by senior managers. The field of software engineering in this domain is relatively new and procedures for quality control and project management, though developing very fast, have yet to evolve fully.

Recent trends show that software developers are going offshore to develop and maintain their software as a strategy to improve cost control, product quality, product development, schedule reduction and focus on core business activities (Amoribieta et. al, 2001). Carmel and Agarwal (2001) outline some potential advantages of offshore software development over in country options as the ability to tap into specialized talent, facilitate a global presence and achieve cost saving of up to 60%. Other general reasons for adopting an offshore strategy include improved product quality, to engage in specialised product development, schedule reduction and to focus on core business activities (Amoribieta et. al, 2001). Among the favourite locations for offshore outsourcing are the first tier vendor nations often referred to as the "3Is" of India, Ireland and Israel. Popular second tier venues for a range of offshore software services are Singapore, Russia and Philippines. Other countries offering various types of IT services plus cheap labour costs are South Korea, Vietnam, Hungary, Czech Republic, Malaysia, Argentina, Brazil, Chile, South Africa, Poland, China and Mexico (Amoribieta et. Al., 2001; Heeks, 1999; Heeks and Nicholson, 2002).

There are very few in-depth longitudinal studies showing how companies achieve the benefits and overcome the problems of offshore outsourcing. The aim of this research is to understand how companies are managing activities and relationships in offshore software outsourcing. The study is concerned with UK companies engaged in offshore outsourcing of software development to Malaysia either through joint ventures or fully owned subsidiaries. There are an increasing number of software firms choosing Malaysia as a venue for software development. The study draws on a framework, which draws together concepts found to be important in previous empirical studies of offshore software outsourcing. The paper contributes to the growing theoretical body of literature on offshore software development outsourcing by building a synthesized conceptual framework for global software outsourcing. The practical contribution of this paper is in illustrating the framework using preliminary data from two ongoing case studies of offshore outsourcing.

The paper is organised as follows: in the next section the conceptual framework is presented which consists of a literature search drawing together literature on offshore software outsourcing. This is followed by a

discussion of the research design for the case studies. Section four provides a description of the two cases followed by analysis drawing on the conceptual framework. The final section summarises the analysis and considers the contribution of the paper as well as outlining future work.

2   CONCEPTUAL FRAMEWORK

In order to select and develop an initial framework for the inquiry previous research was examined. Relevant frameworks and models related to outsourcing of software development are concerned with information technology outsourcing decisions (Willcocks and Fitzgerald, 1994; Apte and Mason, 1995); IT issues confronted by multinational corporations (Deans and Ricks, 1991); organizing existing knowledge about globalization of the software industry (Kim et al., 1989), division of labour in global software development (Meadows, 1995) and selection of projects and sites suitable for offshore software development (Ravichandran and Ahmed, 1993). Other frameworks are explicitly concerned with management of the process of offshore outsourcing of software development (Mitra and Narasimhan, 1996; Heeks et al., 2001).

Heeks et al. (2001) developed a dimensional framework on the basis of their empirical research and other studies of offshore software outsourcing. The COCPIT framework was chosen as a basis for conceptual frame development. The COCPIT categories were derived from empirical evidence from North America, UK, Korea and Japan based companies outsourcing to India involving over 200 interviews over a four-year period. The framework demonstrates that issues in offshore software development can be analyzed as the minimization of "synching gaps" between client and offshore vendor along six 'COCPIT' dimensions: **C**oordination/control systems (management coordination and control system), **O**bjectives and values (sharing the same objectives for their relationship and bringing the same values to that relationship), **C**apabilities (the developer's capabilities such as human capabilities matching the requirements of the customer); **P**rocesses (synchronization of work processes), **I**nformation, and **T**echnology (information completeness and technology usage). Within the COCPIT categories, successful relationships with a high degree of congruence achieved between vendor and client is termed 'synching'. The unsuccessful relationship with a low degree of congruence is termed 'sinking'. Using the framework Heeks et al. (2001) suggest that creating congruence or "synching" between client and developer along the COCPIT dimensions may lead to the higher project success rates in offshore software outsourcing. They explain that synching is also a means of reducing risks and costs. Their

case studies also suggest that synching strategies have been poor at dealing with three overlapping issues; tacit knowledge, informal information and culture that indirectly affects all COCPIT dimensions.

