DECIPHERING SOFTWARE DEVELOPMENT OUTSOURCING THROUGH TRANSACTION COST THEORY

Benoît A. AUBERT1
Suzanne RIVARD4
Michel PATRY5

1Laval University, Quebec City
2Ecole des Hautes Etudes Commerciales, Montréal

ABSTRACT

This paper reports the results of a study that was aimed at better understanding decisions regarding outsourcing of software development, as well as their consequences. In order to do so, a study was conducted in three large organizations, each reflecting an extreme situation: the first enterprise had outsourced all its software development activities; the second conducted all software development in-house; finally, the third had first chosen outsourcing, and had later reversed its decision. The data collected were analyzed using transaction cost theory as a conceptual framework, and each situation was examined with respect to the theory’s three main variables: asset specificity, uncertainty and measurement problems, and frequency of the transaction.

1. INTRODUCTION

Software development is a risky venture. Information systems (IS) projects have behind them an history of failures, delayed implementations and budget overruns [3, 10]. Nevertheless, IS can provide benefits in terms of efficiency and strategic advantages, and firms continue to invest heavily in it [8].

Recently, many companies have chosen outsourcing as an alternative for providing information services [9]. In doing so, they seek cost reductions, improvements in service quality, increased efficiency, and better access to scarce resources [2]. While outsourcing often brings advantages, it introduces a new kind of risk in the software development activity, namely, contractual risk [7]. This type of risk is due to the fact that contracting parties have to agree on specifications, schedules, and pricing arrangements, all being subject to uncertainty. Moreover, once agreed upon, these constraints have to be respected, and each party has to protect itself from opportunistic behaviour from the other party.

In order to obtain actual benefits from outsourcing, organizations conducting software development projects should outsource those activities that will be best performed by an outsourcing, while retaining in-house the activities for which they are better equipped to succeed, or for which outsourcing would create too many contractual problems.

Transaction cost theory is a useful conceptual framework for assisting in such decisions. This paper reports the results of a study that was aimed at better understanding decisions regarding outsourcing through the use of transaction cost theory. In order to do so, three organizations that reflected extreme situations were studied. Prior to presenting the organization settings and their analysis, a brief overview of transaction cost theory is provided.
2. TRANSACTION COST THEORY

Transaction cost theory views the market and the firm as alternative governance mechanisms, each being more appropriate to a given situation [5]. As shown in Table 1, the decision to use the market or the firm to regulate a transaction depends on three dimensions: (1) specificity of the assets required to produce the goods; (2) uncertainty and measurement problems surrounding the transaction; and (3) frequency of the transaction [11, 16].

Specificity characterizes an asset that cannot be redeployed without sacrificing its productive value if the contract is interrupted or prematurely terminated. Because the "next best use" value of the asset is much lower, the investor would lose part of its investment if the transaction was not completed. This creates a lock-in situation where the other party (not investing) could extract an advantage from the investor by threatening to withdraw from the transaction [14].

For a market to be efficient, parties must be able to predict with enough certainty the activities to be performed in a contract and to measure the value of the elements exchanged [12]. However, many activities bear a certain level of uncertainty and are subject to measurement problems [1]. For example, it may be difficult to predict future user needs in a given project. Evaluating the adequacy of a specific system delivered is also an arduous task.

Frequency is another key dimension of a transaction. Organizing a transaction within the firm implies creating a governance structure. This generates important and irreversible costs. If a transaction is known to be unique, these costs will likely be too important to allow for the integration of the activity within the firm [13]. The firm will prefer to bear the risks associated with investments or uncertainty rather than invest in order to internalize a single transaction [17].

<table>
<thead>
<tr>
<th>in-house</th>
<th>outsourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>highly specific investments required</td>
<td>no or generic investments required</td>
</tr>
<tr>
<td>high level of uncertainty or measurement problems</td>
<td>low level of uncertainty or measurement problems</td>
</tr>
<tr>
<td>activity performed on a continuous basis</td>
<td>activity is either rarely performed or subject to volume fluctuations</td>
</tr>
</tbody>
</table>

Table 1: Transaction Dimensions

3. METHODOLOGY

The objective of this study was to better understand outsourcing decisions using transaction cost theory as a conceptual basis. In order to do so, a qualitative approach was selected, since the level of detail and the insights provided by qualitative methods enable the researcher to identify possible rival hypotheses to explain a given phenomenon [4, 18].

