

# Applying Game Theory to the Incremental Funding Method in Software Projects

Eduardo Mattos, Marcelo Vieira, Eber Assis Schmitz, Antonio Juarez Alencar

Email: {eduardocm,mvmonteiro}@ufrj.br, {eber,juarez}@nce.ufrj.br

**Abstract**—Game theory has been successfully used to analyze situations in several areas, such as economics, politics, sociology and biology. The Incremental Funding Method (IFM) is a well known technique for optimizing the financial return of software projects under monopolistic conditions. This paper presents a new approach for the maximization of software projects' financial returns under duopolistic situations, based on the application of game theory concepts to the IFM. It provides decision makers with policies, which demonstrate how and when the product, divided into modules, should be developed and launched in order to maximize return on investment of a project.

**Index Terms**—Game theory, Incremental Funding Method, value-based software engineering.

## I. INTRODUCTION

Game theory is a mathematical modeling technique based on strategic interactions of cooperation and competition. Since the publication of “The Theory of Games and Economic Behavior” by [1], game theory has been applied to diverse fields ranging from war conflicts to economics and animal behavior.

Recent studies applied game theory concepts in order to better understand and improve the software development process. In order to calculate the payoffs of software development “games”, Grechanik [2] and Yilmaz [3], focused on how team restructuring can improve project performance indicators, taking into account elements such as personal preferences and self motivation. Sazawal [4], noticing that software engineering teams lacked a simpler and more straightforward development method, proposed a lightweight approach to software development using game theory.

The Incremental Funding Methodology (IFM) is a collection of software development concepts, created by Denne and Cleland-Huang [5], intended to support software development planning in order to maximize the project value. The essence of IFM is the decomposition of the software product into a set of modules called Minimum Marketable Features (MMFs), which have the ability to generate value to the business immediately after their deployment. The IFM also focuses on other aspects such as: MMF identification, incremental architecture definition and the evaluation of intangibles. It is important

to notice that the maximization of the financial return of a project, is a well known problem in operations research called the *maxNPV problem* [6].

Although Alencar et al. [7] highlighted the merits and pitfalls of the IFM method, their work did not touch one very important limitation – the IFM financial optimization techniques are applicable only to non-competitive situations. In other words, it can be applied only to monopolistic environments. This paper presents a method that improves upon IFM using Game Theory concepts and provides decision makers with tools to generate optimum software product development plans in the presence of competition. The method can be used to analyze real situations and bridge the gap between financial and IT managers, precisely because it enables the definition of optimal software development strategies under more realistic competition conditions.

This paper is organized as follows: Section 2 describes the conceptual framework, presenting an overview of the IFM and Game Theory concepts used in our analysis. Section 3 introduces the application of Game Theory to the IFM framework, taking into consideration external competition. Sections 4 applies our method to a real-world inspired scenario. Finally, Section 5 presents our conclusions and discusses some ideas for research.

## II. CONCEPTUAL FRAMEWORK

### A. The Incremental Funding Methodology

MMF are component modules of a software product that have a well defined marketable value, adding value by facilitating customer's interaction with the company. An example of an MMF is a flight reservation module embedded in an airline company's website. Once deployed, an MMF generates a stream of revenues to the project and like any other product development, its development is associated with a cost consisting of the sum of resources needed for its creation. Finally, in order to be completely operational, an MMF may require the presence of additional MMFs, which creates some constraints for its deployment.

We will present the above concepts based on an example by Denne [5, p. 67]. In this example, the

software product  $SP$  is composed of 5 MMFs:  $SP = \{M_1, M_2, M_3, M_4, M_5\}$ . The precedence relation between MMFs is shown in Figure 1 and the cash flow associated with each MMF is shown in Table I

TABLE I.  
MMF CASH FLOW

MMFs	1	2	3	4	5	6	7	8
$M_1$	-100	80	81	82	83	84	85	86
$M_2$	-50	100	102	102	106	108	110	112
$M_3$	-200	75	76	77	78	79	80	81
$M_4$	-75	100	102	102	106	108	110	112
$M_5$	-100	60	61	62	63	64	65	66