COCPIT is useful to sensitise researchers and practitioners to the issues in offshore software development outsourcing. However, the framework does not explain each dimension in detail. For instance, Heeks et al. (2001) do not explain how and when each dimension influences the relationships between developers and clients. Furthermore, COCPIT does not explain other factors that might influence the existence of each dimension. Other weaknesses of this contingency framework are that the authors do not fully identify theoretical concepts within each dimension to facilitate in depth analysis. To overcome some of these limitations, the authors enhanced the COCPIT framework with other literature into each dimension and illustrate it using cases. The new framework may serve as a useful approach to systematically address the issues that arise in the context of offshore outsourcing of software development.

2.1 The Revised Framework

The revised framework is based on COCPIT with the extension on the explanation of each dimension based on existing literature. Table 1 lists current literature on each dimension of COCPIT. Although it is accepted that these categories overlap, the revised framework is used in the empirical study as a guide to data collection. Subsequently, the associated theory will be used to make sense and explore the empirical data and assist in explaining the issues encountered.

Table 1. Related Literature on each dimension of COCPIT

COCPIT	Related Literature
Coordination & control	Henderson and Lee (1992), Carmel and Agarwal (2001)
Objective & values	Hofstede (1991), Ramarapu and Parzinger (1997), Herbsleb and Moitra (2002), Walsham (2002)
Capabilities	Grimaldi and Torrisi (2001), Zhuge (2002), Gao, Itaru and Toyoshima (2002)
Process	Curtis, Krasner and Iscoe (1988),Rautiainen, Lasenius and Sulonen (2002)
Information	Heeks, Krishna, Nicholson and Sahay (2002), Herbsleb and Moitra (2002)
Technology	Whittaker and Schawarz (1999), Choi and Lee (2002)

In the *coordination and control* dimension of COCPIT congruence may be achieved when client and developer use corresponding management and control systems. Carmel and Agarwal (2001) explain that *coordination* is the

act of integrating each task with each organizational unit, so the unit contributes to the overall objective, which requires intense and ongoing communication while *control* is the process of adhering to goals, policies and contracts, standards or quality levels. Carmel and Agarwal (2001) stress how distance can affect communication, which in turn can reduce coordination and control effectiveness. Developers not located together have very little informal, spontaneous conversation, which may help actors remain aware of dynamic progress. Failure in this respect might result in misalignment and rework. Synchronization requires commonly designed milestones and clear entry and exit criteria. Henderson and Lee (1992) combine research on managerial control and team-member control in order to explore a range of control behaviours that can affect the performance of an offshore software development team.

In the *objectives and values* dimension of COCPIT; Heeks et al. (2001) suggest that to achieve congruence, vendor and client should share the same objectives for their relationship and bring the same values and culture to that relationship. The issue of culture is particularly pertinent to offshore software development. Herbsleb and Moitra, (2002) explain that differences in cultural background sometimes lead to serious misunderstanding. Cultural barriers may hamper communication among the user, designer and programmer and can decrease productivity thereby affecting the development of quality software (Ramarapu and Parzinger, 1997). Some authors suggest different attitudes towards authority and hierarchy in the Eastern (less open and les confrontational) and Western culture (more open and extrovert) (Hofstede, 1991). This type of research is useful to sensitise the researcher to potential cross-cultural issues. However, more sophisticated analysis was performed by Walsham (2002) using concepts drawn from structuration theory. The theoretical approach is used to analyze cross-cultural conflict and contradiction, cultural heterogeneity, detailed work patterns and the dynamic nature of culture.

The COCPIT dimension of *capabilities* suggests that congruence can be achieved when human capabilities, skills and knowledge match the requirements of the clients. Grimaldi and Torrisi (2001) analyse the process of knowledge codification and the division of labour between software firms. They focus on the nature of tacit versus codified knowledge that firms share and exchange with other partners by means of collaborative agreements. Zhuge (2002) presents a notion of knowledge flow and the related management mechanism for realizing and ordered knowledge sharing and cognitive cooperation in a geographically distributed team software development process. Gao et al. (2002) suggest that to deal with complexities of knowledge, firms should try to simplify tasks and to

coordinate activities of partners across time and space by standardizing various processes of knowledge transfer.