Three sites, which reflected extreme situations, were selected: Publicor had outsourced all its software development activities; Distributor, all software development activities were conducted in-house; finally, Insuror had chosen outsourcing as governance mode and had later reversed its decision.

Each organization's leading IS manager acted as the respondents. Semi-structured interviews with these managers provided substantial information and gave an overview of the situation. The interview data were analyzed, and a series of questions were sent to each respondent to follow-up on issues raised during this analysis. Finally, phone interviews were conducted to gather any missing information.

4. PUBLICOR: TOTAL OUTSOURCING

At the time of the study, Publicor made intensive use of information technologies. In order to meet the technological challenges it was facing, the firm had outsourced all of its IT activities a few years before the study was conducted, and concluded a complex deal with three large outsourcing: Alpha, Beta, and Gamma.

Publicor was using a wide variety of ITs, and a wide variety of software running on various platforms. An IT organization of more than 1000 people was needed to fulfill the corporation’s needs. Each outsourcing employed approximately one-third of the workforce.

Publicor had decided to deal with three suppliers, instead of only one, for a straightforward reason: competition. In doing so, management ensured that there would always be a supplier able to take charge of the workload of another supplier on relatively short notice.

This type of arrangement placed the outsourcers in a situation they were not used to. They had to cooperate on almost every project. Since the architecture work was outsourced to Beta and the database and technology support was outsourced to Alpha, any new application development needed approval from both suppliers. Similarly, Gamma had the responsibility for the utility on which all the software was running. The suppliers had no choice but to cooperate with Gamma. An interesting result introduced by this type of arrangement was that Publicor was automatically getting a third party's view of each supplier's work.

If a supplier lost a portfolio, the costs of handing this portfolio to another supplier were hard to quantify. Since it would have been one of the three suppliers mentioned before, this learning curve would have been minimized, it would be limited to the time needed to understand the specific applications.

Elsewhere in the world, firms doing business in the same field as Publicor were just starting to realize the new technology challenges. This created an opportunity for Publicor and its suppliers. At the time the study was conducted, Publicor's technology was being implemented in eight other countries. This technology transfer was done in partnership with one of the three outsourcers.

These deals were extremely attractive to the outsourcers. Publicor received calls on a regular basis from other very large public sector organizations, considering the outsourcing route. These organizations would also contact Publicor's outsourcers in order to obtain information. The suppliers used Publicor as a reference point and a marketing tool.

Publicor had a history of measurement, in every department. All managers were steadily measuring all activities. This importance of measuring each activity was a fundamental element of the management of the corporation. Therefore, IT managers had been measuring their activities for six years before entering into total outsourcing. They possessed a lot of measurement experience and had internal estimates and evaluation indicators for each type of activity.

Prior to outsourcing, key indicators like cost per milestone, total development cost, elapsed time, and total cost minus fixed assets, were used for assessing software projects. These parameters were clearly specified ex ante so the suppliers knew on what basis they were being evaluated. Activities were measured on a regular basis; graphing the number of problems, the types of problems, their category according to the security level, and their overall impact. Publicor was constantly looking for productivity upgrades and it was constantly forecasting.

Publicor also had measurement guidelines for mainframe applications, distributed systems applications, and all other systems. These guidelines reflected the entire spectrum of systems. Some of these standards were established with projects that were conducted internally, others with projects that were competitively awarded. These estimation techniques were an incentive for the suppliers to remain fair.

1 At the request of the participating firms, their identity is to remain confidential.
4.1 Analysis

4.1.1 Asset Specificity

There were numerous specific investments in the deals linking Publicor with its suppliers. The suppliers invested heavily in acquiring knowledge about the firm's business and its situation. Therefore, there was a risk of quasi-rent extraction by the suppliers. Publicor would not have been able to change suppliers on short notice without having the new supplier go through the whole learning curve. Publicor established several protection mechanisms. The first and foremost was the presence of three suppliers having made specific investments. This meant that competition was always possible between the suppliers. They would have been on equal footing to compete since they all owned the same knowledge about the situation. In fact, the arrangement created a kind of quasi-market where three players of comparable strength were competing for the contracts. This significantly lowered the hold-up possibilities since two competitors would always wait to take over the work of the third one.

