Game Theory Perspectives on Client – Vendor relationships in offshore software outsourcing

Nilay V. Oza
Helsinki University of Technology
Software Business Laboratory,
P.O. Box 5500, 02015 TKK, Finland
Tel: +358 50 534 3677
nilay.oza@tkk.fi

ABSTRACT

The objective of this paper is to provide the initial literature based insights into the game theory specifically with the viewpoint of client - vendor relationships in offshore software outsourcing. Game theory has been used for long in understanding various contexts in economics and other disciplines. Offshore software outsourcing relates to the situation in which client and vendor are operating from different countries. Subsequently, in this paper, the initial understanding of game theory focusing on software engineering community is developed. Particularly risk, rationality, payoffs, and other elements of game theory are explored in terms of how they affect offshore software outsourcing. The paper is structured as follows. Section one provides introduction to game theory concept. Section two explores the history, representation and types of games. Section three compares offshore software outsourcing with types and elements of game theory. Section four discusses one of the most famous game theory examples - 'prisoners-dilemma' and relates it to software outsourcing context. Finally, section five concludes this paper with the intended future work.

Categories and Subject Descriptors

D.2.9 Management - *Programming teams*

General Terms – Management, Economics, Human Factors, Theory

Keywords – Game theory, Software outsourcing, Software engineering, Economics.

1. INTRODUCTION

Game theory is an important tool for understanding the strategies of business world. Game theory is reported to focus on finding the right strategies and making the right decisions (Nalebuff and Brandenburger, 1996). According to encyclopedia of Wikipedia, Game theory is closely related to economics in that it seeks to find rational strategies in situations where the outcome depends not only on one's own strategy and 'market conditions', but upon the strategies chosen by other players with possibly different or overlapping goals. Mobious (2006) defines game theory as a formal way to analyze interaction among a group of rational agents who behave strategically. Rasmusen (2001) explains that game theory is concerned with the actions of decision makers who are conscious that their actions affect each other. Subsequently, it can be said that game theory is concerned with the actions of decision makers who are aware that their actions affect each other (Osborne et. al, 1994; Rasmusen, 2001). However, central to all definitions of game theory is strategy, payoffs and rational attitude. It is hypothesized that human beings are absolutely rational in their economic choices. In other words, each person maximizes his or her rewards – profits, incomes, or subjective benefits – in the circumstances that he or she faces. To understand this in terms of game theory, players try to maximize their rewards i.e. payoffs. To maximize the payoffs, players will develop plans known as strategies that pick actions depending on the information collected at each moment. In this context, the game theory uses metaphor of game to explore the interactions in human beings. Therefore as in games, the individual has a strategy to play in the game, and the outcome of actions depends on the strategies chosen by each individual in the game based on the information available to them. The essential elements of game theory are often classified as PAPI - Players, Actions, Payoffs and Information. It is to be noted that there is more than one decision maker in any game who is referred to as player.

Game theory has been used in many disciplines including economics, social science, ethics, computer science, and environmental studies. Rasmusen (2001) claims that the game theory has highly abstract representation of classes of real-life situations which allow them to be used to study a wide range of phenomena. However based on this paper, I intend to further develop comprehensive links between game theory and software organizations. It is imperative to focus on non-technical aspects of software engineering organizations, which in turn affects the business of the company.

2. OVERVIEW OF GAME THEORY

It is often reported that central to the management philosophy of many successful companies is the belief that a good manager could turn the task of business efficiency into a game every employee could play. However, in practice in many managers and lower-level employees are reported to uninformed about how the company generates cash and makes a profit. Subsequently, Burton and Terborg and Burton (2002) notes that by turning business into a game, one can tap into *the universal desire to win*. And in order to win, everyone needs to understand the rules, master the

EDSER'06, May 27, 2006, Shanghai, China.

fundamentals, know the score, and play together as a team. However, such knowledge is yet not embedded in the current practices of software organizations' working framework. Subsequently, this indicates great research opportunities in terms of extending the current body of knowledge in software engineering by game theory principles that has been successfully applied in a variety of disciplines. Next, I provide a brief history of game theory development.