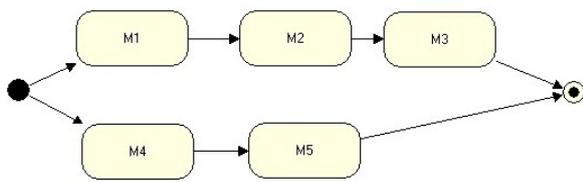


Figure 1. MMFs Precedence Graph

TABLE II.  
NPV OF FEASIBLE SEQUENCES

Code	Sequence	NPV
S1	$M_1 M_2 M_3 M_4 M_5$	1421
S2	$M_1 M_2 M_4 M_3 M_5$	1450
S3	$M_1 M_2 M_4 M_5 M_3$	1439
S4	$M_1 M_4 M_2 M_3 M_5$	1449
S5	$M_1 M_4 M_2 M_5 M_3$	1438
S6	$M_1 M_4 M_5 M_2 M_3$	1397
S7	$M_4 M_1 M_2 M_3 M_5$	1474
S8	$M_4 M_1 M_2 M_5 M_3$	1463
S9	$M_4 M_1 M_5 M_2 M_3$	1422
S10	$M_4 M_5 M_1 M_2 M_3$	1404

In this example, by using a interest rate of 2.41% per period, the optimal implementation order is  $M_4 M_1 M_2 M_3 M_5$ , which will provide a return, given by its Net Present Value (NPV), of \$1474.00.

**B. Optimal implementation order using IFM method**

The IFM method for defining the deployment order of the product is comprised of 3 basic steps. The first one is the partition of the product into a set of MMFs. The second step of the IFM requires the estimation of the cash flow stream associated with each MMF. Although the actual techniques used to produce this result are not discussed in the IFM proposal, there is a considerable amount of literature available on the subject.

The final step is the evaluation of the optimal implementation sequence. The challenge of finding the optimal

sequence is classified as NP-hard problem, due to the fact that the optimal implementation order is a combinatorial optimization problem [5], whose solution requires the evaluation of the present value of all feasible sequences. A feasible sequence is an ordered sequence of MMFs that respects the precedence relation between MMFs. As part of their work, Denne and Cleland-Huang suggest several heuristics that can be used for finding approximations to the optimal solutions [5, p. 67].

The IFM method can now be formally stated as:

**Definition 1: MMF set**

$SP = \{M_1, M_2, \dots, M_m\}$  is the set of all MMFs that compose the product  $P$ .

**Definition 2: Life cycle**

is the number of periods ( $n$ ) of the product life. Once deployed, MMFs will generate cash flow streams until the end of the project life.

**Definition 3: Cash flow stream (CFS)**

is a time series of values  $\{C_{i,1}, C_{i,2}, \dots, C_{i,n}\}$ , where  $C_{i,k}$  represents costs or revenues generated by  $M_i$  at period  $k$ .

**Definition 4: Precedence constraint**

there exists a precedence constraint between  $M_i$  and  $M_j$ , when  $M_j$  can only be deployed after  $M_i$  deployment. The set of precedence constraints between a set of MMFs, can be defined as a relation containing all 2-ples  $(m_i, m_j)$  such that  $m_i$  precedes  $m_j$ . This relation can also be depicted graphically as a acyclic directed graph as shown in Figure 1.

**Definition 5: Net Present Value**

If  $r$  is the interest rate used to discount all monetary values, then the present value of  $MMF_i$  deployed at period  $s$ ,  $V(i, s)$ , can be expressed as:

$$V(i, s) = \sum_{i=s}^n \frac{C_{i,(i-s+1)}}{(1+r)^i}$$

**Definition 6: Implementation sequence**

The implementation sequence  $S$ , is an ordered set of integer values  $1 \leq j \leq m$  that expresses the implementation order of the MMFs of  $SP$ . For example, the sequence  $S = \{3, 1, 2\}$  indicates the following implementation order:  $M_3$  is deployed at period 1,  $M_1$  at period 2, e  $M_2$  at period 3.

The problem of finding the optimal deployment sequence of a project can be presented as a combinatorial problem: Find an order  $o_i$  for each  $M_i \in SP$  that maximizes:

$$SNPV(S) = \sum_{i=1}^m V(i, o_i)$$

In conclusion, given the product, the set of MMFs that compose it, the precedence relation between the MMFs, and the cash flows associated with each one, the

decision problem of defining the optimal implementation sequence, under monopolistic conditions, turns out to be an easy (although resource consuming in some cases) computational problem.

*C. Fundamentals of Game Theory*

Game Theory is a method of studying strategic decision making through mathematical models of conflict and cooperation among self-interested rational agents. In other words, it predicts how others will respond your actions. Moreover, one’s decision is linked to their competitors’ reaction, with each seeking the best outcome for themselves in a puzzling relationship of interdependence strategies. Graham Romp (1997) describes it as mutual interdependence, in which the welfare of any one individual in a game is, at least partially, determined by other players’ actions. Therefore, individuals are encouraged to act strategically in order to achieve the most desirable outcome.