There was also a risk for the suppliers making these specific investments. They required long-term arrangements with Publicor in order to recoup their investment. They received these long-term guarantees and would deal with their client as long as they respected the norms, prices, and service levels agreed upon. The long-term guarantees may have been the most conventional means to encourage the suppliers to make specific investments, but they certainly were not the most intriguing features of the arrangement. The suppliers were given the potential to leverage these investments outside their contract with Publicor, by dealing with similar organizations in other countries. Investments lose their specificity when they are considered in an international setting. There are other markets, in other countries, for the services offered to Publicor. Their contracts provided the suppliers a privileged access to these markets by specifying that Publicor would pilot them in this closed environment. Suppliers needed to be formally introduced by a member of public sector group to which Publicor belonged, if they were to be acknowledged by the other members. This possibility, clearly stated before the arrangements were signed, was an additional incentive for the suppliers to invest in their relationship with Publicor. They were also assured that Publicor would put forth the necessary efforts to introduce them elsewhere in the world since it was also profitable for it when the solutions developed for its needs were sold elsewhere.

Expansion to outside markets emphasized the effect of reputation safeguarding protecting the client from sub-standard performance. Since suppliers heavily depended on Publicor to introduce them to other similar organizations, the cost of not providing adequate service to Publicor was much higher than the simple loss of the Publicor contract. It also meant that the international deals would be threatened.

4.1.2 Measurement Problems

Measurement was part of Publicor's management culture. The measures used for software development projects were very detailed and their experience with measurement enabled them to perform comparisons for each activity. They had been measuring these activities long before outsourcing. Therefore, they had a substantive basis of information against which to compare each activity. This alleviated measurement problems.

Another interesting feature of the arrangement is that the suppliers must receive each other's approval on several dimensions of the projects. This meant that, for any piece of work a supplier was providing, there were two independent and "non-collusive" critics to evaluate the work. Publicor was guaranteed of having two informed and critical opinions on any task conducted for them. In this sense, quality control was partially done by the other two suppliers not performing the work.

4.1.3 Frequency

The scale of software development activities at Publicor was considerable. The frequency of information systems development transactions was extremely high, counting the development hours by the million. This high frequency and volume of development activity justified the establishment of complex governance structures, sophisticated management mechanisms to govern these transactions. Publicor's practice was far from a simple market transaction. It was a long-term relationship between three suppliers and a client, with multifarious features enabling each party to gain from the association. Each supplier delegated more than three hundred employees to accommodate Publicor business. If the size of the contract had been smaller, or the frequency of software development activities had been lower, it would not have been as profitable for the suppliers to invest in such a relationship and it would have been more difficult for Publicor to convince them to make the required investments.

It was also this high frequency that enabled Publicor to simultaneously work with three suppliers, providing them with a sufficient workload to keep them motivated. It also increased their ability to measure their activities since their measurement system was based on comparisons with similar activities. The more activities there were, the easier and the more accurate the comparison became.

5. DISTRIBUTOR: IN-HOUSE

Distributor was a leading distributor and retailer in the food industry, employing approximately 14000 people, with 75 billion annual sales.

Distributor conducts most of its software development projects in-house. The company's application portfolio included four major systems. The first one, the retail system, processed the price information and performed several analyses. It ran on an IBM 3090, using DB2. The second was the distribution system, and consisted in three sub-systems: grocery, fruit and vegetables, and meat. It also ran on an IBM 3090. The third one was the finance system, which ran on an AS400. The last group of applications pertained to EDI. Expertise between the three main groups (retail, distribution, and finance) was not transferable because the systems were not operating in the same technological environment.

Use of consultants was limited. Consultants had worked in the past for Distributor on very well-defined projects. For example, modifications to the accounting programs required by the introduction of a new taxation scheme were made by consultants. The new taxation regime commanded modifications of the firm's software of approximately 5000 person-days, and were required by law. The supplier that made the modifications knew the financial system, and the changes required by the new tax were similar across companies since it was a tax applicable to all types of products. External resources were used because the internal workforce was tight and no employee was available for this particular task.

