

Master Thesis
Software Engineering
Thesis no: MSE-2002-23
August 2002



Game Theory and Bidding for Software Projects

An Evaluation of the Bidding Behaviour of Software Engineers

Jacco Buisman

Department of
Software Engineering and Computer Science
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

This thesis is submitted to the Department of Software Engineering and Computer Science at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 10 weeks of full time studies.

Contact Information:

Author:

Jacco Buisman

Bonairestraat 32

9715 SE Groningen

The Netherlands

jacobuisman@hotmail.com

<http://jbuisman.web1000.com>

University advisor:

Claes Wohlin

Department of Software Engineering and Computer Science

Department of
Software Engineering and Computer Science
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

Internet : www.ipd.bth.se
Phone : +46 457 38 50 00
Fax : +46 457 271 25

Abstract

The conception phase is one of the most important phases of software projects. In this phase it is determined which software development company will perform a software project. To obtain a software project, companies can have several bidding strategies. This thesis investigates if and how game theory can be a helpful tool to evaluate bidding for software projects.

This thesis can be divided into two different parts: a theoretical and a practical. The theoretical part investigates the applicable parts of game theory in this thesis, explains what software projects are, explains the difference between costing and bidding and provides results of a literature survey about bidding behaviour. The practical part introduces a study to investigate strategies and bidding behaviour of software engineers, explains the experimental design that found the study, provides the results of the performed study and a discussion of the results.

This thesis concludes that game theory contains some concepts that make it possible to evaluate bidding for software projects.

Keywords: Game theory, bidding behaviour, study, software project, strategy

Acknowledgements

First of all, I would like to thank my advisor Claes Wohlin, for offering me this opportunity. His supervision and guidance were a major value to keep me on track and motivated continuously. Furthermore I would like to thank the country of Sweden for giving me the opportunity to study and work abroad. Thanks to all the teachers and colleagues of the IPD department at Blekinge Institute of Technology for the great and valuable experience. In particular I would like to thank Bengt Carlsson, Kennet Henningsson, Jelte Jansons, and Stefan Johansson for their offered time and knowledge to participate in the study. In addition, I would like to thank my friend/class mate/colleague Nina Dzamashvili for all the valuable discussions we have had. Thanks to these discussions I was able to solve unclear statements in this thesis. I would also like to thank my friend Hanna Sjöberg for reviewing my thesis before final submission. Finally, I thank my opponent Ulf Paulsson for his valuable comments I received from him during the presentation of my thesis.

Above all, I would like to thank my father Bert, my mother Ankie, and my two brothers Reinier and Robert for their patience, support and motivation during my stay in Sweden. I also would like to thank all the members of the Dutch colony in Ronneby and in particular my friends Jelte Jansons and Fernand Le Roux for their companion.

Contents

1	INTRODUCTION.....	1
2	GAME THEORY.....	2
2.1	INTRODUCTION.....	2
2.2	GAME CONCEPTS.....	2
2.3	INFORMATION IN GAMES.....	3
2.4	PLAYER(S).....	4
2.5	APPLICATION OF GAME THEORY FOR BIDDING FOR SOFTWARE PROJECTS.....	5
3	EXTENSIVE GAMES.....	6
3.1	PERFECT INFORMATION.....	6
3.1.1	<i>Repeated Games</i>	6
3.1.2	<i>Infinite vs. Finite</i>	7
3.2	IMPERFECT INFORMATION.....	7
3.2.1	<i>Concept</i>	7
3.2.2	<i>Perfect and Imperfect Recall</i>	8
3.3	BENEFITS AND DRAWBACKS.....	8
3.3.1	<i>Perfect Information</i>	8
3.3.2	<i>Imperfect Information</i>	8
3.3.3	<i>Infinite and Finite Games</i>	9
3.3.4	<i>Conclusions</i>	9
4	SOFTWARE PROJECTS.....	10
4.1	WHAT IS A PROJECT?.....	10
4.2	WHAT IS A SOFTWARE PROJECT?.....	10
4.3	ESTABLISHMENT OF SOFTWARE PROJECTS.....	12
4.3.1	<i>Conception</i>	12
4.3.2	<i>Request for Proposal</i>	13
4.3.3	<i>Project Proposal</i>	13
4.3.4	<i>Software Contract</i>	14
5	COSTING AND BIDDING.....	15
5.1	INTRODUCTION.....	15
5.2	SOFTWARE PROJECTS.....	15
5.2.1	<i>Costing</i>	15
5.2.2	<i>Bidding</i>	16
5.3	FOCUS OF THESIS.....	16
6	BIDDING BEHAVIOUR.....	17
6.1	INTRODUCTION.....	17
6.2	GAME THEORY.....	17
6.3	MOTIVATION.....	18
7	STUDY.....	20
7.1	GAME THEORY.....	20
7.2	EXPERIMENTAL DESIGN.....	20
7.3	RULES.....	21
7.4	PLAYERS.....	22
7.5	GAME PERFORMANCE.....	23
8	RESULTS.....	24
9	CONCLUSIONS.....	27
10	DISCUSSION.....	28
11	FUTURE WORK.....	29
	REFERENCES.....	30

APPENDIXES

A - EXPERIMENTAL DESIGN.....

B - PLAYER’S MOTIVATIONS

1 Introduction

Game theory is a mathematical tool for analysing decisions, which has been used for several decades now. It is a widely used tool for analysing decisions in game situations. The process of bidding for projects where the customer determines the desired company can be seen as a game as well. This is a rather new approach of using game theory, especially in the software science area. The reason that it is rather new for the software science area is that software engineering is rather new itself, compared to other branches of industries.

The purpose of this thesis is to study the possibility to apply game theory to evaluate the bidding for software projects. Because software projects become a more and more common way in software engineering to solve a certain task, it would be useful to understand bidding for software projects. To understand this bidding behaviour, game theory can be a useful tool. Starting from this assumption, a hypothesis has been formulated in the following way: Game theory can be applied to evaluate bidding for software projects.

To investigate this hypothesis the thesis consists of a literature survey and a study to perform the evaluation.

The thesis gives an answer to the following research questions:

- How is the bidding behaviour of software engineers?
- Does bidding for software projects differ from bidding for other projects?
- What is then typical for bidding for software projects?

The scope of this thesis is to do research in the software engineering area. The results can be useful for different sub-areas in and perspectives of software engineering. Examples are customers/clients, developers, and project managers. The results can even be useful for bidding for projects in general, but that is not the purpose of this thesis.

The thesis is organized as follows. The next chapter provides a general description of game theory. The chapter concludes with an indication and motivation of the applicable parts of the game theory principles to evaluate bidding for software projects. Chapter 3 provides some more details about these applicable parts and ends with a number of considerations (i.e. benefits and drawbacks). Chapter 4 provides information about software projects, like what they are and how they are established. The most essential activities during the establishment of a software project (in order to get it) are costing and bidding. These activities are explained in chapter 5. Chapter 6 presents the results of the performed literature survey with respect to existing studies on bidding behaviour. A study is conducted to evaluate bidding behaviour when bidding for a software project. The performed study is described in chapter 7 and the results of this study are presented in chapter 8. Finally, chapter 9 offers concluding comments, chapter 10 a discussion of the thesis and chapter 11 a notification of possible future work.

2 Game Theory

This chapter gives a general description of game theory. It provides information about the game concepts, different kind of information in game theory, and type of players that the game theory distinguishes. The chapter concludes with an indication and motivation of the applicable parts to evaluate bidding for software projects.

2.1 Introduction

Game theory is a collection of analytical tools that are designed to help to understand the phenomena that are observed when decision-makers interact. This sentence can be broken down in:

- *Game*: A game is a description of strategic interaction that includes the constraints on the actions that the players can take and the players' interests, but do not specify the actions that the players do take [2].
- *Analytical tools*: models that are highly abstract representations of classes of real-life situations. Because of their abstractness they can be used to study a wide range of phenomena.
- *Decision-makers*: The players who make the decisions. By using the analytical tools their behaviour can be observed.

2.2 Game Concepts

Game theory distinguishes two different concepts of games based on the moments when actions of the players are performed.

In *strategic games*, each player chooses his plan once and for all, and all players' decisions are made simultaneously and independently. This concept specifies a set of possible actions for each player and a preference ordering over the set of possible action profiles. The player can only form his expectation of the other players' behaviour on the basis of information about the way that the game or a similar game was played in the past. One of the most well-studied and interesting games in game theory is the Prisoner's Dilemma. Table 2-1 illustrates this example based on the example from Osborn and Rubinstein [2]. The figures in this table represent the number of years the suspects can be sentenced. For example 4, 0 means that suspect 1 will be sentenced for 4 years of prison and suspect 2 for 0 years.

		Suspect 2:	
		Not Confess	Confess
Suspect 1:	Not Confess	1, 1	0, 4
	Confess	4, 0	3, 3

Table 2-1The Prisoner's Dilemma

The two suspects of the crime are put into separate cells. If they both confess, each will be sentenced to three years in prison. If only one of them confesses, one will be set free and used as a witness against the other, who will receive a sentence of four years. If neither confesses, they will both be convicted of a minor offence and spend one year in prison.