Game theory provides a powerful toolbox with methodologies capable of organizing reasoning. Along with other traditional decision models and economics concepts, it has been used extensively to model competing behaviors of interacting agents. Such models allow strategic thinking and support the dynamic maneuvers against rivals, while consequently promoting competitive advantages.

A game theory model is basically built around the strategic choices available to players, in which the preferred outcomes are defined and known. Determining the best moves, however, depends on the context, or essence of the game, which may involve cooperation or competition. When cooperation exists, players can assume binding and enforceable commitments with one another, so that real intentions – agreements, promises and threats – of each individual are flagged and formalized. Each player takes advantage of these agreements, thus characterizing a cooperative game. By contrast, in a non-cooperative game, players need to conceal their strategies from each other, since coalitions are absent and a common goal is not shared. Due to this assumption, non-cooperative games are individualistic by nature, resembling the real world environment of competition between companies. This is a primary motivation for this study.

Game theory differentiates between two types of games: strategic games and extensive games [8]. The difference between the two games stems from the information set available to each player when they make a decision, as well as the timing of decisions, and finally, how often the game is played.

A strategic game is a model of interactive decision-making in which each player selects his plan of action once and for all, and these choices are made simultaneously [8]. The decisions are made in isolation, that is

independent of other players. The players do not have information about which strategy has been adopted by their opponent. Also known as simultaneous games due to their static nature, strategic games are inherently games of imperfect information. In addition, these games are predominantly represented in normal (matrix) form, since the amount of information available to players is constant within a game (no information is obtained during its course), and timing of decisions has no effect on the players choices. A common interpretation of a strategic game is that it models a unique event, also called single-stage, one-off or non-repeated game. An extensive game, on the other hand, is dynamic in nature, as time plays a critical role in the outcome of the game. Players make decisions sequentially while observing the precedent before pursuing an action. This sequential game is usually represented by decision trees (an extensive form game), and players get additional information throughout the game.

One example of a widely discussed strategic game is called the “prisoner’s dilemma”, originally framed by Merrill Flood and Melvin Dresher while working at RAND in 1950. Albert W. Tucker formalized the game with prison sentence rewards and coined its the name: “prisoner’s dilemma” [9]. It’s presented as follows:

Two members of a criminal gang are arrested and imprisoned and placed in solitary confinement without any means of communicating with each other. Since the police doesn’t have enough evidence to convict them on the principal charge, the maximum they can get is sentence both to a year in prison for carrying a gun. The officer makes the same offer to each prisoner: you go free if you testify against his partner, who will then get three years in prison on the main charge. But, if they both testify against each other, both will get two years in jail. Each prisoner can take two actions: (Denounce, Keep Mum).

The game can be shown in a tabular form in Table III, the number of years in jail, are shown as negative of the sentences being imposed. In each cell, the first number is the outcome from the first prisoner followed by the payoff of the second. For example, if both denounce each other, the outcome will be (-2) for each one of them.

TABLE III.  
PRISONER’S DILEMMA

Players 1/2	Denounce	Keep Mum
Denounce	-2,-2	0,-3
Keep Mum	-3,0	-1,-1

Assuming that both prisoners are perfectly rational,

each one will reason as follows: my partner has only two choices, if he chooses to cooperate with the police and give me up, I will be better off cooperating with the police too (denouncing him); on the other hand, if he chooses to keep quiet, I will be better off by cooperating with the police (denouncing him). So, my best decision, independently of what he does is to cooperate with the police (denounce him). Since the same chain of reasoning occurs with the second prisoner, the predicted behavior (solution) to the dilemma is both prisoners denouncing each other and spending 2 years in jail.

1) *Strategic Games*: A formal model of an strategic game contains the basic elements as follows [10]:

**Definition 7: Players**

$N = \{P_1, P_2, \dots, P_n\}$  where  $N$  is a finite set of players. The players in a game are decision-maker agents whose behaviors dictate their choices. In order to establish interaction between strategies, or mutual interdependence, at least two players are required to compete in a game.

**Definition 8: Pure strategies**

$S_i = \{s_1, s_2, \dots, s_n\}$  is the finite set of pure strategies for player  $i$ . A strategy is an entire description of how a player could behave in response to what can be observed about its opponents' actions during a game. A game is considered symmetric when the set of pure strategies is identical for all players, and the payoff to playing a given strategy depends only on the strategies being played, not on who plays them.