The *process* dimension in COCPIT; suggests that congruence is achieved when the vendor and client manage work processes and software development methodology effectively. Rautiainen et al. (2002) present a general framework for managing software product development. The framework combines business and process management through four cycles of control: (1) strategic release management; (2) release project management; (3) iteration management and (4) mini-milestones are used for daily or weekly task scheduling and monitoring to get an indication of system status during development. On the other hand, Curtis et al. (1988), used layered behavioural model in order to understand how different tool, methods, practices and other factors actually affect the processes that control software productivity and quality. The model emphasizes factors that affect psychological, social and organizational processes. The model focuses on the behavior of those creating the artifact, rather than on the evolutionary behavior of the artifact at each individual level, team level, project level, company level and business milieu.

The *information* dimension of COCPIT describes that congruence is achieved when the client and developer have access to the same information; for example, information relating to project requirements and timescales. The share of information requires both explicit and tacit knowledge. Herbsleb and Moitra (2002), suggest that without effective information and knowledge-sharing mechanisms, managers cannot exploit global software development benefits. Herbsleb and Moitra (2002), also suggest that to prevent assumptions and ambiguity and to support maintainability, documentation must be current and reflect what various teams are using and working on. In addition to documenting the various artefacts, updating and revising the documentation is equally important.

The COCPIT dimension of *technologies* is concerned with the software and hardware platforms for development work. Whittaker and Schwarz (1999) found that technology adept groups prefer to use what seem to be outmoded "material" tools in critical projects, despite a wealth of electronic group tools for coordinating the software development process. They also found that the medium of the schedule has a major impact on coordination problems. Choi and Lee (2002) suggest that a "system oriented" strategy and "human strategy" are important determinants of success of the offshore software development process. The system-oriented strategy emphasizes codified knowledge and focuses on codifying and storing knowledge via information technology to share, access and use knowledge formally. The human-oriented strategy emphasizes dialogue through social networks and

person-to-person contacts focussing on acquitting knowledge via experienced and skilled people to share knowledge informally.

3 RESEARCH DESIGN

This research adopts a qualitative design as it aims to highlight the experiences of individual organizations in the process of offshore software development. The overall research strategy could be categorized as interpretivist guided by the knowledge of reality as socially constructed by individual human actors (Walsham, 1995). An interpretivist approach tries to understand the perspective of different actors involving multiple perspectives. This requires in-depth qualitative data including views of members in the software development teams concerning the process and management issues in the contexts within which the process is taking place. The methodology draws on contextualism (Pettigrew, 1987; 1990), which was found useful by Walsham and Waema (1994). The resulting interpretive case study (Walsham, 1995) is hoped will explain how and why contextual conditions and project management tactics and strategies interact. Pettigrew (1987) states that the researcher should come to the field situation equipped with a number of theoretical concepts, which could be used to analyse the data. The researchers using the revised COCPIT framework discussed in section 2, are equipped with range of theories to help and understand issues in offshore software development outsourcing.

The companies chosen for this study are currently engaging in offshore outsourcing software development. The sites were selected firstly through a literature search that indicated a number of UK companies who were engaged in outsourcing software development to Malaysia. A list of companies was contacted through telephone and fax. Data collection is through a variety of methods: unstructured and semi-structured interviewing, documentation review and observation. Triangulation or using various techniques of data collection is particularly beneficial as it provides multiple perspectives on issues, supplies more information on emerging concepts, allows for cross-checking and yields stronger substantiation of constructs (Eisenhardt, 1989; Glaser and Strauss, 1967; Pettigrew, 1990; Orlikowski, 1993).

Information was gathered on the software industry in Malaysia and two case companies known as Das Ltd and Ace Ltd. The real names of the companies are disguised for confidentiality. The study is ongoing and at this preliminary stage there have been a total of ten interviews with programmers and managers at both companies in UK and Malaysia. Interviews lasted for at least one and half-hours. The interviews were taped and transcribed and

subsequently summarised Data from both companies were gathered and grouped into the dimensions of COCPIT using a data display method (Miles and Huberman 1994) which also enabling cross case analysis. Analysis of the data from interview transcripts in relation to the theory as suggested by Klein and Myers (1999) revealed the issues that exist in the process of offshore software development outsourcing and tentative explanation drawing on relevant theory.