2.1 A brief history

Walker (2005) gives the chronological development of game theory invention which traces history of game theory rooted in compilation of ancient law and tradition in first five centuries A.D. However, key period for the emergence of game theory was the decade of the 1940's. It is widely accepted that the publication of *The Theory of Games and Economic Behavior* by John von Neumann and Oskar Morgenstern and Tucker's invention of the Prisoners' Dilemma were some of the most influential in game theory development in latter half of the twentieth century. Game theory has recently come into attention to computer science research, particularly in the area of artificial intelligence and cybernetics.

2.2 Representation of games

Game theory is presented in normal or extensive form. In normal form, it is each player acts simultaneously or, at least, without knowing the actions of the other. Normal form of presentation is a tabular representation of the game (usually for two player games), where payoffs all players corresponding to players' strategy are presented in the table. Table 1 shows an example.

Table 1: Normal form of representation

	Action 1 - Player 2 chooses left	Action 2 - Player 2 chooses right
Action 1 - Player 1 chooses top	5,4	-2,-2
Action 2 - Player 1 chooses bottom	0,0	4,5

Here, there are two players; one chooses the row and the other chooses the column. Each player has two strategies, which are specified by the number of rows and the number of columns. The numbers presented in the table are payoffs for the players. For example, in table 1, the first number is the payoff received by the row player (Player 1); the second is the payoff for the column player (Player 2). Suppose that Player 1 plays top and that Player 2 plays left. Then Player 1 gets 5, and Player 2 gets 4.

If players have some information about the choices of other players, the game is usually presented in extensive form. Extensive form games are presented in trees structure. Here each node represents a point of choice for a player. The player is specified by a number listed by the node. The lines out of the node represent a possible action for that player. The payoffs are specified at the bottom of the tree. The example is shown in the following figure.

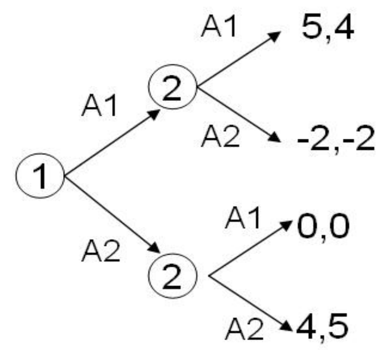

Figure 1: Extensive form of representation

Figure 1 shows that Player 1 moves first and chooses either A1 (action 1) or A2 (action 2). Player 2 sees Player 1's move and then correspondingly chooses A1 or A2. For example, Player 1 chooses A1, then Player 2 chooses A1, then Player 1 gets 5 and Player 2 gets 4.

2.3 Types of games

There are mainly four types of games – symmetric and asymmetric, zero sum and non-zero sum, simultaneous and sequential, and perfect information and imperfect information. In symmetric games, payoffs are dependent only on strategies employed and not on who is playing the game. In other words, identities of the players can be changed without changing the payoff to the strategies. Contrastingly in asymmetric games strategies identified for both players are different. In zero sum game, a player benefits only at the expense of others. For example chess or poker is zero sum games. However, in non-zero sum game, benefit of one player does not necessarily correspond with the expense of the other. In simultaneous games both players move simultaneously, or in other words, the later players are not aware of the actions taken by earlier players. Contrastingly in sequential games, the later players have some knowledge (may not be perfect knowledge) of the actions taken by earlier players. Subsequently, if all players know the moves previously made by all other players, the game becomes perfect information games. One example of perfect information game is chess whereas prisoners' dilemma can be considered as imperfect information game. It is to be noted that perfect information may not be the complete information. Perfect information comprises all the actions (or moves) of the other players whereas complete information includes strategies and payoffs and not actions.