The situation where both suspects confess and will be sentenced to three years in prison is called the nash equilibrium because neither suspect 1 nor 2 can benefit by changing strategy while the other does not change strategy. Osborn and Rubinstein [2] formulate the nash equilibrium as: “A steady state of the play of a strategic game in which each player holds the correct expectation about the other player’s behaviour and acts rationally.”

By contrast, the concept of an *extensive game* specifies the possible orders of events. Each player can consider his plan of action not only at the beginning of the game but also whenever he has to make a decision. Consider a game with two players, where player 1 is considering entering a market and player 2 is already on the market. Figure 2-1 shows a game tree of this game. This example is based on an example of Nachbar [3].

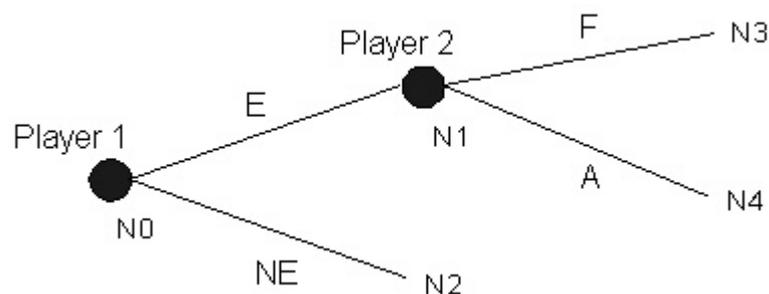


Figure 2-1 Extensive Game of considering entering a market

Player 1 has two possible actions at the initial decision node labelled N0: Enter (E) the market or Not Enter (NE) the market. If player 1 chooses to enter the market, the play moves to the decision node labelled N1 and Player 2 has to make a decision. If player 2 chooses F (Fight; launch a price war) then the play ends at the terminal node labelled N3. If player 2 chooses A (Accommodate), the play ends at node N4. Finally, if player 1 chooses NE, then the play ends at terminal node labelled N2. Each terminal node in a play corresponds to a possible outcome, which corresponds to an assignment of a set of payoffs. A payoff is an ordinal number assigned to a player. For example outcome (1, 2): 1 is the payoff for player 1, and 2 is the payoff for player 2.

In extensive games it is not possible to have a nash equilibrium because a player can change his plan of action. But in extensive games it also possible to achieve a ‘steady state’. This is called subgame perfect equilibrium [2] wherein the strategy is optimal after every history. To be able to keep track of this optimal history a strategy profile is used.

This is just a simple example to illustrate the basic ideas behind extensive games: more than one moment to consider the plan of action. But it is common that an extensive game has more than two decision nodes. In such cases the games increases rapidly: each decision node has two or more decision outcomes that have there decision nodes, and so on. This is why the games are called extensive games. A good example of a repeated extensive game with perfect information is Chess. Chapter 3 provides more information about repeated extensive games.

2.3 Information in Games

The previous example was an example of an extensive game with perfect information. This means that each player, when making any decision, is perfectly informed of all the events that have previously occurred. But there exist also games with imperfect information. In this kind

of games the player has only partial information about the actions taken previously by another player. The players will basically rely on expectations. Actually, this also implies that sequential games are games with imperfect information because the other player does not have any information about the previous events.

Consider the next example, also of Nachbar [3], where again two players perform a game. This is illustrated in Figure 2-2.

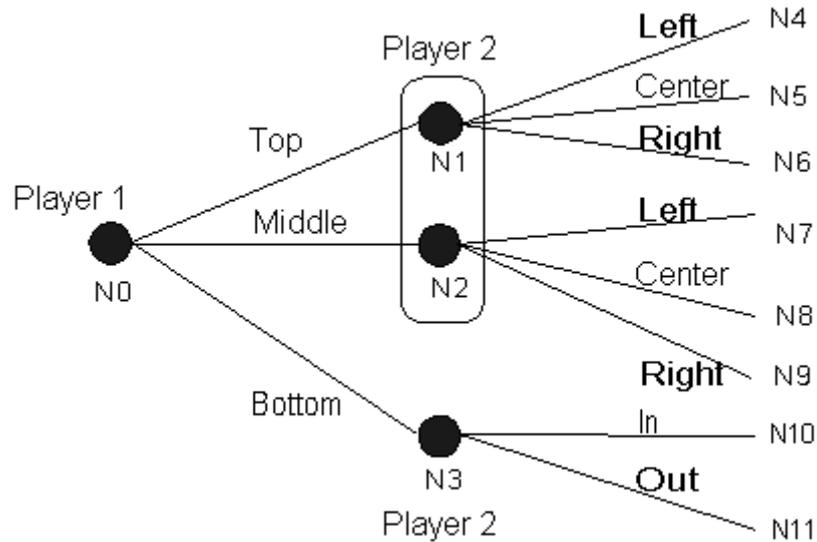


Figure 2-2 Extensive Game with Imperfect Information

In this game player 1 moves first and chooses “Top”, “Middle” or “Bottom”. If he picks “Bottom”, the game moves to N3 and player 2 chooses either “In” or “Out”, after which the game is over. But if player 1 picks either “Top” or “Middle” then the game moves to one of the two decision nodes N1 and N2 inside the oval. This means that player 2 knows that he is either at decision node N1 or N2 but he does not know which node he is at. Good examples of repeated extensive games with imperfect information are Black Jack, Poker, etc.

Because the players are only partially informed about the actions taken by the other players, it is not possible to have a nash equilibrium or subgame perfect equilibrium. The solution for extensive games with imperfect information is called sequential equilibrium [2] that consists of both a strategy profile and a belief system, where a belief system specifies, for each information set, the beliefs held by the players who have to move at that information set about the history occurred.

2.4 Player(s)

In all theoretical game models the basic entity is a player. A player may be interpreted as an individual or as a group of individuals making a decision. Then, the player(s) can act towards two types of models:

- The sets of possible actions of individual players are primitives, sometimes referred to as noncooperative.
- The sets of possible joint actions of groups of players are primitives, sometimes referred to as cooperative.

The previous games were all noncooperative games: a single player, or a group of individuals, takes each action autonomously. In cooperative games, often referred as coalitional games, a part of the total group of players can take actions independently of the remaining players.

To illustrate the differences between the two approaches, consider the following situation [2]. Each player owns a package of inputs and has access to a technology for producing a valuable single output. These inputs are unproductive in the player's own technology but productive in some other player's technology. A noncooperative model of this situation specifies precisely the set of action that is available to each player: perhaps each player can announce a price vector at which the player is willing to trade input, or perhaps the player can propose a distribution of inputs for the whole of the society. A coalitional model, by contrast, starts from the set of payoff vectors that each group of individuals can jointly achieve. A coalition may use contracts, threats, or promises to achieve a high level of production.

2.5 Application of Game Theory for Bidding for Software Projects

To be able to evaluate the bidding for software projects, some principles of the game theory are not applicable in this thesis.

The study can not be set up as a game in strategic form because then the players should define their possible actions once and for all. When defining the possible actions once and for all it is not possible to determine the bidding behaviour when a player reacts on a certain outcome of a previous action from another player. Coalition games are not applicable either because of the same kind of reason. The focus of this thesis is to evaluate the behaviour of the individual players in the game.

In the rest of this thesis when talking about a player this can mean an individual player or a group of individual players but not a coalition group of players that will achieve a better payoff when they cooperate. The only concept that is applicable in this thesis is an extensive game because then it is possible to determine the behaviour of each player for every bidding round in the game and it is possible to perform this game repeatedly. The next chapter describes some further details about applying an extensive game in this thesis.

3 Extensive Games

The previous chapter introduced the main concepts of game theory. It concludes that only extensive games are applicable in this thesis. The extensive games are described as a specification of the possible order of events. Each player can consider his plan of action not only at the beginning of the game but also whenever he has to make a decision. The previous chapter assumed that each player makes a decision after another player's decision. But in this thesis this is not the desirable situation because the basic idea of performing this research is to let the complete population bid simultaneously. And after each player's offer, they are informed with perfect or imperfect information. This chapter describes the important properties of perfect and imperfect information when applying simultaneously moves. The chapter concludes with benefits and drawbacks of applying either perfect or imperfect information in this thesis.

3.1 Perfect Information

Osborn and Rubinstein [2] define an extensive game with perfect information as: *“A detailed description of the sequential structure of the decision problems encountered by the players in a strategic situation. There is perfect information in such a game if each player, when making any decision, is perfectly informed of all events that have previously occurred”*. This section provides more details about extensive games with perfect information.

3.1.1 Repeated Games

The model of a repeated game is designed to examine the logic of long-term interaction. It captures the idea that a player will take into account the effect of his current behaviour on the other players' future behaviour. The basic idea behind repeated games is illustrated by the case in which two individuals repeatedly play the Prisoner's Dilemma (recall Table 2-1 on page 2). The main idea behind the theory of repeated games is that if the game is played repeatedly then the mutually desirable outcome in which both suspects do confess occurs in every period is stable if each player believes that a disloyalty will terminate the cooperation. This results in a following loss for that player that outweighs the short-term gain.

The primary achievement of the theory is to isolate types of strategies that support mutually desirable outcomes in any game. The theory gives insights into the structure of behaviour when individuals interact repeatedly.