4   CASE BACKGROUND: DAS AND ACE

Das and Ace are software development companies and have their headquarters in UK with wholly owned subsidiaries in Malaysia. Both companies are involved in offshore software development outsourcing. Das is a custom e-business software solution provider while Ace is a process plant engineering software services provider. In general, both companies are in the process of setting up software development teams in Malaysia for incoming projects. Both companies are currently recruiting software programmers in Malaysia for their incoming projects. Das is using Lotus and Domino while Ace is using Microsoft as a based for software development. Das is a medium-sized company established in 1994 with approximately fifty staff in three countries (UK, Malaysia and Bangladesh). They have a wholly owned research and development centre in Bangladesh. The development centre in Bangladesh serves both UK and Malaysia. Since 1997 Das has been using software development teams in Bangladesh. Das's office in Malaysia is currently tasked with client facing, consultancy and local project delivery. Ace is a large sized company established in 1983 with 340 staff in 21 locations across Europe and the Middle East, Asia Pacific and North America. Ace is currently using software development teams in UK and India. Ace's office in Malaysia is headquarters for administration in Asia Pacific with small project development capabilities to provide direct support for local clients.

The software industry in Malaysia to date has been import oriented and reliant on foreign base technologies. Overall, the information technology market is expected to grow by 10% to about RM 8 billion in 2003. The various Multimedia Super Corridor (MSC) flagship applications in Malaysia are expected to provide significant growth opportunities for the IT industry. Most of the MSC's companies are in the software development sector followed by Internet based business, content development, telecommunication, data centre, system security, consulting and incubators. The breakdown of European MSC approved companies' shows that most are from United Kingdom (Multimedia Development Corporation, 2002).

Likewise, the present size of software industry in Bangladesh is very small. Only a few firms are involved in the export of software and data entry business. The size of IT industry is estimated at around US$ 150 million, which is growing at more than 20% each year. In contrast, the Indian Software industry is the fastest growing industry in India. Indian software sector is forecast to continue growth at 50% per annum over the next few years.

5 ANALYSIS

In this section, selected preliminary empirical data will be explored using the revised framework outlined in section 2 above. The analysis will use the theoretical framework (COCPIT and associated literature) to illustrate the issues raised in offshore software development outsourcing.

5.1   Process, Coordination and Control

In the *process, coordination and control* dimension of COCPIT, a significant theme in the data collected so far is that of the pivotal importance and skills of those assigned to "straddler" roles. In both cases key individuals were seen to be instrumental in managing knowledge transfer and facilitating disembedding of coordination and control processes across time and space (Giddens, 1990) between on and offshore groups. Although this liaison person is regarded as important in managing the offshore process, little is understood about the skills and qualities of this person and the importance of the role in effective software development. A Das manager describes the importance of a "technical guide" in ensuring that specifications are understood and the process is controlled:

> *'In UK, we do all the upfront work with the customers over the analysis of the requirements. We manage the project here with our customers and with them (software development teams in Bangladesh). We will send a specification to one of the developers who act as a technical guide to them (the Bangladesh software development teams) to make sure we get the solution what we want. Then they will do all the programming and we will do the online testing with them. When they have completed it, they deliver back to us and we will deliver the final product to customers'.*

The liaison person or "technical guide" is a key personality in the "synching" between the onshore and offshore groups. This person is responsible for interpreting requirements and re interpreting and instituting

control locally. Henderson and Lee (1992) stress that team member control (in this case; using technical guide) is necessary in software development practice because tasks in planning and design are difficult in nature. Henderson and Lee (1992) suggest that effective teams should have a manager with the skills and capabilities to influence how work is accomplished and to influence terms to behave in accordance with organizational gaols. In Ace, the role of project manager has full control on the project and he is a key person to liase with software engineers in offshore business units. In Ace, the project manager role is multifaceted as indicated by the company president:

> *'The way we manage the system is that we have a project manager. The project manager will define what we will develop. He gets into customers and brings back information about how the system will be defined, what improvement is needed for the future such as particular changes for certain industry, strategic program, define the kind of development project. Software development teams work close with the project manager. The project manager will allocate resources for them in UK and India. The software engineer cannot decide what to change but can decide the best way to do on the programming side and they can only make recommendation. The project manager is the key person who manages the project deliver the final product to business unit like us. The project managers are based in UK and they are in very close contact with project. One of the key tasks of project manager is to make sure that they deliver the product on time'.*

Despite the use of liaison persons in coordinating and controlling work offshore, distance can affect the way software is developed. This may reduce coordination and control effectiveness and might result in misalignment and rework (Carmel and Agarwal, 2001). The director of Das raised the issue as follows;