3. OFFSHORE OUTSOURCING FROM GAME THEORY PERSPECTIVE

3.1 Types of games

Offshore software outsourcing is often referred to the context where outsourcing company i.e. client and vendor are operating from different countries. In other words, client outsources software development to the vendor located in other country. Success of the client - vendor relationship is mainly dependent on the management of relationships between them. Here I focus on some of the types and critical elements of game theory in terms of the offshore software outsourcing perspective.

Offshore outsourcing scenario matches with types of games identified in game theory. However, offshore outsourcing, if mapped to the metaphor of game, is difficult to classify as one type of game. It can be classified across nonzero-sum games, cooperative games or complete information games. Offshore outsourcing is a nonzero-sum game if any gain achieved by client or vendor does not correspond with a loss of the other. For example, if client gains higher productivity from vendor's technical competence, it does not create loss for vendor in terms of losing technical competence. Offshore outsourcing is also asymmetric in terms of different strategies identified for both clients and vendors. Furthermore, it can be complete information game if both clients and vendors have all the strategic and payoff related information about the outsourced project. However, it may become perfect information game if both clients and vendors have full knowledge about the actions taken by each other. In practice, this is rare though. The theory of incomplete contract and hidden expectations of clients and vendors dominate their actions and strategies which can make offshore outsourcing both incomplete and imperfect information game. Nevertheless, in the long term outsourcing contract, scenario of complete or perfect information game may become possible. Offshore outsourcing is unlikely to turn into zero-sum game where gain for one party is a loss for another party. This may be the case when offshore outsourcing relationship fails and gain achieved by one party can prove as loss for the other. If client or vendor becomes opportunistic in terms of damaging the 'common' interests aligned to outsourcing contract, it can turn into zero sum game.

In many cases, offshore outsourcing becomes cooperative where both clients and vendors form a contractual relationship between them. Here cooperation is not referred to players' cooperation to each other but cooperation is enforceable by an outside party (e.g. a court). In other words, players i.e. client and vendor are able to make enforceable contracts. However, any party may influence the negotiations based on the context. For example, vendor may influence bargain based on its monopoly in providing a particular software competence.

3.2 Elements of games

In this section, I explore some of the critical elements of game theory from the offshore software outsourcing perspective. This exploration may help to strengthen the mapping of game theory knowledge in software engineering.

3.2.1 Power

One of the elements of game is domination i.e. power of one player over other. In offshore software outsourcing, client may have more power over vendor at the initial stage of the relationship. However, power may shift to vendor after long term outsourcing work to a particular vendor when the renegotiation is required. In particular, if it is complete information case, vendor will be powerful enough to dominate the renegotiation cycle. However, client may distribute outsourcing work to more than one vendor so that the game does not turn into complete or perfect information game. Game theory considers that the key to understand who has power in any game is the concept of 'added value'.

3.2.2 Added value

Added value measures what each player brings to the game. The formal definition is: *Take the size of the pie when you and everyone else are in the game; then see how big a pie the other players can create without you* (Nalebuff and Brandenburger, 1996). The difference is the added value. To make it simpler, one's added value is the size of the pie when s/he is in the game minus the size of the pie when s/he is out of the game. In offshore software outsourcing, this may differ according to the type of the contractual relationship. Generally, client retains the control over the value added by the vendor. This can lead to speculation whether added value by vendor confirms more power for the vendor. Added value may strengthen vendor's position in renegotiation of the outsourced work. This seems in line with game theory principle of added value.

3.2.3 Perceptions (Ross, 2002)

Different people view the same thing (or concept) differently. The way people perceive the game influences the moves they make. Nalebuff et al., (1996) note that any description of a game must include how people perceive the game – even how they believe other people perceive it, how they believe other people believe the game is perceived, and so on. In offshore software outsourcing, client and vendor sign a contract that specifies all the possible rules to be followed during the relationship including what to do if one of the partners wants to end the relationship. If we consider that, both partners have agreed that the unsatisfied partner will state a price. The other partner must then either buy the first one out at that price or sell his partnership interest at that price. In most general cases, it will be perceived best to state a price at which you are equally happy being bought out or buying the other partner out. Based on game theory principle, it is reported that in such cases, if the venture is valued at $100 million, then one should state a price of $50 million. You do not know how the other person will reciprocate, but this way you have guaranteed yourself half the pie! However, the other partner might have valued the venture at $60 million. If it is the case, he would rather sell to you at $50 million, than pay that amount to buy something worth only $60 million. This suggests that it is worth thinking about the other partner's perception of the pie in addition to own pie. Subsequently, the right strategy takes account of your perception of the other partner's perception of the pie.