**Definition 9: Mixed strategies**

$S'_i = \prod(S_i)$  be the finite set of mixed strategies for player  $i$ . If  $\prod(X)$  be the set of all probability functions over a set  $X$ , then  $S'_i$  is the set of mixed strategies for player  $i$ .

**Definition 10: Strategy profile**

$S_p = \{S_1 \times S_2 \times \dots \times S_n \text{ be } \}$  the set of strategy profiles. The set  $S_p$  of strategy profiles is the Cartesian product of the  $S'_i$ s, consisting of every possible game combination.

**Definition 11: Utility function**

$u_i : S_p \rightarrow \mathbb{R}$  is the utility function for player  $i$ . An utility function represents a player's preference relation over each strategy profile. Given that this study involves monetary rewards as outcomes, e.g. profits, a payoff function is suitable to evaluate them. In general, rational players are induced and stimulated to prefer higher payoffs, and naturally choose a strategy profile that promotes its maximization.

2) *Solution of Strategic Games*: A solution is a systematic description of the outcomes that may emerge in a game [8]. The major technique for solving a strategic game relies on the concept of the *Nash equilibrium*.

This equilibrium allows us to deal with games that cannot be solved using the concept of dominance [11]. In certain types of games the strict dominance concept may not predict a unique solution when strictly dominated strategies are non-existent, and thus iterated elimination cannot be employed. On the other hand, if both players have a strictly dominant strategy, the game has only one unique Nash equilibrium. For this reason, the Nash equilibrium is a broader and stronger approach to solving a game, and captures the notion of a steady state.

Finally, the Nash equilibrium solution method can now be formally stated, by the two definitions that follows:

**Definition 12: Best response**

Player  $i$  best response to strategy profile  $s_{-i}$  is a mixed strategy  $s_i^* \in S_i$  such that  $u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}) \forall$  strategies  $s_i \in S_i$ .

**Definition 13: Nash equilibrium**

A strategic profile  $s = \{s_1, s_2, \dots, s_n\}$  is a Nash equilibrium if, for all players  $i$ ,  $s_i$  is a best response to  $s_{-i}$ .

In non-cooperative games, a Nash equilibrium occurs when none of the players has an incentive to deviate from the predicted solution, because each strategy in the equilibrium is a best possible response to all another strategies from the opponents, and this is true for all players. An important theorem by Nash [12], showed that, in a finite game, composed of a finite number of players and strategies, there always exists at least one Nash equilibrium, either pure or mixed.

In a mixed Nash equilibrium, each of the chosen pure strategies has the same expected payoff value, leaving each player indifferent as to which strategy will actually be played. A mixed strategy Nash equilibrium is, therefore, said to be a weak equilibrium, because none of the players are worse off if they abandon their mixed strategy, and play any one of the pure strategy components of their mixed strategy. It can be shown that in the case of symmetric games, the equilibria are also symmetric, that is, if  $(a_i, a_j)$  is a NE, then so is  $(a_j, a_i)$  [12].

One of the puzzling results of the equilibrium solution concept is that an equilibria situation does not imply the best possible outcome for the players. For example, one of the most known games, the prisoner's dilemma, shows a situation where the equilibrium solution gives both players payoffs that are less than the optimal. This outcome is also known as a non-Pareto optimal equilibrium.

### III. APPLYING GAME THEORY CONCEPTS TO IFM

We start with the assumption that we have two competing companies (players) planning the launch of similar software products in the market. The final products will contain the same set of features, each one described by a MMF. During the planning phase, a strategic plan should

be drawn to decide which release sequence of MMFs will be adopted and followed, in order to achieve maximum ROI.

A software development project can be seen as a unique event, with well-defined start and end dates. Although, the span of a software development life cycle increases with the products size and complexity, the development of a single product development can be seen as an strategic action undertaken by the company. The situation where two companies undertake the development of very similar products can then, be modeled by an strategic game, where the actions are the release ordering of the products and the payoffs are the NPVs generated by the products sales. It is a fact that the two competing companies are rational, choosing their actions simultaneously and independently. Under this interpretation, each company is unaware of the choices being made by its competitors when choosing a strategy.

The next paragraphs describe the steps needed for the proposed framework.

The first step is to decide which MMFs will be part of the two competing software packages that both companies wish to develop. Afterwards, we have to evaluate the cash-flows generated by each MMF in a monopolistic situation. As stated previously, this can be done using different tools or methods, but this data is necessary to calculate the NPV of all possible implementation orders.