> *'Because we have a couple of project recently not going as well as we plan and it is painful to put it right, we recognize that a lot of the problems came from the project process. When we go to the next level down, we do not necessarily get a consistent delivery; every single project, every single time. We have been spending a lot of time helping them (software development teams in Bangladesh). We can't afford to have time helping them because that will delay the project here (in UK).*

5.2   Objectives and Values

In this dimension of COCPIT, the issues raised are concerned with cultural barriers. The director of Das compares the culture in UK and Bangladesh that sometimes leads to project delay:

> *'Here (in UK), the environment is very open where people have great trust on each other, ask questions very openly and freely. It is an informal collaboration where people do not think that asking question is a sign of failure. Bangladesh is a completely different situation partially because of culture and largely because of their education system. If we give so much freedom to the software development team in Bangladesh, they don't know what to do. The team wants specific instruction. The developers in UK have developed common sense of how things should work while people in Bangladesh are working in isolated world We notice that most of the time, humour is left out. We learn over the years that we have to develop specific techniques in dealing with them. Now, we have improved a lot because we have now got senior people there. What we know is that when we talk to the senior developer we notice that their knowledge and experience are very limited. We have to spend quiet a lot of time explaining to them'.*

In the case of Das, our respondents told us that Bangladesh teams are not willing to express opinions overtly especially if contradictory to those of senior staff. Therefore, there appears to be a culture gap that is hard to bridge. The problem of culture is less intense in Ace because most of their software development teams are British. At Das, aspects of the reported need for specific instruction, education, humour and issues of common sense may be usefully explored drawing on Walsham (2002) structurational analysis once there is sufficient empirical data and multiple perspectives.

5.3   Capabilities, Information, Technology

The problems of capabilities and information are discussed here in terms of sharing knowledge and training new programmers. As in Das, knowledge sharing is important as they rely very much on Bangladesh expertise. Das use most of expertise in Bangladesh but sometimes using expertise from India. One of the managers explains how they build up their experts, which is problematic and also suffers from cultural difficulties:

> *'We have expertise in Bangladesh who can share knowledge with the new software teams in Malaysia. Sometimes, we go to India and get*

people when we are short of workers. We realize that Indian developers bring severe problems. They interpret in their own way. The do not understand the business process very well. They have problems with UK staff too. The Bangladesh staff are used to UK culture while Indians are not used to it. When we first start a centre in Bangladesh, we used British staff to give training, set standards, procedures, coding etc. After two years the British staff were sent to Bangladesh to get training. We tend to do a lot of research. Once a week there will be staffs training. We come up with some topics that are beneficial to them and the company and do research on. Some times there is no time to train everybody. So, they just choose one small project say new language and study it on Saturday.

In contrast, Ace has more formal procedures in knowledge sharing and training new staffs. The project manger of Ace describes:

'The post of every new engineer and electrical staff in the company is made changeable. The purpose is to help he/she to be familiar with the company quickly. The development teams are going into intensive training, which is very different, and in timely change. Every year, we update training of new product for development teams because the development tools changes fast'.

To deal with complexities of knowledge, firms such as Ace try to simplify tasks by standardizing various processes of knowledge transfer (Gao et al., 2002). These standardized systems, often codified in manuals and databases, serve as points of reference to coordinate activities of partners across time and space. Such attempts to standardize are rarely unproblematic, and in tension with the need for flexibility at the 'local' level. As one of the managers in Ace indicates;

'We have concrete procedure and process on software development project because the software development task is very big. We need to be very rigid in testing, to make sure the system is stable, because we are the one that supposed to be doing it before we release it to our customer. We have a well-proven process anyway. We have it documented'.

While some degree of standardization is essential to enable global coordination, there is always the question of how much and what to standardize. A project manager in Das raises this issue;

'There is a lot of work to document what you have done. To document the work requires a discipline to ensure that the application is updated every

single time. We try to work out how to solve out most of this documentation problem'.

Documentation of systems by software developers is often regarded as a tedious task. In global software development, in addition to documenting the various artefacts, updating and revising the documentation is equally important. Herbsleb and Moitra, (2002) suggest that to prevent assumptions and ambiguity and to support maintainability, documentation must be current and reflect what various teams are using and working on.