3.2.4 Rationality

Researchers often consider that game theory requires all the players to be rational. It is also assumed that everyone intends to maximize profits. In this context, a person is rational if he does the best he can, given how he perceives the game and how he evaluates the various possible outcomes of the game (Ross, 2002). Two people can both be rational and yet perceive the game quite differently. One person may have better information than the other. However, people can guess wrong and still be rational (Rasmusen, 2001). This is because they do the best they can, given what they know. Likewise, two people can both be rational and yet evaluate the same outcome quite differently. In offshore software outsourcing, client and vendors are rational partners. They direct their strategies based on how much benefit they can avail from each action they take. Their rationality perspectives may not consider harming each other's interests. This may be because client and vendor have their own businesses and different business objectives. Therefore, although working on a common project i.e. a game they may also have different goals to achieve out of it. However, one player's opportunism may damage interest of the other player. For example, two telecom outsourcing companies are outsourcing telecom software development contract to the same vendor. Here, vendor, by revealing strategy knowledge of one client to the other, who are in competition with each other, can become opportunistic. However, it is also likely that vendor's payoff turns out higher if he looses both clients! The rationality perspective is also important in terms of deciding and sharing risk with the other players in the game.

3.2.5 Risk

Elitzur and Wensley (1997) identify risk in terms of finance, business and technical aspects in software development outsourcing. They note that risk between the outsourcing company and the outsourcing vendor has a direct impact of the payoffs available to each of them. They also note that in many cases where outsourcing vendor company is bigger than outsourcing client company, it is possible that the vendor is ready to share certain risk that client may not be able to take. In terms of finance, outsourcing is often coined as cost-saving strategy for outsourcing company. However, it is imperative to note that all financial uncertainties are difficult to predict and resolve beforehand. Particularly, in offshore outsourcing, where coordination costs might be much higher than domestic outsourcing, financial risks may also increase. An outsourcing company should carefully calculate coordination costs in addition to development costs when considering financial risks.

Most outsourcing companies eventually sell their developed products (by offshore vendors) to their clients or consumers in the market. It may turn out potential risk if the outsourcing fails as it will potentially impact the business volume of client. Therefore the competence of the outsourcing vendor in terms of creating business value for the software is important in terms of business risk calculation. Elitzur and Wensley (1997) recommend that outsourcing vendor should be exposed to some of the business risk. If vendor has such expertise in the client's line of interest that he can develop better software than the outsourcing company itself, then, the business risk is less for the client. However, the payoffs for the vendor will increase because much of the risk will be shared by the vendor.

Technical risk relates to the functionality of the software being developed. At the outset, outsourcing company may decide to outsource to a specific vendor to averse the technical risk. This often happens if vendor is in better position to retain and increase the skilled manpower required for that particular software development. Vendor also achieves economies of scale and subsequently the technical skills for developing software in line of client's interests. Economies of scale allows vendor to bring in more and matured skills to the client's interest of software development. However, technical risk will be higher for outsourcing company if the vendor is very new in the client's software outsourcing interest.

All the above risks may become more severe in the 'offshore' context. The risks increase because the 'uncertainty' increases. Particularly in offshore outsourcing geographic distance crease many other challenges including lack of access, distance, loss of control, and less common patterns of working. Furthermore, uncertainty increases as client can not physically check the progress of the outsourced work. However, communication tools and project office tools are nowadays used that allow clients to see real time updates of their outsourced work. The increased uncertainty also impacts the development of trust in the vendor. Subsequently, the clients are likely to identify higher magnitude of risks in outsourcing their software offshore. However, the higher risk sharing from the vendor in the offshore context may also provide better payoffs!