The second step is to include a model of the effects of competition on the estimated monopolistic cash flows. When two companies wish to enter the market with similar products, they will have different ROIs than if they were the lone player. Two well known models can be used: Cournot-Nash and Stackelberg [11]. In the first model, companies compete on quantities and in the second on price. The application of game theory concepts to these models predict the equilibrium quantities, prices and profits that can be gained by the competing companies. Any of these methods could be applied to evaluate the effects of competition on ROI values.

The third step is to define how the companies will organize themselves in order to get the maximum return in the case of a market dispute. Both can ex-ante define a MMF launching order or, conversely, they could change the implementation order as the market dispute game proceeds. Each situation can be modeled as a different type of game – the first can be treated as a strategic game and the second, which will not be discussed in this paper, as a sequential game. In the former case, both companies will compromise on the delivery sequence and stick this strategy to the end of the product life cycle. The standard solution technique is to decide the implementation order by examining the Nash equilibria of the game.

When constructing a model of the strategic game, each of the implementation sequences will be mapped

to one action for every one of the players. The payoffs will be the ROI values each of the players will receive under the market model being used. This will result in a strategic game model, represented in normal form by a matrix where each dimension contains all possible implementation orders available to the players. If we assume the same finite set of actions for both players, the resulting game will be a symmetric game. The solutions can be generated by feeding the final NPV value of each implementation sequence of both players as an input for the game solver software like Gambit [13], which uses sophisticated algorithms to calculate the Nash equilibria for the game.

#### IV. AN APPLICATION EXAMPLE

We will apply our method to a scenario where two companies, ERP Co. and MPR Ltd., who produce Enterprise Resource Planning software, are planning the launch of new software packages with the same set of features. They wish to obtain the best possible ROI by dividing the roll out plan for their software into well defined MMFs. Since each company has an accurate estimate of the others' capabilities and resources required to create the product that will satisfy the market's current needs and expectations, they arrive at the same estimates of costs and cash flows associated with each MMF.

As this is a strictly non-cooperative game, ERP Co. and MPR Ltd. are self-interested companies fighting for their own benefit. A common practice is to hide sensitive information – firms protect their commercial secrets so as to avoid losing competitive advantages and resources. Beyond that, governments strive to increase competition in oligopolistic markets and enforce antitrust laws to prohibit collusion. For such reasons, the available information is limited: (1) each firm only knows which are the MMFs due to market research, (2) market research will also provide good estimates of the sales potential for each software feature and (3) development durations and costs can be derived from a more detailed specification for each MMF [14].

When referring to the competing companies' actions being chosen simultaneously, it does not necessarily mean that these actions are taken at the same point in time. Rather, it portrays these decisions as if they were made at the same time. Thus, the timing of launching a MMF has no effect on companies future choices. Each company is committed to following the selected strategy with no deviation during the course of the strategic game and both parties will fulfill the release sequence plan of MMFs.

The next step defines the rules that simulate the free market behavior in the face of competition. In this paper will use a very simple rule, the 50/50 rule, which states that whenever a player launches an MMF in the market,

the cash-flow it generates is divided by half in the period following the competitors launch of a similar MMF.

We will assume that both companies will use the same MMFs ( $M_1, M_2, M_3, M_4$  and  $M_5$ ), with the precedence graph as shown in Figure 1 and the same period adjusted cash flows as portrayed in Table I, one can evaluate all possible implementation scenarios for this game. This results in 100 possible action profiles depicted in Table IV below.

The strategic game model displayed below in Table V represents all possible interactions between ERP Co. and MRP Ltd. in a situation of conflict. We define ERP Co. as the row player and MRP Ltd. as the column player. Although in this case, due to the symmetry of the game, there is no impact regarding any future analysis if we switched their positions. So, when both companies simultaneously choose an available strategy, they receive a corresponding payoff shown in each strategy profile from the given matrix, where the first number denotes the payoff of the row player, and the second that of the column player.

Gambit, a popular tool for solving finite games, was used to compute all Nash equilibria. The tool found four pure strategies, shown in Table VI. Since this is a symmetric game, the four equilibria correspond to only two symmetric strategic profiles: (S1,S10),(S10,S1) and (S7,S3),(S3,S7).

TABLE VI.  
REDUCED PAYOFF TABLE SHOWING NE

ERP/MRP	S1	S3	S7	S10
S1	466,466	534,550	585,637	<b>703,685</b>
S3	550,534	478,478	<b>589,623</b>	646,610
S7	637,585	<b>623,589</b>	494,494	610,543
S10	<b>685,703</b>	610,646	543,610	459,459

#### A. Solutions to the Strategic MMF Development Game

The solution to a game is a recommendation for the players on how possible outcomes can be achieved. Since this is a strictly non-cooperative game with multiple solutions, in particular, it is expected that the players will act rationally and choose a strategy that will maximize their returns. The difficulty arises in predicting which one of the possible equilibria will be adopted by both players.