The COCPIT dimension of *technologies* is concerned with the software and hardware platforms for development work. Both companies are using communication technology such as groupware applications in the software development process. The manager in Das describes the difficulties she has, especially on explaining certain procedures to the development teams in Bangladesh.

' I have been doing all of the work with the development team on line. They send me the application and I test it and I go online. If there is a problem we chat about it, doing it online and show the problems. We chat about it before we go to the customers to implement it. When you try to do things online and over the phone, you can make progress but ultimately teaching them down to absolute detail is a lot harder compared to doing it face to face'.

The biggest limitation Das face of using groupware is that they perceive it is not the same as face-to-face communication. Whittaker and Schwarz (1999) stress that electronic systems may reduce face-to-face communication, decrease awareness of the actions of other group members, and suffer from lack of visibility and performance.

6 CONCLUSION

The growth of offshore software development outsourcing indicates the need to better understand associated issues or problems. Literature reviews in the areas of offshore outsourcing indicate several frameworks to identify problems and issues in the area, which were discussed in section 2 above. Among the available frameworks, COCPIT was chosen as an all-embracing framework suitable for a processual study. However, this framework does provide sufficient detail or explanation of each dimension. Therefore, this paper has presented a revised framework drawing on other literature and illustrated it using preliminary data from two ongoing case studies. The new

framework serves as a useful tool to systematically address the issues that arise in the context of offshore outsourcing of software development. This new framework was used as a guide to data collection and to help make sense and explore the empirical data as well as assist in explaining the issues encountered. The revised framework discussed in section 2 enables the researchers to be equipped with range of theories to help to understand issues in offshore software development outsourcing.

The contribution of this paper is two fold. For other researchers, this study contributes theoretically by drawing together relevant and current literature in each of COCPIT dimensions. If used as a contingency framework, researchers and practitioners can identify and "diagnose" issues in managing successful relationships in offshore outsourcing according to the specifics of a situation. For practitioners, this study illustrates selected aspects of the process of offshore software development outsourcing using two cases within the dimensions of COCPIT. The understanding of issues in the process of offshore software development is important for practitioners to manage successful relationships between clients and developers, which in turn will contribute to the success of offshore outsourcing.

REFERENCES

Amoribieta et al. "Using a Global Services Delivery Model to Rapidly Develop Quality E-Business Software Solutions", *The McKinsey Quarterly*, (2), 2001.

Apte, U. and Mason, R.O. "Global Disaggregation of Information-Intensive Services", *Management Science*, (41:7), July 1945.

Carmel, E. and Agarwal, R. "Tactical Approaches for Alleviating Distance in Global Software Development", *IEEE Software Journal*, March/April 2001.

Choi, B. and Lee, H. "Knowledge Management Strategy and its Link to Knowledge Creation Process", *Expert Systems with Application,* (23), 2002, pp. 173-187.

Curtis, B., Krasner, H. and Iscoe, N. "A Filed Study of the Software Design Process For Large Systems", *Communications of the ACM*, (31:11), November 1988.

Deans, P.C. and Ricks, D.A. "MIS Research. A Model for Incorporating the International Dimension", *The Journal of High Technology Management Research*, (2:1), 1991, pp. 57-81.

Eisenhardt, K., M. "Building Theories from Case Study Research", *Academy of Management Review*, (14:4), 1989, pp. 532-550.

Gao, J. Z., Itaru, F., Toyoshima, Y. "Managing Problems for Global Software Production – Experience and Lessons", *Information Technology and Management*, (3), 2002, pp. 85-112.

Giddens, A. "Structuration Theory and Sociological Analysis", in Anthony Giddens: *Consensus and Controversy*, J. Clark, C. Modgil, and J. Modgil (eds.) Brighton, England: Falmer Press, 1990, pp. 297-315.

Glaser, B. and Strauss, A. *The Discovery of Grounded Theory, Strategies for Qualitative Research*, Chicago: Aldine, 1967.

Grimaldi, R. and Torrisi, S. "Codified-Tacit and General-Specific Knowledge in the Division of Labour Among Firms: A Study of the Software Industry", *Research Policy*, (30), 2001, pp. 1425-1442.

Heeks, R. "Software Strategies in Developing Countries", *Communication of the ACM* (42:6), 1999, pp. 16-20.