A repeated game is an example of extensive games with perfect information. Another example is a bargaining game. This game deals with situations in which people's interest conflict. The people involved may try to resolve the conflict by committing themselves voluntarily to a course of action that is beneficial to all of them. With other words: cooperation. But as mentioned in chapter two, cooperation is not desirable in this thesis. The rest of this section mentions the important characteristics of repeated games that are useful in this thesis.

3.1.2 Infinite vs. Finite

The important characteristics (also referred to as models) of repeated games are infinite and finite, which means that the horizon of the games may be infinite or finite. According to Osborn and Rubinstein [2] a model should attempt to capture the features of reality that the players perceive; it should not necessarily aim to describe the reality that an outside observer perceives. However, obviously there are links between the two perceptions. So the fact that a situation has a horizon that is in some physical sense finite (or infinite) does not necessarily imply that the best model of the situation has a finite (or infinite) horizon. A model with an infinite horizon is appropriate when after each period the players believe that the game will continue for an additional period. And a model with a finite horizon is appropriate when the players clearly perceive a well-defined final period. In the case of an infinite repeated game, each move can be interpreted as a sub game because the players always have to consider that the current move can be the final move.

Both of the models have some considerations that should be taken in account:

- *Cheating*: in infinite repeated games, cheating is a risky event because a player never knows if the current sub game is the final iteration. For example if the player makes a bid to test the reactions of the other players, it can be the final bid for the player. In finite repeated games the players know when to stop cheating because of the final iteration is approaching.
- *Punishing*: a player that is cheating can/should be punished as well. In infinite repeated games this is (again) of course a risky event because of the same reason as cheating.
- *Shadow*: in finite repeated games the outcome in the last part of the complete game is in fact only interesting because here the players actually launch a well-considered move. So it casts a shadow over the rest of the game. In infinite repeated games this is not the case because each move can be the final move.

3.2 Imperfect Information

Extensive games with imperfect information were developed intensively only in the 1980's. In comparison with extensive games with perfect information, they are less mature. But extensive games with imperfect information can still be applicable in this thesis. As already explained in the previous chapter (section 2.3), the players in extensive games with imperfect information do only have partial information about the actions taken previously by another player. This section describes more about extensive games with imperfect information.

3.2.1 Concept

In each of the models described in the previous chapter there is a sense in which the players are not perfectly informed when making decisions. In a strategic game a player, when making an action, does not know the actions that the other player takes. In an extensive game with perfect information a player does not know the future moves planned by the other players.

Extensive games with imperfect information differ in that the players may in addition be imperfectly informed about some (or all) of the choices that have *already* been made. These choices rely on expectations about the unknown.

3.2.2 Perfect and Imperfect Recall

An extensive game is capable of capturing a wide range of informational environment. In particular, it can capture situations in which at some points players forget what they knew earlier. This is called a game with imperfect recall. But games with imperfect recall occur seldom. The kind of extensive games with imperfect information that occurs in most situations is games with perfect recall: players know at every point whatever they knew in the past.

3.3 Benefits and Drawbacks

This section describes the benefits and drawbacks of all the issues discussed in this chapter: extensive games with perfect information, extensive games with imperfect information, infinite games, and finite games. This section concludes with considerations of the benefits and drawbacks. The benefits and drawbacks provided in this section are benefits and drawbacks with respect to this thesis and not with respect to extensive games with (im)perfect information in general.

3.3.1 Perfect Information

The benefits and drawbacks of extensive games with perfect information are described below.

Benefits:

- A player can see and follow the other's behaviour/decisions precisely.
- Cheating and punishing are easy to apply because the players have all the information they need.
- It is easy to adjust a price to the offer of the other players. This implies that the risks are not so big.

Drawbacks:

- There is a risk that the players lose their focus. Instead of making an offer based on their own capacity, they only focus on the offers of the others.
- It demands a lot of time to come with a next offer because the players have to study a lot of data.

Besides these drawbacks of extensive games with perfect information, which are mainly focused on the players, there is also a drawback for the researchers. Because applying a game with perfect information results in a lot more work than applying a game with imperfect information since all the players need to be informed with all the information available.

3.3.2 Imperfect Information

The benefits and drawbacks of extensive games with imperfect information are described below.

Benefits:

- The players are not distracted by the bid of other players.
- Not so time consuming for any party: players and researchers.

Drawbacks:

- It is hard to apply cheating and punishing. The risk can be too big for the players.
- It is possible that the game causes panic moves of the players because they have no idea of how the other players perform the game.

3.3.3 Infinite and Finite Games

Infinite and finite games have also some benefits and drawbacks. Although infinite and finite games are described under the perfect information section, it is also possible to have infinite and finite imperfect games. But it is described in the perfect information part because it is slightly related to repeated games. The benefits and drawbacks described below apply both to perfect and imperfect extensive games.

Benefits infinite:

- The players have no idea of when a game will end, which makes the game exiting.
- The players offer a well-considered bid, because it can be the final bid.

Drawbacks infinite:

- There is a big chance that the players do not dare to take big risks because a game can end at any time.

Benefits finite:

- There is more action in the game because the players dare to cheat and punish.

Drawbacks finite:

- Although a game is more exiting, it still puts a shadow over most of the game.

3.3.4 Conclusions

To study and evaluate the bidding for software projects all the concepts of extensive games described in this chapter could be applicable in theory. And all of the concepts can produce interesting outcomes as well. But, to be realistic, just some of the concepts are applicable with respect to the establishment of software projects (more information in the next chapter).

Therefore, the study in this thesis should be performed as a:

- Game with imperfect information: when bidders place their bids, they never get to know the competitors' bid afterwards.
- Infinite game: bidders never know how long the customer continues to ask them to bid for a new software project, so the players should not know when the game will end.

An additional advantage of performing the study like described above is that the game can be stopped whenever possible (e.g. when sufficient information is collected).

The next chapter describes what is meant by software projects.

4 Software Projects

The previous chapters described only the first part of the title of this thesis: game theory. During describing game theory, the words software projects often appeared, without further explanation. This chapter describes what is meant by software projects. In order to understand what is meant by a software project, this chapter starts with a brief description of a project, followed by a section about software projects. The last section describes how software projects are established.

4.1 What is a Project?

This section describes what is meant by a project and the main characteristics of a project.

A project can be seen as a natural process that occurs in all dynamical systems: that of birth, life, and death. A project can for example be a wedding, moving, military campaign, etc. They all have a start, a performance, and an end. Nicholas [4] defines the following characteristics of a project:

1. A project involves a single, definable purpose, end-item, or result, usually specified in terms of cost, schedule, and performance requirements.
2. A project is a one-time activity, never to be exactly repeated again (unique).
3. Projects are temporary activities.
4. Projects cut across organizational lines because they need the skills and talents from multiple professions and organizations.
5. A project often involves unfamiliarity because of its uniqueness. It holds significant elements of uncertainty and risk.
6. The organization usually has a kind of backup when doing a project to perform a special investigation or effort because failure would jeopardize (endanger) the organization or its goals.
7. A project is the process of working to achieve a goal. During the process, projects pass through several distinct phases, called a project life cycle.

When a project involves the development of a system, the project life cycle is called a systems development cycle [4]. The next section starts with a description of different phases in a systems development cycle, followed by a description of the systems development cycle in software projects.

4.2 What is a Software Project?

The process of developing, implementing, and operating the system that involves a logical sequence of activities for any human-made system, is called the systems development cycle. Typically a systems development cycle is divided into four phases [4]:

- A. Conception: project initiation and project feasibility.
- B. Definition: project definition, system definition, user and system requirements.
- C. Execution: design, production and implementation.
- D. Operation: system evaluation and maintenance.

Actually, a project typically spans phase A through C. In this context, a project can be thought of as an organization that exists to develop a system, and the project life cycle as being the first three phases of the systems development cycle. So the stated description in the previous section that a project life cycle when developing a system is called a systems development cycle is not completely true. But because of explanation reasons stated like this.

But as the first sentence of this section mentioned, this approach is applicable for any human-made system. This can for example be a space station, a nuclear power plant, constructing a building, etc. The “Encyclopedia of Software Engineering” by Marciniak [5] gives the following definition of a software project: “A project is a one-time effort having well-defined objectives and occurring within a specific time frame. A project plan is prepared, people are assigned to the project, resources are allocated, and success criteria are specified. A software engineering project is a project in which the objective is to produce a software product, on schedule and within budget, which satisfies a set of requirements”. To be able to reach this objective, different models for the systems development cycle have been developed over the past. Boehm [10] describes for example the “waterfall” model. This model is one of the oldest and probably the most commonly used model [9] in software engineering. This model is illustrated in Figure 4-1.