5.1 Analyse: "Analysis"

5.1.1 Asset Specificity

Many elements from this site indicate that significant asset specificity was present in the development of the information systems at Distributor. IS was viewed as a strategic weapon in the food industry. The systems were extremely close to the firm's activities and strategic decisions. Structural liaison mechanisms were an illustration of this close relationship between the IS function and the firm's strategy.
Asset specificity constraints mostly prevailed in terms of human resources specificity. Employees working on software development needed to gain significant knowledge of the food industry and of Distributor characteristics. Transaction theory predicts that the more specific the knowledge required, the more incentive the firm has to conduct these activities in-house. This was evident in the pattern of governance observed. The first phases of the software development process were primarily conducted in-house. These were the activities where project leaders and analysts were more involved. Project leaders and analysts presented the highest replacement delays and the highest required learning, when a new employee joined the firm. These long replacement delays and learning periods meant that the switching cost for changing a project leader or an analyst were more important than for other categories of employees like programmers. Therefore, the use of internal staff was more advantageous.

On the other hand, the last phases were the ones where more consultant help was used. The coding phase presented little specificity. The programmers performing this task needed little new knowledge, which was hired and could be relatively easily replaced. These lower switching costs facilitated the reliance on external suppliers for these tasks.

The choice of the systems on which consultants could be employed was also linked to asset specificity. Consultants were used mainly to work on financial systems. These systems were less specific than the ones linked to the food industry. Financial systems are used in several industries, as opposed to the other systems which were either unique to Distributor or used by only a few firms in the industry. The less specific knowledge required to work on the financial system facilitated the use of external suppliers. The choice of tasks given to the suppliers was also consistent with the asset specificity argument. Modifications imposed by the introduction of the new tax were standard across companies and industries. Consultants could perform the same modifications for several clients. These modifications were non-specific. It was therefore logical to acquire this unspecific expertise outside the firm since more than one supplier could provide it and since it had already been developed for other companies.

5.1.2 Measurement Problems
Management of software development projects at Distributor relied on a well-structured methodology, various deliverables enabling the management team to follow the progress of the project. There was always some uncertainty surrounding the projects. This was illustrated by the remarks made about "drawer systems", parallel procedures used to develop to handle special cases. These special cases often remained hidden during the requirement analysis and appeared only when the system was being tested. Of course, the modifications needed to accommodate these exceptions were much more costly than if the exceptions had been foreseen at the start. There was no malvolent intent from the users in these cases, usually it was a simple gap in the problem definition, attributable to a lack of attention from the project team and from the users who reviewed the specifications.

The presence of such discrepancies between actual user needs and system specifications prevented the firm from using comprehensive contracts to outsource software development. Turnkey-type arrangements do not permit software modification or adaptation without an adjusting the price. This adjustment can significantly reduce the benefits of the system. The flexibility Distributor maintained by being able to "pull the plug" on a project, whenever management felt that the system benefits would not repay the costs involved, would be lost if systems development was outsourced.

5.1.3 Frequency
In software development there were two main streams of work: maintenance and new system development. Distributor had a large portfolio of applications to maintain. There was also a significant demand for development activities. The firm had a strategic plan providing architecture forecast for new systems that would be needed in the future. Distributor representative mentioned that 85000 person-days of development demand were already planned. This guaranteed that an internal workforce would have sufficient work to remain occupied for a long period of time. Therefore, in this case, outsourcing could not be justified by a short term need of IS employees.

The allocation of the tax modifications to a supplier was another manifestation of the frequency constraint. These modifications were not anticipated. Companies were forced to comply with the law in a short time-frame. Thus, it made sense not to disrupt the on-going software development work flow and transfer this activity to suppliers.

6. INSUROR: AN OUTSOURCING DECISION REVERSED
Insuror was involved in both casualty insurance and reinsurance, and had assets of approximately 4505 million. The casualty insurance business was very standardized, operating in a regulated environment. Information systems handling the information related to these products were similar from one firm to another. There was no strategic element built in them to handle this type of insurance. Yet, systems required for casualty insurance were extremely complex, handling several transactions for the same policy, along with some back-dated transactions. There was a massive amount of information to process for each policy.

IS were at the core of any insurance company's activities. When a company or a portfolio of policies was bought, it had to be transferred onto the firm's current information systems. When the company offered a new product, it had to modify some features of the systems. After a major reorganization, Insuror started to search for an outsourcer. This resulted in the selection of Omega, a firm from outside North-America, already heavily involved in the insurance business. Omega had developed a package especially for the insurance industry, and was already active in several countries. For Omega, the contract with Insuror was an opportunity to enter the North-American market. Although Omega was not primarily an outsourcer but a software vendor, it managed data centers for some clients (outside North America) using its product. Omega offered to manage Insuror's data center and the offer was accepted. It was definitely a win-win situation. This represented an opportunity to buy a data center and acquire qualified staff along with an important client. For Insuror, it represented access to the software developed by the international division of Omega and the avoidance of updating their own information systems.