Heeks, R., and Nicholson, B. "Software Export Success Factors and Strategies in Developing and Transitional Economies", *Development Informatics Working Paper Series*, Working Paper No. 12, Institute for Development Policy and Management, 2002.

Heeks, R., Krishna, S., Nicholson, B. and Sahay, S. "'Synching' or 'Sinking': Trajectories and Strategies in Global Software Outsourcing Relationships", *IEEE Software*, (18.2), 2001, pp. 54-62.

Henderson, J., C., and Lee, S. "Managing I/S Design Teams: A control Theories Perspective", *Management Science*, (38:6), June 1992.

Herbsleb, J.D. and Moitra, D. "Global Software Development", *IEEE Software*, March/April 2001, pp 16-20.

Hofstede, G. *Cultures and Organisations*, Maidenhead: McGraw Hill, 1991.

Kim, C., Westin, S., and Dholakia, N. "Globalization of the Software Industry: Trends and Strategies", *Information & Management*, (17), November 1, 1989, pp. 197-206.

Klein, H. K. and Myers, M. D. "A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems", *MIS Quartely*, (23:1), 1999, pp. 67-93.

Meadows, C. J. "Globalizing Software Development", *Journal of Global Information Management*, (4:1), 1995.

Miles, M. and Huberman, A. *Qualitative Data Analysis*, California: Sage, 1994.

Mitra, S. and Narasimhan, S. "International Outsourcing of Software Development: A Framework for Issues", *Information and Management*, (31), December 1996, pp. 165-175.

Multimedia Development Corporation. Available at: http://www.mdc.com.my/

Orlikwoski, W. J. "CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development", *MIS Quarterly*, (17:3), September 1993.

Pettigrew, A. "Context and Action in the Transformation of the Firm", *Journal of Management Studies*, (24:6), 1987, pp. 649-670.

Pettigrew, A. "Longitudinal Field Research on Change: Theory and Practice", *Organization Science*, (1:3), 1990, pp. 267-292.

Phillips, D. E. "Selected Legal Issues in International Software Outsourcing, American Intellectual Property Law Association", *Spring Meeting*, April 19, 2002, New York.

Ramarapu, N. and Parzinger. M. J. "Issues in Foreign Outsourcing", *Information Systems Management*, (14:2), Spring 1997, pp. 27-31.

Rautiainen, K., Lasenius, C. and Sulonen, R. "4CC: A Framework for Managing Software Product Development", *Engineering Management Journal*, (14:2), June 2002.

Ravichandran, R., Ahmed, N. "Offshore Systems Development", *Information & Management*, (24), 1993, pp. 33-40.

Walsham, G. "Interpretive Case Studies in IS Research: Nature and Method", *European Journal of Information Systems*, (4:2), 1995, pp. 74-81.

Walsham, G. "Cross-Cultural Software Production and Use: A Structural Analysis", *MIS Quarterly,* (26:4), December 2002, pp. 359-380.

Walsham, G. and Waema, T. "Information Systems Strategy and Implementation: A Case Study of a Building Society", *ACM Transactions on Information Systems*, (12:2), 1994, pp. 150-173.

Whittaker, S. and Schwarz, H. "Meetings of the Board: The Impact of Scheduling Medium on Long Term Group Coordination in Software Development", *Computer Supported Cooperative Work*, (8), 1999, pp. 175-205.

Willcocks, L., and Fitzgerald, G. *Guide to Outsourcing Information Technology*, London: Business Intelligence, 1994.

Zhuge, H. "Knowledge Flow Management for Distributed Team Software Development", *Knowledge-Based System,* (15), 2002, pp. 465-471.

About the Authors

Aini Aman is a doctoral student in Information Systems at the School of Accounting and Finance, University of Manchester, England. Her research interest is in offshore software development outsourcing. For her PhD research, she is currently looking at the process of collaboration using groupware application in offshore software development outsourcing. She has a Master in Business Administration from Manchester Business School, University of Manchester. Her academic background is in Accounting Information Systems and Control and she is currently in a study leave as a university lecturer at Universiti Kebangsaan Malaysia, Malaysia. Aini can be reached by e-mail at ainiaman@hotmail.com.

Brian Nicholson is a lecturer in Information Systems at the School of Accounting and Finance, University of Manchester, England. His research into global software development and software export success for developing countries has involved several international collaborations involving fieldwork in India and recently Iran. Brian can be reached by e-mail at brian.nicholson@man.ac.uk.