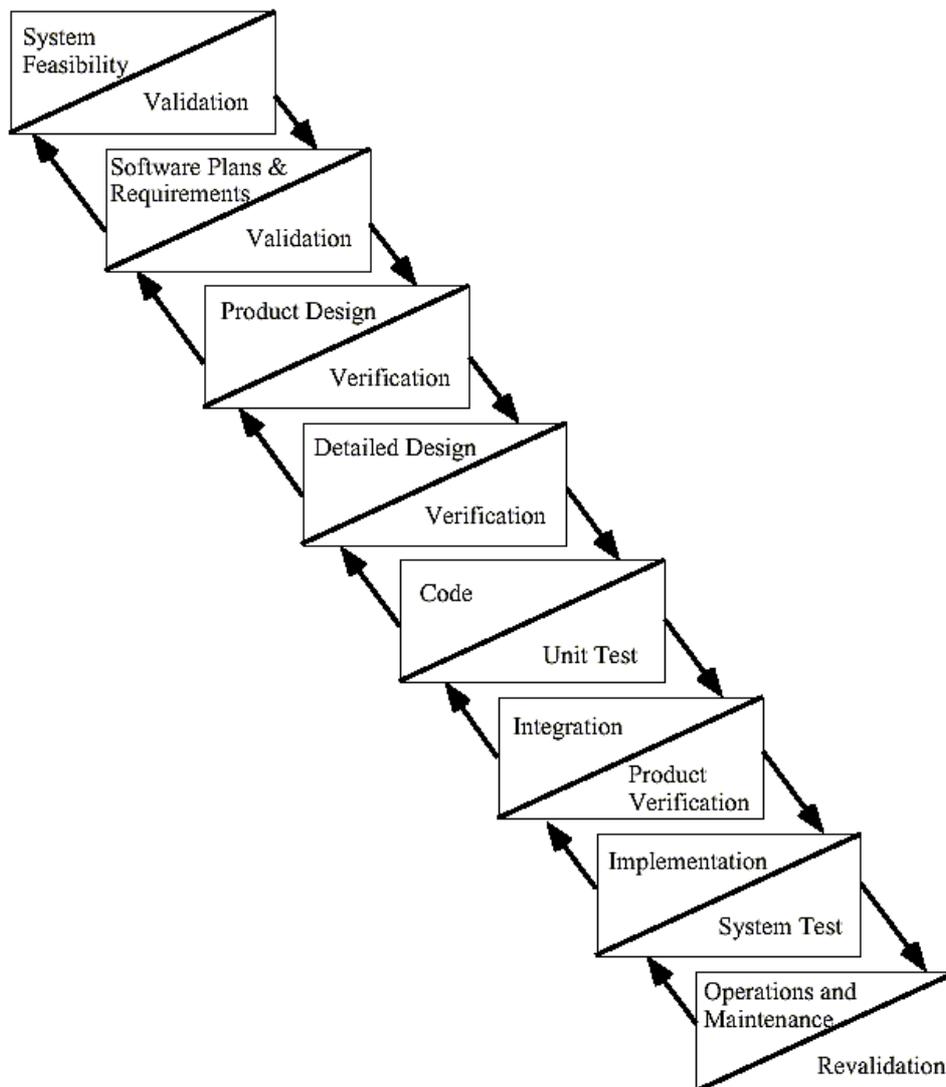


Figure 4-1 The waterfall model

The major overall features of the waterfall model are the following [10]:

- Each phase is ended by a verification (“Are we building the product right?”) and validation (“Are we building the right product?”) activity which objective is to eliminate as many problems as possible in the products of that phase.
- As much as possible, iterations of earlier phase products are performed in the next succeeding phase.

Besides this waterfall model, there exist several other models (often called software life cycle models in software engineering). But they will not be explained here because this is outside the scope of this thesis. The waterfall model is referred to as an example of a systems development cycle model in software engineering.

This thesis focuses mainly on phase A of the systems development cycle because it captures the initial stages of a software project. The phases B, C, and D are more applicable when reviewing software projects or when developing and implementing a system. An example of phase A in a software project is the first two stages of the waterfall model. But the rest of this thesis refers to the more general phase A.

Typically, there are two possible outcomes of phase A:

1. A system dies, which means that it is not realistic to develop a system: the project is terminated.
2. A software contract.

There exist several reasons of why a system could die, but these are not described because this thesis assumes that a system will be developed.

The next section describes the activities that are performed to establish software projects.

4.3 Establishment of Software Projects

Software projects are established in so called software contracts. A contract is defined as an agreement between two parties wherein one party (the contractor) promises to perform a service, and the other party (the customer) promises to do something in return – typically make a payment for the service. In software projects, also other names appear for these parties. A contractor is also referred to as a developer and a customer is also referred as a user or a client. But they still mean parties and not individuals.

To establish a software project, several activities are performed. This section describes briefly these activities, which are: conception, request for proposal, project proposal, and software contract [4].

4.3.1 Conception

The conception activity relates to phase A of the systems development cycle. As mentioned in the previous section, the conception phase comprises two separate stages.

The first stage, *project initiation*, establishes that a “need” exists and that the need is worthy of investigation. This stage begins when the user perceives a problem, need, or opportunity. With other words the user has an “idea”. To select the few good ideas, the user organization undertakes a brief, initial investigation. This investigation consists of interviews, background research, and review of existing documentation. The emphasis is on breadth, not depth.

During the second stage, *project feasibility*, a detailed investigation is conducted and a solution (in sufficient detail) is developed to determine if it is economically viable and worth of development. The purpose of this stage is to establish which solution and which developer, if any, are best. The user initiates this stage by creating a request for proposal to find a developer to perform a feasibility study (from the developer's point of view).

4.3.2 Request for Proposal

To find the applicable developer, the user composes a request for proposal (often abbreviated as RFP). Writing this RFP has two major purposes:

- To outline the user's idea (problem, need, etc.).
- To collect suggestions (proposals) for solutions, usually with the intent of awarding a contract for the best one.

The user describes his problems, objectives, and any requirements in the RFP. Nicholas [4] describes a typical RFP. Figure 4-2 shows the contents of this typical RFP.

<p><u>Statement of Work</u></p> <p>a. Description of problem, need, or general type of solutions to be investigated.</p> <p>b. Scope of work to be performed by contractor, work to be included, work excluded, and work restrictions; criteria of acceptance for result or end-items.</p> <p>c. Requirements for the solution, results, or end-item, including specifications and standards; description of how work will be measured; expected relationship between user and contractor; expected completion date, constraints on cost of work to be performed.</p> <p><u>Proposal Requirements</u></p> <p>Conditions placed on the proposal such as proposal contents and format, data requirements, sample forms to include, and submission location and deadline.</p> <p><u>Contractual Provisions</u></p> <p>Type of contract to be awarded, sample contract, and non-disclosure provisions.</p> <p><u>Technical Information or Data</u></p> <p>Any additional data, or name of a contact person for requesting additional data, necessary to develop a solution and prepare the proposal or price quote.</p>

Figure 4-2 Contents of a Request for Proposal

An RFP is sent to the potential developers from the user's bidding list. These developers respond by sending a project proposal.

4.3.3 Project Proposal

One of the first steps of the developer in response to an RFP is to develop a more concise, accurate statement of the problem. This is the way to gain a full and complete understanding of the user's problem situation. After this step the developer composes a project proposal.

Typical characteristics of a project proposal are:

- Executive Summary: to convince the user that the proposal is worth considering.
- Technical Section (Statement of Work): indicates the scope of the work, realistic benefits, and a schedule when end-items will be delivered.

- Cost and Payment Section: breaks down projected hours for direct, indirect, and special activities, associated labour charges and materials expenses, and price of project.
- Legal Section: contains anticipated, possible, or likely problems and provisions or contingencies.
- Management/Qualifications Section: consist of background of the contractor organization, related experience and achievements, and financial responsibility.

This project proposal should be sent to the user. The user evaluates each proposal according to criteria related to system performance, price, and schedule. Based on the results of this evaluation, the user selects the desirable developer. So it is obvious that the project proposal is the most important activity for a developer in order to get a software contract. The developer that is able to offer an accurate proposal gets the software project. The other potential developers are finished. So it is essential to perform a thorough cost estimate and offer an accurate bid. The next chapter describes more about costing and bidding for software projects.

If the user selected the desirable developer, it is confirmed in a software contract. The next section describes the final activity: software contract.

4.3.4 Software Contract

As stated in the beginning of this section, a contract is an agreement between two parties wherein one party (the contractor) promises to perform a service, and the other party (the customer) promises to do something in return – typically make a payment for the service. This agreement is a result of negotiating the contract, whose purpose is to clarify technical or other terms in the contract and to reach agreement on time, schedule, and performance obligations. The negotiation is also the last opportunity to correct misperceptions that might have slipped through the RFP. Actually a contract is nothing more than a confirmation of who will do the work stated in the RFP and a specification of the customer's role in tracking progress and making trade-off decisions.

More formally, Whang [7] describes, among other things, a summary review of software developing contracts. This review concluded that a software contract consists of three parts:

1. Product Definition: The contracts include the definition of the product, services and delivery conditions.
2. Intellectual Property Protection: The contracts specify the ownership for each contingency, like license to the software and its side-products, rights to new ideas, patent protection, ownership of copyright to the source code and supporting documentation, rights to modify and enhance the software, restriction in entering into the other party's business as a competitor, restriction in hiring away key employees of the other party, and protection of trade secrets and confidentiality.
3. Payment: Explanation of the software development fee by software license, service, training, maintenance and documentation, or/and by the modules of the software.

But still, it is basically the RFP that contains the explanations of what should be done and how it should be done. As stated before, a contract is more a formal confirmation.

As mentioned earlier, the next chapter provides more information about costing and bidding for software projects.