4. PRISONER'S DILEMMA

The Prisoner's Dilemma (PD) is one of the earliest (in 1950) "games" developed in game theory. It highlights an excellent way of studying the issues of conflict vs. cooperation between players. Blumen (1995) claims business life is predominant with PD including the employer-employee relationship. Similarly, a company engages in a PD with its vendors and its clients. If we look at software development, every software development relationship is a PD, and most (if they involve multiple phases) are iterated PDs. The description of PD game is available in almost all game theory papers. However, it is worthwhile to explain it here to extend the understanding of it in software engineering community.

Based on Tucker who developed this classic game, a prisoners' dilemma can be described as follows. Two suspects, X and Y are arrested by the police for the Z crime. The police separated both prisoners and visited each of them and offered the deal: Each has to choose whether or not to confess and accuse the other. If neither X nor Y confesses, both will serve one year on a charge of conducting Z crime. If each confesses and accuse the other, both will go to prison for 10 years. However, if one confesses and accuses the other, and the other does not confess, the one who has collaborated with the police will go free, while the other will go to prison for 20 years on the maximum charge. Table 2 presents this scenario.

The strategies in this case are: not to confess or accuses other with the evidence (to go free). The payoffs are the sentences served.

Table 2: An example of prisoner's dilemma

	Y confess	Y doesn't confess
X confess	10,10	0,20
X doesn't confess	20,0	1,1

The table is read like this: Each prisoner chooses one of the two strategies. The two numbers in each cell tell the number of years of prison for the two suspects. The number to the left of the comma tells the payoff to the person who chooses the rows (X) while the number to the right of the column tells the payoff to the person who chooses the columns (Y). For example, if both X and Y confess, each gets 10 years of prison, but if Y confesses (and accuses X) and X does not confess, then, X gets 20 years of prison and Y goes free. If both does not confess and accuses each other, both will have 1 year of prison (1, 1). However, the rational attitude helps to address this game. For example Y may think that two things can happen, either X can confess or X does not confess. If X confesses, and I (Y) do not confess that I get 20 years. But if I confess (given X has confessed), I will get 10 years. Therefore it is best to confess. However, if X confesses, and I don't confess as well, then, I get one year. But if I confess (and x doesn't confess) I can go free. Either way, it is best if I confess. Subsequently X can also reason the same way as Y does and it is likely that they both will confess and get prison 10 years each. However, if they would had acted "irrationally," and would not have confessed, they each could have gotten only one year of prison.

The above game the dominant strategy is 'to confess' as it has given the best payoffs given that each player doesn't know about the possible action of the other. This is also referred to as dominant strategy equilibrium. It is to be noted that if both prisoners would have given chance to communicate, the expected outcome would be quite different.

4.1 Software outsourcing perspective

Here I quote an example from *The Ethical Spectacle*, which claims every software development relationship as a prisoner's dilemma. It gives example of software vendor exchanging the code for money with the client. For example, if vendor delivers something that does not work or not aligned with client's expectations, and the client pays for it, vendor has received higher payoff of the transaction and the client has gotten the cheater's payoff. If vendor delivers functional software, and the client keeps it but fails to pay, then vendor have received the cheater's payoff. If both vendor delivers working software and the client pays for it both gets the expected payoffs. If vendor does not deliver working software and client does not pay for it then both lose out but the client has kept its money and vendor retained the code. However, in practice, the client may examine the acceptability of the code in each round of payment. The satisfaction of client from the vendor's code will determine the next move. This makes it iterated PD where each round will determine separate payoffs. Here it should be noted that the perceptions of client are important in terms of judging the acceptability of the code provided by the vendor. If the judgment is wrong, the vendor may get higher payoffs and the client may get cheater's payoffs.