As time passed, Insuror began to experience problems with its IS operations. The volume of operations was increasing and batch window problems started to appear. There was not enough time at night to process the jobs and have the systems available in the morning. Some jobs had to be skipped during the batch run and processed during the week-end. This was a major problem. Insuror began to argue with Omega over these problems. The supplier did not want to invest more in software in hardware since they had no other clients. Omega began to debate over the definitions of the service level clauses, stating that their commitment was to have the system available 97% of the time and that if a batch run was skipped and the system was still on line at seven in the morning, the 97% target was met, simply because the system was on.

Insuror did not want to bring the case to court to try to get the service level it felt entitled to. Its representative indicated that a firm would require a very large contract to make it worth the legal expenses of finding experts to establish in court what constitutes good service. For example, even supposedly standard measures like response time could lead to costly arguing. If there is a maximum response time established in a contract, it must specify where it must be measured. It can be at the data center or at the client's site. Even if the parties agreed, network experts would be needed to establish if the slow response time was due to a problem with the CPU, the line speed, or the controller. There may be a problem in a local area network. The judge would have to be educat-
ed and experts would be needed to demonstrate what is the real source of the problem. That can be very difficult to demonstrate the source of these problems. In the case of Insuror, overnight batch problems were the result of hardware problems and some architectural flaws introduced when Omega redesigned the system. These resulted in excessive runs, therefore the batch processing took more CPU time than needed.

Omega soon experienced cash-flow problems which meant that they could not commit the required resources to pursue the development effort, and could not deliver the promised functionalities. The contract defined two groups of functionalities: the base system and the extra functionalities. Omega and Insuror began arguing about what constituted the base system and the extra functionalities, i.e., what was to be delivered for the fixed price, and what would be developed for additional compensation.

Going back to their business plan, Omega realized that their only division really involved in data center management was their international division. They were more involved in the software side of the industry. Omega realized that it would be more profitable for them to refocus on their main line of business. They amicably renegotiated to transfer the data center back to Insuror, which bought back all the equipment and all the personnel. During the following months, Insuror upgraded the CPU and some critical applications. Rapidly, without spending much money, the data center problems were solved. The batch window problems and the on-line response time problems belonged to the past.

6.1 Analysis

6.1.1 Asset Specificity

At first, the parties evaluated asset specificity as being low. The deal concluded with Omega was supposed to involve a simple adaptation of another system. There were many casualty insurance companies in the world, which may support the first opinion of the supplier regarding low asset specificity involved in this transaction. This opinion was proven false. After work began, the parties soon understood that it involved a lot more specific investments than what was expected. The particularities of the U.S. insurance system were very different from the other ones. The unique procedures of U.S. companies, and the additional constraints imposed by state regulations made the information systems needed both highly complex and specific.

This type of transaction called for important investments from the supplier and a long-term contract. Omega poorly evaluated these specific investments and therefore “backed out” of the transaction. It was not ready to commit additional resources without having guarantees from its client for more money or a longer term commitment. The client did not feel responsible for the supplier’s errors. Furthermore, the client had no guarantee that the supplier would put forth the necessary efforts to solve the problems after giving more resources.

The idea behind the evaluation of the arrangement was not to attribute fault to either party but to identify what went wrong in the relationship between the two organizations. It seemed that an underestimation - by both parties - of the specificity of the U.S. casualty insurance environment and its information systems needs was at the origin of most of the problems.

When the information about the extent of specific investments needed to complete the transaction was known to the parties, it was too late for them to adjust the rules governing their relationship. The contract was signed. If the information had been known before the contract was signed, the parties would have probably discarded the contract. The supplier would have required a very long term contract which the client could not have accepted. In this case, sequential spot contracting might have been a much better solution for both parties, enabling them to adjust the contract parameters as the information about the activities and the information systems requirements came available.

6.1.2 Measurement Problems

The contract was not accompanied by sufficient measures for quality control. The measures were not precise enough to be enforced. Consequently, they were useless. No arbitration mechanism was planned ex ante. A more detailed contract could have induced the supplier to comply with the contract in order to avoid a legal battle. Parties could not agree on what constituted the base functionalities versus the supplementary ones. Agreeing on what constituted an acceptable version of a function was even more difficult. There were too many moving targets to develop a workable contract.