5 Costing and Bidding

As stated in the previous chapter, it is essential to perform a thorough cost estimate and offer an accurate bid in order to obtain a software project. This chapter describes some more details about costing and bidding for software projects. But before describing the details regarding software projects, this chapter starts with an introduction of what is meant by costing and bidding in general. The second section describes the costing and bidding for software projects. This chapter concludes with a discussion of the focus of this thesis.

5.1 Introduction

In order to understand costing and bidding for software projects, it should be clear what is meant by costing and bidding. Some dictionaries/encyclopaedias give the following definitions.

Costing:

- Calculated cost of something: the cost that has been calculated for undertaking a project [11].
- Calculations of future cost [12].

Bidding:

- The act or process of making bids; an offer; a proposal of a price, as at an auction [13].

The last definition is perhaps still unclear. What is meant by bids? The Encarta World English Dictionary [11] gives the following definition of a bid: “An offer to do a piece of work for a particular price.”

Concluding, this implies that the difference between costing and bidding is that costing relates to the estimated costs and bidding relates to the offered price to perform the work. The next sections give more details about costing and bidding related to software projects.

5.2 Software Projects

This section examines the two essential activities to obtaining a software project: costing and bidding.

5.2.1 Costing

As stated in the introduction of this chapter, costing is the calculation of costs. In projects, the activity of costing is often called cost estimation. There exist several techniques to perform cost estimation. Fenton and Pfleeger [14] make a distinction between four techniques for estimating (expert opinion, analogy, decomposition, models) and Boehm and Sullivan [15] make a distinguishing of six techniques to estimate costs (expertise-based, model-based, regression-based, composite-Bayesian, learning-oriented, and dynamic-based). In spite of using completely different distinctions both conclude that the decomposition [14] or more specifically the work breakdown structure (WBS) [4, 15] is the most commonly and preferred used technique. Both [14, 15] also relate to a model [14] or model-based [15] technique in software engineering: the constructive cost model (COCOMO) [10].

Both techniques have some strengths and weaknesses [10, 14, 15]:

Strengths COCOMO:

- Based on the waterfall method.
- Specially developed for cost estimates in software engineering projects.
- Therefore it contains a lot of software project characteristics.

Weaknesses COCOMO:

- Uses 15 so called cost drivers that are expected to influence effort or duration. They are divided in product, process, resource and project characteristics. This makes the technique complex.
- Focus is on size, like product size and delivered source instructions.
- The price is just a derived result.

Strengths WBS:

- Based on real experience.
- Easy to adjust for special situations.

Weaknesses WBS:

- General. But this can also be seen as strength: everybody knows/understands it.
- Estimator bias (\approx prejudice) and the representatives of the experience upon which the estimates are based [15].

The applicability of these techniques in this thesis is discussed in chapter 7, the study.

5.2.2 Bidding

Bidding is the process of making bids. And each bid is the offer to do a piece of work for a particular price. This particular price is a result of the addition of [4]:

$$\text{Price} = \text{Cost} + \text{Profit}$$

The cost is derived from the costing activity and the profit is the desirable amount that the bidder would like to earn for performing the project.

The price agreements are usually of two types: fixed price or cost-plus. In the fixed price agreement, the price is agreed upon and remains fixed as long as there are no changes of the scope or provisions of the agreement. In the cost-plus agreement, the contractor (developer) is paid for all or some of the expenses occurred during the performance of the agreement, and as a result, the final price is unknown until the project is completed.

The next section discusses the focus of this thesis.

5.3 Focus of Thesis

Because the aim of this thesis is to evaluate the bidding for software projects and not the costing, the focus of this thesis is on the bidding activity. The next chapter provides an evaluation of existing studies performed on bidding behaviour. Chapter 7 provides detailed information about how the costing part is covered in the study, the type of price agreement, etc.

6 Bidding Behaviour

The previous chapter described already what bidding means. This chapter provides information about existing studies performed on bidding behaviour. The first section gives an introduction to bidding behaviour and some general studies performed on bidding behaviour. The second section offers some results of studies performed on game theory related to bidding behaviour. The last section gives a motivation for the conducted study.

6.1 Introduction

Before going into details, it should be clear what bidding behaviour is. According to the Cambridge Dictionaries Online [12], behaviour is a way of action. So when merging this together with the definition of bidding in section 5.1, bidding behaviour can be described like: “A way of action when performing the process of making bids”.

Bidding behaviour is a topic that is reasonably investigated in economical articles. Bodington [16] investigates the bidding behaviour on acquiring domestic generating assets (e.g. gas and electric). One of his main conclusions is that bids of less than 50% percent of the average bid are common, and bids between 150 percent and 200 percent of the average are not unusual. The main reason to intentionally submit a high bid is to gain an exclusive negotiation position. But for 3 of the 4 investigated projects, the highest bids were rejected because of the risk. Another conclusion was that well-informed and experienced bidders often have substantially different views on the value of a project (i.e. future energy prices, operating costs, contract changes, etc.). Another article, written by Meyer [17], provides some evidence from the rice market by investigating competition and bidding behaviour. His main conclusion is that increasing the competition (number of bidders) increases the bid, even when the product has an uncertain value.

The results of both articles are interesting, but they only partly cover the aim of this thesis or have another approach:

- They are not based on games, so there is no data available on how the bidding behaviour is established.
- The bidding behaviour data is a result of investigating auctions. But auctions have some characteristics that are not applicable in this thesis. Section 6.3 explains why auction data is not applicable in this thesis.

The next section describes more about game theory related to bidding behaviour.

6.2 Game Theory

To investigate behaviour, a game is a good approach because a game offers:

- Competition.
- A returning observed phenomenon.
- Players’ strategies.
- Different views/opinions of players.
- Intuition of players.
- An outcome (a winner).

Games are also used to investigate bidding behaviour. Although not all of the articles refer to game theory, the studies certainly contain some characteristics of game theory. For example Ivanova-Stenzel [18] uses a game to perform an experiment carried out in two different social and economical cultures to analyse the bidding behaviour. Four different game types were implemented to evaluate the bidding behaviour and to draw the conclusions. Her main conclusion was the similarity of individual behaviour in both countries (Germany and Bulgaria). But no other conclusions of individual bidding behaviour were drawn. An article that uses game theory to investigate bidding behaviour is the article written by Pitchik and Schotter [19]. The article presents an experimental study of bidding behaviour in sequential auctions where there are budget constraints and perfect information. Their main conclusion is that budget constraints affect the behaviour of bidders. Berk et al. [20] investigate the television game show “The Price Is Right” as a laboratory to conduct a preference-free test of rational decision theory. In this game, four opponents sequentially guess a retail price of durable goods without going over the retail price. Their main conclusions are that the fourth player has the biggest chance to win and that learning during the show reduces the frequency of strategic errors.

The next section provides a motivation for conducting this study and what is lacking or not investigated yet in the previously described articles.

6.3 Motivation

The articles described in this chapter have one major drawback in common of why they are not applicable in this thesis: they are based on auctions. Recall the extensive game concept with perfect information (sections 2.2 and 3.1). An auction fits perfectly in this game concept because this concept contains a sequential structure: a player offers a bid after another player. But this study focuses on simultaneous moves. There exists one kind of auction that captures the idea of simultaneous moves: sealed bid [22]. In sealed bid auctions the bidders bid simultaneously and they do not know the other’s bid. But no empirical data could be found about this kind of auctions related to bidding behaviour in extensive games with imperfect information.

Another important drawback of auctions is that in auctions the highest bidder wins. But in this study it is the other way around: the lowest bidder wins (this issue is more described in the next chapter). There is one type of auction that fulfils the criteria of the lowest bidder wins: Dutch auctions [23]. In Dutch auctions, the seller lists multiple identical items and all winning bidders pay the same price: the lowest successful bid. The problem with this kind of auctions is (for this thesis at least) that bidders bid for multiple items with the idea of the more we buy, the cheaper it becomes. A good example of a Dutch auction is the internet site eBay [24].

Besides these major drawbacks the articles contain some specific lack of information, but it is outside the scope of this thesis to discuss them because they will not have any contribution.

It can be concluded that auctions are seen as the ‘normal’ bidding process and that the focus of this thesis is rather new: no empirical data could be found about extensive games with imperfect information related to bidding behaviour, and in particular in the software domain. And this was already the hypothesis by initiating this thesis.

Bidding for software projects is used because of two major reasons:

1. Introducing game theory in the software domain for evaluating the bidding behaviour of software engineers.
2. As an object to let the software engineers bid for. It does not make sense to let the software engineers bid for building a bridge for example.

The next chapter describes the performed study.

7 Study

This chapter describes the performed study that makes it possible to give an opinion about the bidding behaviour of software engineers. To give a complete description of the study, this chapter starts with a repetition of the applicable parts of game theory. The second section provides a description and motivation of the experimental design document. The last section of this chapter provides information about the players of the game and the rules/constraints for the players.

7.1 Game Theory

As stated in section 3.3.4, the applicable part in this thesis is an infinite extensive game with imperfect information. In this game, the software engineers bid simultaneously for software projects for an unknown period of time.

During the performance of the study, there are two parties involved:

- The researcher who acts as the customer by offering software projects to bid for.
- Software engineers who act as players that make a bid for software projects.