Table VI shows that the first two solutions are symmetric Nash equilibria VI. The first pair of NE involves the action profiles (S1,S10) and (S10,s1) and yields the highest revenues. When Player 1 plays S1 and Player 2 plays S10 their payoffs will be (703,685) respectively. If they reverse their actions the payoffs will also be reversed (685,703). The second pair of NE contains the action profiles (S3,S7) and (S7,S3) which return values of (589,623) and (623,589) respectively.

In order to find which equilibrium will be played, one can search for a Pareto-optimal equilibrium. In this case, the first NE (703,685) dominates the second. That is to say, the lowest return that can be obtained in the first (685) is greater than the maximum of the second NE (623). Thus, rational players should strive to achieve the first NE. In this case, they could compete with actions S1 and S10. But, if they simultaneously try to achieve the maximum value, by playing (S1,S1) they will both be worse-off and receive (466,466). Similarly, if they play (S10,S10) the payoff will be (459,459).

This competitive situation turns out to be an example of coordination – in order to optimize their returns the players must coordinate to play different sequences. The reduced model shown in Table VI is a version of the classic game called “Battle of the Sexes” [15, p. 22]. The only solution to this game, given the impossibility of coordination, consists of a mixed strategy where both players randomize over their actions. Although what seems a sensible solution for the “Battle of the Sexes” game, can hardly be applied to a real business situation [16]. This is simply due to fact that a mixed strategy may result in lower returns than using a pure one. So, one can disregard this type of solution in the current context.

Another solution to this coordination problem is for one of the players to publicly announce their implementation strategy (the optimal one) thus leaving the second best sequence as the only option for the other player to choose. For example, assume that one of the players (ERP Co.) makes the first move, announcing the development its first MMF. As the first mover, it will certainly try to attain to the optimum ROI path. Therefore, MRP Ltd. will be forced to choose a different implementation sequence in order to avoid dividing its revenues up with ERP Co. This results in payoffs of 703 and 685 for ERP Co. and MRP Ltd. respectively. In this case, the best strategy is for the leader to select the optimum implementation order. This strategy forces its competitor to develop his product in a different sequence, while pursuing the second best implementation order.

These results concur with the Stackelberg model of oligopolistic competition with two players. In this model, the player who makes the first move is dubbed the market leader, while the follower, observes the leader and reacts to the leaders actions. When two identical players move sequentially in a game, the player that moves first – the Stackelberg leader – earns higher profits than the player that moves second – the follower [17, p. 649]. This is called the *first-mover advantage*, and [18] explains this situation, using as an example, the advantage of product placement:

One way in which the first-mover advantage is manifested is in the first-mover’s preemption

TABLE IV.  
GAME EXAMPLE, WITH READJUSTED CASH FLOWS, SHOWING THE EFFECTS OF THE 50/50 RULE

ERP	1	2	3	4	5	6	7	8	Total
$M_1$	-100	80	81	41	41.5	42	42.5	43	
$M_2$	0	-50	100	102	52	53	54	55	
$M_3$	0	0	-200	75	76	38.5	39	39.5	
$M_4$	0	0	0	-75	50	51	52	53	
$M_5$	0	0	0	0	-100	30	30.5	31	
CF	-100	30	-19	143	119.5	214.5	218	221.5	827.5
								NPV	703

MRP	1	2	3	4	5	6	7	8	Total
$M_4$	-75	100	102	104	53	54	55	56	
$M_5$	0	-100	60	61	62	31.5	32	32.5	
$M_1$	0	0	-100	40	40.5	41	41.5	42	
$M_2$	0	0	0	-50	50	51	52	53	
$M_3$	0	0	0	0	-200	37.5	38	38.5	
CF	-75	0	62	155	5.5	215	218.5	222	803
								NPV	685