However, the reverse situation is also possible where client deliberately reject the code to move out of the game for his own opportunistic reasons. Therefore, well-drafted and fair contract can help vendor if he determines clients' requirements effectively and can agree them in the contract. This will limit the client's opportunism to reject by withholding acceptance (Humphrey, 1989). Here outsourcing game becomes cooperative as tightly defined contract will enforce the cooperative behavior. Without such enforceable contract, it is possible that a client withhold payment on the last round, to force the inclusion of additional requirements not previously thought of. However even with a contract, this may happen; since the iterated prisoner's dilemma is for a known number of rounds, the client, not expecting to deal with you again, may reject on the last round, simply hoping to keep the balance of its money (Blumen, 1995). But a well-drafted contract will at least limit its ability to get away with such a rejection. However, if client's business risks are higher than the payoffs achieved by not paying the vendor for the added business value, the client is not likely to reject in the last round. On the contrary if the added value from the vendor is higher, he may dominate the round when it comes to renegotiation.

5. SUMMARY AND FUTURE WORK

This paper only provided provisional insights into game theory and exemplifies it with the offshore software outsourcing context. It is identified that game theory can be helpful in evaluating many of the software business related aspects such as value perspectives, risk sharing and contractual matters. Particularly prisoner's dilemma is useful in understanding what different kinds of actions can be taken by each player in software outsourcing. However, this is just one example of understanding the game theory. Based on this initial understanding I intend to enhance my investigations into game theory and propose a game theory based model for software business. Based on this provisional literature survey on game theory, I derive the following critical issues worth exploring for further investigation.

- How game theory can be helpful in managing software development collaborations?
- Can a presence of trust between parties change the results of prisoner's dilemma?
- How risk can be managed based on game theoretic principles in software business?
- Which types of games can develop in software business?
- How game theory can be used in evaluating competing suppliers for software contracts?
- How game theory can be used in developing better contracts in software business?

The central to game theory is the strategies, payoffs and rational attitude of the players. Software engineering is also a human oriented endeavor and therefore all three central aspects of game theory are relevant to it. However, it is worthwhile to investigate how they can help in conducting better software business.

6. REFERENCES

[1] Blumen, J. (September 1995). The prisoner's dilemma in software development. *The Ethical Spectacle*. 1(9)

[2] Elitzur, R. and Wensley, A. (1997). Game theory as a tool for understanding information services outsourcing. *Journal of Information Technology.* 12. 45 -60.

[3] Humphrey, W. (1989). *Managing the software process.* Addison-Weseley Publishing Company, Inc.

[4] Mobius, M.M. (February, 2006). Lecture I -II: Motivations and decision theory. Lecture notes on Introduction to Game theory course, Harvard University. [http://www.courses.fas.harvard.edu/~ec1052/Lectures/lecture1.pdf]

[5] Nalebuff, B., and Brandenburger, A. (1996). *Co-opetition*. HarperCollins Business:London

[6] Osborne, M., and Rubinstein, A. (1994). *A course in game theory*. The MIT press: Cambridge

[7] Rasmusen, E. (May 2001). *Games and Information - An introduction to Game theory*. Blackwell:Cambridge 3rd (ed).

[8] Ross, D. (Spring 2002). Game Theory. *The Stanford Encyclopedia of Philosophy*. Edward N. Zalta (ed.) [http://plato.stanford.edu/archives/spr2002/entries/game-theory/].

[9] Terborg, J. and Burton, R. (Sep/Oct, 2002). , Making Business a Game We play sports to win. Isn't it the same in business? *Marketing Management.* 11(5).

[10] Von Neumann, J., and Morgensten, O. (1964). *Theory of games and economic behavior*. New York: Science Editions J. Wiley

[11] Waddington, C H. (1973). World War II: Operational Research against the U-Boat. *London: Elek Science*.

[12] Walker, P. (October, 2005). A chronology of game theory. [http://www.econ.canterbury.ac.nz/personal_pages/paul_walker/gt/hist.htm]