The performance of the study where these 2 parties act in is called the game.

To be able to evaluate the bidding behaviour of the software engineers, they all receive an experimental design document. The next section describes the motivations and the content of this document.

7.2 Experimental Design

The experimental design is a document that all the players receive. This document describes the general instructions for the players, like an overview of the game, goals of the game, rules of the game and how the game is performed. It also contains a pre-estimated cost of a project which can be a result of the conception phase (see section 4.3). But this pre-estimated cost is just a fixed amount of units. It is decided that all the projects have a pre-estimated cost of 100 units. The actually estimates are not performed because the emphasis of this game is on the bidding behaviour and not on the exact cost. So the costing activity stated in section 5.2.1 is handed to the players.

The activity that the players are asked to perform is bidding (section 5.2.2). Their aim is to gain at least an average profit of 15% above the pre-estimated cost. In this study this implies that the players have to try to earn an average of 115 units.

Some of the rules are hidden for the players and some other rules need some motivation. The next section gives more explanation of certain rules.

The other information described in the experimental design document is straightforward and can be found in Appendix A.

7.3 Rules

The experimental design in Appendix A explains most of the rules of the game. But some rules are just given to the players without any further explanation and some rules are not given to the players at all. Besides these two types of rules, there is also a lack of rules. This section provides some explanations about all of these certain rules.

Number of rounds

The number of rounds is not given to the players because the game is set up as an infinite game with imperfect information. This choice is already made in chapter 3 and section 3.3.4 provides the motivations for this choice. But the plan is to perform approximately 10 rounds because this provides sufficient information to draw some conclusions within the feasibility of this thesis.

Type of price agreement

The experimental design gives no information about the type of price agreement because of the following reasons:

- Not important for evaluating bidding behaviour: no extra value to have this property in the game.
- The players probably do not have any knowledge about the types of price agreement so they will not consider this either.
- It can create unnecessary confusions.

So the type of price agreement is a fixed price: once agreed upon, the price cannot be adjusted.

Price for Bidding

It is decided not to have a price for bidding because the cost is the same for all the players. There is a chance that a player bids the same price all the time, without costing anything. But with this approach the player seldom wins a project.

Budget

It is also decided not to give the players a budget because this gives the players an idea of when the game can be ended. And this is in contradiction with an infinite game with imperfect information. But to be able to determine the overall winner, the costs for each player are logged during the game. So the player that makes the most profit and collects the most number of projects, wins the game.

Boundaries

To be able to perform the bidding process as realistically as possible, the game contains a kind of boundary in a form of distrust value. Every player starts with a distrust value of zero and the maximum value is twenty-five. Every time a player underbids, the player's value increases. But the value can never decrease. If a player reaches the value of twenty-five, all the player's further bids are ignored without notification to the player. So the player continues bidding, even without winning projects. This is the price the player has to pay. It is decided to let the player continue because this is also common in real situations. It is also decided not to have a distrust value for overbidding. It is possible but it does not add any value to this study because with a high bid, a player will never win anyway.

Another kind of boundary is the threshold boundary. For example if two bidders bid with a certain difference, it is not always the case that the project goes to the lowest bidder. If the higher bidder won the previous project, why not continue with this bidder? The threshold boundary is a fixed amount of five units. The players do not know this exact figure, but they know that there exists a certain internal rule.

7.4 Players

Before the actual start of the game, the players are selected. It is decided to select software engineers from the Department of Software Engineering and Computer Science (called IPD) of Blekinge Institute of Technology, Ronneby, Sweden.

To select the software engineers they have to meet some requirements. According to the TechEncyclopedia [22] a software engineer is: “*A person who designs and programs system-level software, such as operating systems, database management systems (DBMSs) and embedded systems. The title is often used for programmers in the software industry that create commercial software packages, whether they be system level or application level*”. But this definition is modified to a more general description: A person that has reasonable experience in developing software. The main reason of modifying the definition of TechEncyclopedia is that most of the software engineers at the IPD department did not see themselves as software engineers according to this definition.

The main reasons to select the software engineers from IPD are:

1. Easy to communicate with them because they are in the near surroundings.
2. It demands relative less effort to perform the selection procedure.

Even that the players were classified as software engineers, they still vary a lot. Two of the four were primarily teachers and the other two were researchers. Both researchers had already knowledge of game theory and the teachers did not. It should also be noted that they all know each other so it cannot be guaranteed that they did not talk to each other, but probably they are so competitive that they did not do.

Probably it does not make a big difference if the software engineers are working at a university or in the software industry because in a real life situation it will not be the software engineers who offer a bid, but a customer service in close cooperation with the software engineers. So the customer service represents the opinions of the software engineers. But in this thesis, the software engineers are directly addressed because the aim of this thesis is to retrieve information about the behaviour of the software engineers.

7.5 Game Performance

The game is initiated with an introduction meeting where all the selected players are invited. During this meeting they get a short presentation of the thesis, they all receive the experimental design description (see Appendix A) and the possibility to ask questions.

After this introduction meeting, the actual game starts. This game is performed as an iterative process with the following activities in each round of the game:

- The players place their bids for a new project, but the costs will be the same. They also offer a short motivation of the strategy for their specific bid (to understand their behaviour better).
- The customer collects all the bids and motivations.
- The customer determines the winner (only based on the bid, not the motivation).
- The customer informs all the players with the following information:
 - o If the player is the winner or not.
 - o If the game continues, or if it was the final round.

The game can be terminated at any time. The major reasons to terminate a game are:

1. Enough information is collected: some conclusions can be drawn and certain behaviour can be determined.
2. There is no significant difference with the previous round: it is not desirable to continue. Perhaps identify a sequential equilibrium.
3. The game runs into an unexpected situation.

The next chapter provides the results of the study (game).

8 Results

This chapter provides the results of, and a discussion about, the performed study. Table 2-1 shows the offered bid for each player each round of the game.

		Round								
		1	2	3	4	5	6	7	8	9
Player	1	101	109	114	114	114	114	115	115	11500
	2	102	89	96	115	135	125	125	120	114
	3	75	90	90	90	70	NB	NB	NB	NB
	4	99	115	115	115	98	120	120	117	118

Table 8-1 Offered bid for each player and each round

The bold numbers in the table represent the winner of each round. To get a better overview, the table is converted into a graph that shows the differences, and/or similarities, between each player during the game. This is shown in Figure 8-1.

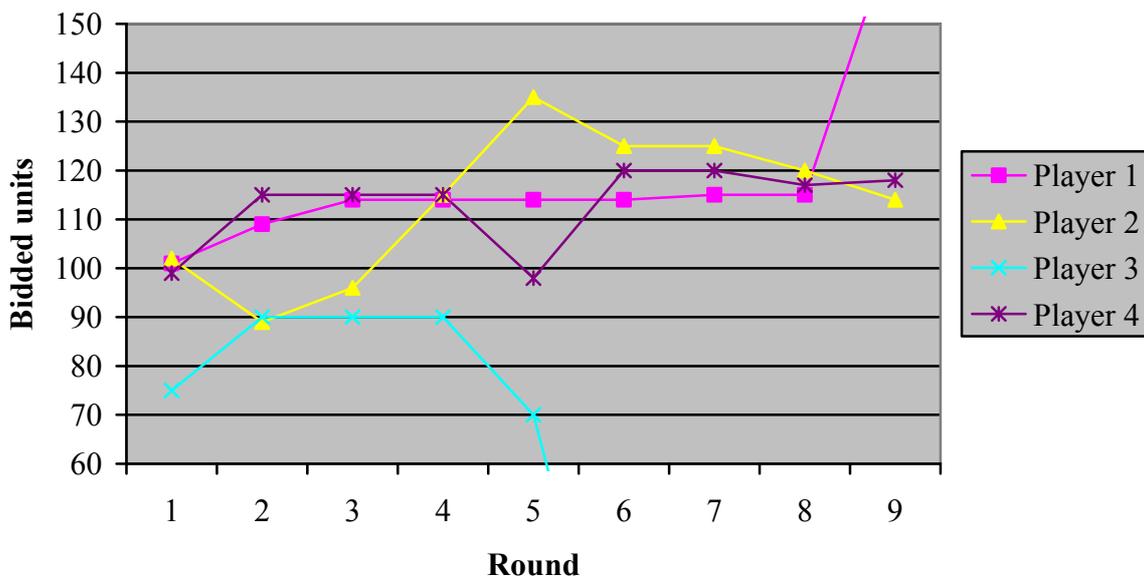


Figure 8-1 Differences, and/or similarities between each player

One of the first things that attract attention is player 3. The player started by offering a bid of 75 units. This number of units was exactly on the boundary of acceptance, but with this number of units the player lost all the credits of the distrust value (25). The player's motivation was that if you are new, you have to conquer the market. But then the player under bid again. So therefore all of the following bids are ignored. Probably player 3 noticed this and stopped bidding after several rounds (NB=No Bid).