TABLE V.  
MATRIX WITH ALL POSSIBLE GAMES BETWEEN THE COMPETING COMPANIES

ERP/MRP	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	466,466	500,528	534,550	547,574	581,597	627,603	585,637	619,659	665,666	703,685
S2	528,500	482,482	515,504	529,528	562,550	557,608	567,589	612,600	646,618	684,638
S3	550,534	504,515	478,478	551,561	524,524	571,530	589,623	562,585	609,591	646,610
S4	574,547	528,529	561,551	482,482	515,504	560,510	520,542	553,564	598,571	636,590
S5	597,581	550,562	524,524	504,515	477,477	523,483	542,575	515,537	561,544	598,463
S6	603,627	557,608	530,571	510,560	483,523	456,456	549,621	521,583	494,516	532,535
S7	637,585	589,567	623,589	542,520	575,542	621,549	494,494	528,515	573,523	610,543,
S8	659,619	612,600	585,562	564,553	537,515	583,521	515,528	490,490	535,496	572,515
S9	666,665	618,646	591,609	571,598	544,561	516,494	523,573	496,535	469,469	506,487
S10	685,703	638,684	610,646	590,636	463,598	535,532	543,610	515,572	487,506	459,459

of the ‘best’ locations in attribute space. This preemption is effective because it is costly for the incumbent to change its product locations. (Presumably such change will involve repositioning the products or, worse, manufacturing a different product line.) In other words, the second mover must realize that it has to take the incumbent’s products as given – that nothing it can do will change them. (p. 269)

*B. Comments on the Solution*

A real world example of such competition can be the SAP case, which is one of the few first-movers in IT business to succeed and maintain its first mover advantage to date. On the other hand, many real life examples have shown that a second mover advantage is the most common outcome, as they are capable of observing the first mover flaws and take advantage of a scenario they hadn’t previewed when they first developed their product or service [19].

The results shown here apply to the case of two rational players competing for the highest payoff. However, a complete different scenario would arise if the objective

was to maximize the competitor’s losses. For example, if MRP Ltd. had the intention to hurt ERP Co.’s payoffs, even by sacrificing its own gains, it could choose to implement the same optimum path as ERP Co..Both will end with a payoff of 466, much inferior to the highest possibility in this game (Table VII).

TABLE VII.  
EFFECTS OF IMPLEMENTING THE SAME ORDER

	1	2
Sequence	12345	12345
Revenue	570	570
NPV	466	466

V. FINAL CONSIDERATIONS

A. Conclusions

The IFM presented a new approach to a well known software engineering problem: how to partition, develop and deploy a software product. As part of this method, software modules are thought to be value generators and their successful exploitation of this characteristic can

bring about substantial financial benefits to the developing company.

One of the main weaknesses of the IFM is that it ignores the effects of competition. Monopolistic competition is rarely found in modern business environments and even less so when one considers the information technology industry, in which most technologies are freely available.

The use of the IFM in a competitive environment requires that the business strategists take into account the actions of rival companies operating in the same market. Game theory provides a solid foundation for modeling and analysis of such competitive situations.

The IFM procedure for maximizing financial return on IT projects must be adapted to fit this new situation. The initial steps are still preserved: identification of MMFs, estimation of its monopolistic financial returns and identification of the optimal deployment orders.

The introduction of competition requires some modification to the process. First, one must estimate the market rules to be applied, a process similar to that used in economic models for competition such as Cournot, Bertrand and Stackelberg. At the conclusion of the modeling step, the situation can be seen as a strategic game between two players and the results can be obtained by finding the games solutions. A solution is a suggestion for the players on how to act in any situation. The most widely known solution technique is the Nash equilibrium, which contains a set of each players actions and is considered the best response to its rivals optimal response.

The results of our example show that in a competitive scenario the equilibrium solutions do allow both competitors to select the optimal monopolistic implementation order. Since one of the players will be better off accepting a sub-optimal sequence, a problem of coordination appears. The way around this problem is for one of the players to announce its strategy in advance. If both players ignore the competition, they could both potentially choose to cheat and reach sub-optimal results. Such a result could also occur if one of the players aims is to cause maximum loss to its competitor.

It is important to notice that obtaining the necessary information may not be a simple task. Companies need to count on powerful business analytical tools to support the estimation and forecasting of personal and competitors outcomes, supplying the input values for IFM simulations. These values are the building blocks for both strategic and extensive game models. Business intelligence plays a fundamental role in enabling the employability of the game theory approach. An earlier and precise understanding of market behavior can enhance the predictions and the choice of an optimal strategy to follow, evidencing the fact that a reliable strategic model depends on the quality of knowledge produced by the analytical tools used.

## B. Future Work

Imperfect information games – these types of games would be closer to a realistic scenario, where companies don't have all of the information available before the game begins. In this case, we would also need a more robust tool to simulate several games based on informed predictions and best guesses. After the game begins, it is possible to gather market information about your competitors choices and their products for a later comparison with the game profiles you have previously simulated. To find a best fit, one should base his developments on the Nash Equilibrium – the most probable outcome – of the best-fit game. Additionally, if the game follows a path that differs from the predicted one, it would simply need a second adjustment of the game strategy, in accordance with the companys objectives and possibilities.