The parties had no real basis to judge and agree on the quality of the software delivered and they had no option to go to a third party. Informing this third party would have been much too costly to make the dispute worthwhile for either party. No mechanism had been planned to ensure that the parties worked toward a common goal, which would have lowered the need for clear measures. Under these conditions, the transaction had little chance of succeeding.

Insuror had no control over what Omega was working on, nor over the method the supplier used to perform the tasks. They had to bring all software development activities back in-house to eliminate the time spent arguing over what should have been delivered and then working on the software needs of the company. Bringing the software development back in-house gave them the required flexibility to work on what was really needed without having to specify everything.

6.1.3 Frequency

The frequency of software development activities was high and seemed constant. The information gathered showed that the use of resources was fairly intense and justified the use of a stable workforce. It was possible to maintain in-house staff fully employed. Therefore, frequency was not at the origin of the decision to outsource the information system software development. After the operations returned in-house, the stream of development efforts linked to productivity improvement also eased the constraints related to frequency. As mentioned by the respondent, the personnel allotted to the development of these systems can be reallocated to work on urgent projects (like a modification required by a new law) if necessary. This pool of available resources, already knowledgeable of the firm environment, reduced the incentive to use external help.

7. CONCLUSION

While the results illustrate the explanatory power of the transaction cost model, it also shows that each variable of the model seem to have more explanatory power than the others in one specific situation. For instance, at Publicor, the measure of the activities and the presence of three suppliers to protect the client from lock-in problems resulting from specific investments, were the most relevant variables in understanding the situation. At Distributor, measurement problems were crucial to explaining the failure of the agreement the company had with its supplier. They simply could not force the supplier to provide quality service nor to commit to the specific investments required. Distributor’s situation was the best example, among the three, of the role of frequency in any make-or-buy decision.

Although three cases do not constitute proof of theory, no situations where the information collected contradicted the model was encountered. This enhanced the confidence in the theoretical approach. The next step would probably involve a larger scale approach for testing the model, assuming that comparable elements across several sites can be identified.
EXPLORING THE IMPACT OF CROSS-CULTURAL DIFFERENCES IN INTERNATIONAL SOFTWARE DEVELOPMENT TEAMS: INDIAN EXPATRIATES IN JAMAICA

Michael BARRETT
Dept. of AMIS
Faculty of Business
University of Alberta
Alberta, Edmonton
Canada T6G 2R6
Tel.: +1 403 492-4693
Fax: +1 403 492-3325
E-Mail: Michael.Barrett@ualberta.ca

Aldemir DRUMMOND
The Judge Institute of Management Studies
Trumpington Street
University of Cambridge
Cambridge, CB2 1AG
Tel.: +44 1223 339621
Fax: +44 1223 339701
E-Mail: ald1008@cus.cam.ac.uk

Sundeep SAHAY
ITI
University of Salford
Salford, M5 4WT
United Kingdom
Tel: +44 161 745-5482
Fax: +44 161 745-8169
E-Mail: s.sahay@iti.ac.uk

ABSTRACT

The growing trend of globalisation and IT has facilitated the mobility of IS labour. This has been further encouraged by the rapid growth of outsourcing of IS services in recent years. These outsourced services include software development which is increasingly been staffed with cross-cultural teams, and may often be led by an IS expatriate manager. The utilisation of skilled IS labour is particularly important in countries like Jamaica where these resources are scarce and in high demand. Our research explores a case study in Jamaica where Indian expatriate IS personnel led a software development project along with their Jamaican counterparts. Conflict developed within these teams as a result of cross-cultural differences. We suggest an approach based on Giddens’ structuration theory which links human action and social structure in developing a subtle understanding of this process. This approach may be particularly useful in developing relevant training programs for IS expatriate managers as a means of preventing as well as diagnosing cross-cultural issues.

1. INTRODUCTION

The objective of this paper is to explore how cross-cultural issues between members of international teams impact processes of information systems (IS) development. These issues are becoming of increasing relevance and important as software development is being carried out in different countries, often using expatriate IS managers. In recent years, this phenomenon has received widespread attention in the popular press as a number of companies seek to establish alliances with software houses globally as part of their outsourcing of IS development. For exam-