It is interesting to see that the other three players started the game similarly. They all placed a bid around the cost unit. Their general motivation was that they do not dare to take too much risk, but they all admit that it has to cost to establish a relationship with the customer. Then two of the three players decide to go back to the initial conditions and know that if they do not

win a project, other players do not earn money. Only player 2 decides to under bid with the following motivation*: "Since the last bid round did not go that well, I am aiming at winning a bid and become one of the co-operators for the customer. Have to win a bet to be able to do this." In the third round, player 2 under bid again, but not so low: 96. The player's approach was that in order to achieve a relationship with the customer, it may cost some money in the beginning. But in the fourth round the player tries to climb towards break even by bidding 115. Player 2 wins this round based on the threshold boundary of 5 units. But at the moment that player 2 tries to recover from loses, player 4 decide to under bid. The player's motivation was: "98: From time to time one must try to get a project (and hope for the customer to stick to me long enough for me to gain it in later projects)." So player 4 wins this round. But player 4 will only succeed for one round because player 1 still bids 114 and player 4 bids 120. Player 1 was a really stable bidder. The player's opinion was that everyone must soon understand that the only way to get around 15% profits is not to under bid. Player 1 keeps the stable bid around 115 units and the other players do not dare to under bid again. They know that if they win a project with an under bid, they actually loose money where they have to recover from in the next rounds. So therefore player 1 wins three rounds because the other two players can not compete anymore. Only when player 1 tries to make more profit, the other two players will have the opportunity to win a project without losing money. This happens in round 9, where player 2 wins again a project.

This last round (9) is a kind of special one because of 2 reasons:

1. Player 1 thinks that the game is set up to fool the players by letting all the players win several projects and see their reaction.
2. Player 4 starts to offer a standing bid by applying an exponentially increasing payoff. Depending on if the player wins or not, a calculation is performed to offer the next bid. But this is going to look as a strategic game (see 2.2 Game Concepts).

After this round it is decided to stop the study because:

- The situation described above (the 2 reasons of the special last round).
- Resource problems (due to approaching summer break).
- The initial plan of performing approximately 10 rounds in which sufficient data is collected.

Table 8-1 also shows that there is no player that reached the goal of earning 15 units on the average and that player 2 collected the most projects. So when only looking to the number of projects, player 2 can be announced as the winner of the game. But when also looking to the highest earnings, which also was an important constraint in the game, player 1 is the overall winner with 3 projects and an earning of 44 units.

So when looking to player 1's behaviour the main conclusion of this study is that taking risks do not pay off when other bidders do not take much risk. When a player decide to take a risk and there is/are other player(s) who do not take risks at all (meaning bidding around the stated average earnings every round) it is really hard to recover from this loss, at least if the game is performed around ten rounds. Perhaps in the really long run a player will be able to recover, but this is outside the feasibility of this thesis and therefore no evidence is available.

* All the individual motivations are available in Appendix B – Player's motivations

It is too early to say that a sequential equilibrium has been identified, in particular with the behaviour of the first bidder in the last round. However, it seems reasonable to believe that a sequential equilibrium may occur if the bidders continued the game. This is however an issue for further research.

9 Conclusions

This thesis shows that the game theory contains some concepts that make it possible to evaluate bidding for software projects. These concepts are used in a study to evaluate the bidding behaviour of software engineers. The performed study concludes that when a bidder takes a risk, it is really hard for this bidder to recover when there is at least one bidder that does not take risks.

This thesis also provides evidence that there does not exist any data about bidding for software projects and that game theory is a new approach to evaluate bidding for software projects. And even for projects in general. So it is not possible to compare bidding for software projects with bidding for other kind of projects. The only existing data is bidding behaviour in auctions where the bidder of the highest price always wins. But this is a completely other constraint and therefore it is not possible to compare.

Because it is impossible to compare the data with other projects it is not possible to say what is typical for bidding for software projects. But this thesis shows that the establishment of typical software projects consist of a conception, request for proposal, project proposal, and software contract activity. It also shows that the project proposal activity contains, among other things, of the bidding and costing activity. To calculate costs several techniques exist. The most used are the general WBS model and the more specific software engineering model COCOMO. But these techniques are not used in the study because this thesis focuses on the bidding activity and not the costing.

10 Discussion

The purpose of this thesis was to introduce a new approach in a certain area of the software engineering: game theory into the software project area, and to verify if (parts of) game theory is applicable for bidding for software projects. With this thesis I hope I showed that (some parts of) game theory is applicable to investigate bidding for software projects and that I opened some doors that were closed before. But some marginal notes should be pointed out.

Game theory is a comprehensive collection of analytical tools to analyse decisions. Due to my background, the purpose of this thesis, and the feasibility of this thesis I decided to use game theory mainly as a practical tool and not as a mathematical tool to analyse the decisions (i.e. formulas).

Unfortunately the study was performed with only four players. The intention was to perform the study with approximately seven players, so the results would be more statistically convincing. To gather a group of seven players, eleven possible participants were invited because there are always some that are not able to participate. But it worked out that just four players were able to participate in the study.

Besides the lack of the number of players, another lack can be the number of performed rounds. The study is performed for nine rounds because of several reasons (see chapter 8). But it gives a good first impression about the bidding behaviour of software engineers.

The last issue that can be classified as deficient is that the study lacks reality. The study is performed in theory, without practical issues. It is even possible that this thesis lacks some characteristics of the bidding process, but I was unable to figure this out because it is mainly the theory where this thesis relies on. Another kind of reality deficiency could be some other characteristics like a cost estimates activity, a quality adjustment and a deliverance time adjustment. But I was unable to add these characteristics due to the feasibility of this thesis.

11 Future Work

As pointed out in the previous chapter, game theory is mainly used as a practical tool. With the possibility to have/gain more knowledge of game theory it could be interesting to take a more mathematical approach to the study.

But even without gaining more knowledge, it is possible to improve the results of the study rather easily. One possibility is to gather approximately seven players, perform the study again and see if the results differ. Another possibility is to perform the study again, but for a longer period of time to see if there is a significant difference, if the players will be able to better recover from their losses, etc.

The last issue that could be improved in the future is the lack of reality. It would be interesting to perform and evaluate the study with real software projects. Then it is also possible to figure out the real bidding process and to see if it differs from theory. Then it would also be possible to add the other characteristics (characteristics like a cost estimates activity, a quality adjustment and a deliverance time adjustment).

References

- [1] Johansson, S.J., “Game Theory and Agents”, Licentiate thesis, Department of Software Engineering & Computer Science, University Karlskrona/Ronneby, Sweden.
- [2] Osborn, M.J. and Rubinstein, A., *A Course in Game Theory*, The MIT Press, second edition, 1995.
- [3] Nachbar, J.H., (spring 2001), Washington University, Department of Economics, Econ 467 Course Page, Game Theory Notes (GameTheory.pdf), “Basic Non-cooperative Game Theory”, <<http://economics.wustl.edu/~e467f01/coursepage.html>> (15 November 2001).
- [4] Nicholas, J.M., *Project Management for Business and Technology*, Prentice-Hall, second edition, 2001.
- [5] Marciniak, J.J., (Editor-In-Chief), *Encyclopedia of Software Engineering, Volume 2 O-Z*, John Wiley & Sons, Inc., 1994.
- [6] Elitzur, R and Wensley, A.K.P., “Using Game Theory to Analyze Complex Projects: The Case of Information Systems Outsourcing Arrangements”, *International Journal of Industrial Engineering*, 6 (2), 141-150, 1999
- [7] Whang, S., “Contracting for Software Development”, *Management Science*, Vol. 38, No. 3, March 1992, 307-324.
- [8] Konakayama, A., Mitsui, T., Watanabe, S., “Efficient Contracting with Reliance and a Damage Measure”, *The Rand Journal of Economics*, Vol. 17, Iss. 3, Autumn 1986, 450-457.
- [9] <www.whatis.com> (2 March 2002)
- [10] Boehm, B.W., *Software Engineering Economics*, Prentice-Hall, 1981.
- [11] Encarta World English Dictionary <<http://encarta.msn.com/>> (24 March 2002)
- [12] Cambridge Dictionaries Online <<http://dictionary.cambridge.org/>> (24 March 2002)
- [13] ARTFL Project: 1913 Webster's Revised Unabridged Dictionary <http://humanities.uchicago.edu/forms_unrest/webster.form.html> (24 March 2002)
- [14] Fenton, N.E., Pfleeger, S.L., *Software Metrics: A Rigorous and Practical Approach*, International Thomson Computer Press, Second Edition, 1997
- [15] Boehm, B., Sullivan, K., “Software Economics: Status and Prospects”, *Information and Software Technology*, 41 (1999) 937-946, Elsevier Science B.V.
- [16] Bodington, J., “Bidding Behavior”, *Independent Energy*, October 1999, 32–35.
- [17] Meyer, D.J., “Competition and Bidding Behavior: Some Evidence From the Rice Market”, *Economic Inquiry*, Vol. XXVI, January 1988, 123-132.
- [18] Ivanova-Stenzel, R., “Auctions and Fair Division Games – a Cross-country Bidding Experiment”, *Journal of Socio-Economics*, 30 (2001), 367-374
- [19] Pitchik, C., Schotter, A., “Perfect Equilibria in Budget-constrained Sequential Auctions: an Experimental Study”, *RAND Journal of Economics*, Vol. 19, No. 3, Autumn 1998.
- [20] Berk, J.B., Hughson, E., Vandezande, K., ”The Price Is Right, But Are the Bids? An Investigation of Rational Decision Theory”, *The American Economic Review*, Vol. 86 No. 4, September 1996, 954-970
- [21] TechEncyclopedia <<http://content.techweb.com/encyclopedia/>> (1 May 2002)
- [22] Dufwenberg, M., Gneezy, U., “Information Disclosure in Auctions: An Experiment”, *Journal of Economic Behavior & Organization*, Vol. 48 Issue 4, August 2002, 431-445
- [23] “Auction Cheat Sheet”, *Pc Computing*, Vol. 13 Issue 2, February 2000, p151, 2/5p
- [24] eBay <<http://www.ebay.com/>> (8 September 2002)

Appendix A

Experimental Design

Introduction

You, as being a software engineer, have been selected to participate in a bidding behaviour experiment.