## REFERENCES

- [1] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton University Press, 1944. [Online]. Available: <http://jmvidal.cse.sc.edu/library/neumann44a.pdf>
- [2] M. Grechanik and D. Perry, "Analyzing software development as a noncooperative game," *IEE Seminar Digests*, vol. 29, 2004.
- [3] M. Yilmaz and R. V. O'Connor, "A software process engineering approach to improving software team productivity using socioeconomic mechanism design," *SIGSOFT Softw. Eng. Notes*, vol. 36, no. 5, pp. 1–5, Sept. 2011. [Online]. Available: <http://doi.acm.org/10.1145/2020976.2020998>
- [4] V. Sazawal and N. Sudan, "Modeling software evolution with game theory," in *Trustworthy Software Development Processes*, ser. Lecture Notes in Computer Science, Q. Wang, V. Garousi, R. Madachy, and D. Pfahl, Eds. Springer Berlin / Heidelberg, pp. 354–365.
- [5] M. Denne and J. Cleland-Huang, *Software by Numbers: Low-Risk, High-Return Development*. Prentice Hall PTR, 2003.
- [6] W.S.Herroelen and E.L.Demeulemeester, "Resource-constrained project scheduling: a survey of recent developments," *Computers and Operations Research*, vol. 25, pp. 279–302, 1998.
- [7] A. J. Alencar, J. V. D. Jr., E. A. Schmitz, and A. L. Correa, "On the merits and pitfalls of the incremental funding method and its software project scheduling algorithms," *Communications in Computer and Information Science*, vol. 292, pp. 493–502, 2012.
- [8] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. The MIT Press, 1994.
- [9] W. Poundstone, *Prisoner's dilemma*. Anchor Books, 1992.
- [10] S. Y. Leyton-Brown, K., *Essentials of game theory*. Morgan and Claypool, 2008.
- [11] K. Dutta, Prajick, *Strategies and games*. MIT Press, 1999.
- [12] J. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951.
- [13] R. D. McKelvey, A. M. McLennan, and T. L. Turocy, "Gambit: Software tools for game theory," 2010, version 0.2010.09.01. [Online]. Available: <http://www.gambit-project.org>

- [14] R. Pressman, *Software engineering - a practioners approach*. McGraw-Hill, 2010.
- [15] G. Romp, *Game Theory : Introduction and Applications*. Oxford University Press Inc., New York, 1997.
- [16] H. Levy, *Stochastic dominance: investment decision making under uncertainty*. Springer, 2010.
- [17] E. Gal-Or, "First mover and second mover advantages," *International Economic Review*, vol. 26, no. 3, pp. 649–653, 10 1985. [Online]. Available: <http://www.jstor.org/stable/2526710>
- [18] K. S. Moorthy, "Using game theory to model competition," *Journal of Marketing Research*, vol. 22, no. 3, pp. 262–282, 08 1985. [Online]. Available: <http://www.jstor.org/stable/3151424>
- [19] G. J. Hidding, J. Williams, and J. J. Sviokla, "How platform leaders win," *Journal of Business Strategy*, vol. 32, pp. 29–37, 2011. [Online]. Available: <http://dx.doi.org/10.1108/02756661111109752>

**Eduardo Mattos** is an M.Sc. student with the Federal University of Rio de Janeiro (UFRJ), Brazil. His research interests include software development methodologies and economics of software engineering.

**Marcelo Vieira** is an M.Sc. student with the Federal University of Rio de Janeiro (UFRJ), Brazil. His research interests include software development methodologies and economics of software engineering.

**Eber Assis Schmitz** is a Professor of Computer Science with the Federal University of Rio de Janeiro (UFRJ). He holds a B.Sc. in Electrical Engineering from the Federal University of Rio Grande do Sul (UFRGS), an M.Sc. in Electrical Engineering from UFRJ and a Ph.D. in Computer Science from the Imperial College of Science Technology and Medicine, England. His research interests include software development modeling tools, business process modeling and stochastic modeling.

**Antonio Juarez Alencar** is a researcher with the Federal University of Rio de Janeiro (UFRJ), Brazil. He received his B.Sc. in Mathematics and M.Sc. in System Engineering and Computer Science from UFRJ. He holds a D.Phil. in Computer Science from Oxford University, England. His research interests include economics of software engineering, IT strategy and risk analysis.