Background

The goal of performing this experiment is to obtain an understanding of the bidding behaviour of software engineers for typical software projects. With typical software projects is meant a project that is for example based on the Waterfall model and contains typical phases like project feasibility, project definition, system definition, user and system requirements, design, production and implementation.

Overview

To be able to understand the bidding behaviour, a game is developed. This experimental game involves the offer of several projects, one at a time. In each round of the experiment, a new project is offered which you can bid for. The cost of each project you will bid for is pre-estimated and will not be changed during the game. Your goal is to collect the most projects and earn as much as possible by using a certain strategy. You can 'win' a project if you bid the lowest amount. Your earnings (or losses because you are also allowed to bid under the pre-estimated cost) will be calculated like:

$$\text{Earning} = \text{Bid} - \text{Cost}$$

It is not allowed to offer other adjustable factors that influence the determination of a winner, like quality and time. This is not feasible within the experiment.

There is just 1 bidder who can win a project every round. But be aware that the lowest bidder does not always win a round! Historical data can also influence a decision to continue with a certain software engineer instead of changing to another. For example both parties (provider of the project and software engineer) are satisfied, so why change? But you will not be informed about this kind of information. During this experiment all prices are stated in terms of a fictitious currency called units. The main reasons are that it avoids misinterpretations (for example different currencies) and that the exact amount does not matter as long as the costs are the same for all the bidders.

Pre-estimated Cost

For every round of the game you will provide a bid for a new project. All the projects have a pre-estimated cost of 100 units. Your bid can be any positive whole amount of units. In theory, you can bid from 0 units to infinite, but in practice the game consists of some boundaries that you, as a bidder, do not know. A bidder can be excluded from the game without giving further explanations.

Payoffs

Your aim is to gain an average profit of 15% above the estimated cost. So for the projects in this experimental game this means 115 units. But you are free to use any kind of strategy (as long as it is between the boundaries of the game) to gain this profit. But every time you bid under the estimated cost and you win a project, it will cost you the number of units you

underbid. It is up to you how you will compensate this loss. If you do not win a project, but you have also underbid, it will not cost you any units.

Number of Players

You do not know the exact number of players participating in the experiment because in real bidding situations bidders do not have this knowledge either.

Number of Rounds

Part of the game is that you do not know the number of rounds. So you do not know how many rounds you have left to recover from your possible loss. But again, this is the same in real bidding situations.

Game Termination

The game can be terminated at any time. The major reasons to terminate a game are:

1. Enough information is collected.
2. There is no significant difference with the previous round: it is not desirable to continue.
3. The game runs into an unexpected situation.

Information Exchange

After an initial information meeting, the information exchange will mainly be performed via email. Only in case of emergency (for example the game takes an unexpected or unfavourable turn), another meeting will be arranged.

The instructor

After each round, you will receive the following information from the instructor:

- If you are the winner of the round or not.
- If the game continues, or if it was the final round.

The bidder

You, as a bidder, should provide the following information to the instructor:

- Your bid.
- A short motivation of your strategy for a specific bid: a couple of sentences would be sufficient.

The second issue, a short motivation, is really important in order to understand your bidding behaviour even better. This motivation will be anonymous, which means that your name will not be mentioned in the results of this experiment.

Appendix B

Player's motivations

Player 1:

Round	Motivation
1	"I want to know if there is any other player bidding exactly or below actual price. I don't dare to underbid because I don't know the reaction from Jacco about some hidden rules. If i bid around 115 I probably get no additional information, which I hope to get with this bid. "
2	"Well, now I know that at least one person did an underbid (or even bid). This will be a kind of ""tragedy of the commons"", ie everyone will lose (not get the project or do an underbid). My only logical conclusion is to go back to original conditions and see what will happen. My new bid 109 (I forgot if we were expected to have 10 or 15 % percent profit)."
3	"Well everyone must soon understand that the only way to get around 15 % profit is not to underbid. I will keep on this level for the rest of the game."
4	"Well everyone must soon understand that the only way to get around 15 % profit is not to underbid. I will keep on this level for the rest of the game."
5	"Well everyone must soon understand that the only way to get around 15 % profit is not to underbid. I will keep on this level for the rest of the game"
6	"Well everyone must soon understand that the only way to get around 15 % profit is not to underbid. I will keep on this level for the rest of the game."
7	"I'm very pleased because now we have a cooperative community. My new bid 115. (If you not cheat all of us in the name of an experimental purpose :-)"
8	"No need for changing a winning concept (but I think you fool all of us at the moment :-)."
9	"I bet you are joking with us = all is winning just to see the reaction. This is a joke back!!! "

Player 2:

Round	Motivation
1	"The first interaction with a new customer it is important to be reasonable cheep. But as well not take a too big chance. My initial aim is to break even, with just a small profit, which is needed to finance costs connected to the bidding process and interest from the initiation of work to the first payment from the customer."
2	"Since the last bid round did not go that well, I am aiming at winning a bid and become one of the co-operators for the customer. Have to win a bet to be able to do this."
3	"I still have to lie a little low, to get a track reckord going with the customer. But if I stay to low for a number of bids, I will end upp loosing money. The climb towards break even must start right now."
4	"If I win this bid, I will be on ""break even"", my earnings will be 100 units, per bid. So thats what I am aiming for after this round. Since I have had 2 successful bids lately, I hope that the buyer have gained som possitive track record with me, even though I this time meight not be the cheapest bid. My next goal will be to start

	earning money and aim for the profit margin of 15 units."
5	"After the successful round four I am at Break even, the earned average is now 100 points. This round I would need a bid on 160 to reach my goal, but this would probably not give me a win in this round. So I will settle for 135, and hope that my track record with the customer will give me the project, even though I am probably not cheapest any more. A successful round would give me an average of 109 points."
6	"I assume since I did not get the bid the last time, I have to lower my price. But it is hard for me to possibly gain the profit desired if I go lower than 125, I was costly to win the initial bids, now I have to compensate for the first bids, by having a higher price."
7	"I assume since I did not get the bid the last time, I have to lower my price. But it is hard for me to possibly gain the profit desired if I go lower than 125, I was costly to win the initial bids, now I have to compensate for the first bids, by having a higher price."
8	"Since I did not win the last bid, I lowered my profit further. This is troublesome since this will limit my probability to gain the necessary profit margin in time."
9	"My guess is that some one have put their limit on 115 as was the goal, or that it is just a little bit above that figure. So my new bid is a bit below this, just so I can get back in the game."

Player 3:

Round	Motivation
1	"Since I am new and I have to conquer the market I bid 75. With the excellent results of my team I can show I deliver quality although this is not important for the project according to your definition. However I need to get a good name in the industry."
2	"Still not making profit but perhaps my project is a good reason for him to take my second project as well."
3	"I cannot afford to lose too much money. The customer is hopefully pleased with my first project"
4	"I cannot make too much loss so it must stay at 90."
5	"I will bid 70 now. I have not a good overview of the prices that other people bid. In reality I should quit now since I am afraid I can never make profit for these prices I offer."
6	-
7	-
8	-
9	-

Player 4:

Round	Motivation
1	"99: It has to cost to establish a relationship with the customer, this bid will (if successful), be increased in future games."
2	"115: Several bidders try to underbid. In such a situation, the best thing to do is not to take the bid."
3	"to average on 115, the previous winners must win on higher bids than 115. From

	that perspective 115 is a good competitor."
4	"115: Same as previous"
5	"98: From time to time one must try to get a project (and hope for the customer to stick to me long enough for me to gain it in later projects)."
6	"120: It is time to get economy in the production."
7	"120 I will have to make another try to gain the loss of my win :-)"
8	"117: I have to end on "+", so a bid over 115 is required."
9	<p>"This is the standing bid for me (which may be used in the coming rounds if I do not answer in time):</p> <p>Round $x, x \geq 8$: $112 + a$ where $a_x = a_{x-1} * 2$. If I did not win the last round, a is set to 1.</p> <p>Motivation: I am still behind, and have to get a positive result in the end. By exponentially increasing the payoff, there will be a quick payoff.</p> <p>Example: I win round 42 on 113 (= 112+1) in round 43 i bid $112 + 2*1 = 114$, which I also win, the 44th bid is therefore $112 + 2*2 = 116$, which I also win, the 45th bid is therefore $112 + 2*4 = 120$, etc...</p> <p>I will run this manually when I am at place, but if I am not answering, you may without asking use the revealed strategy above. "</p>