ON THE ECONOMIC RETURN OF A SOFTWARE INVESTMENT

MANAGING COST, BENEFIT AND UNCERTAINTY

Emil Numminen

Blekinge Institute of Technology
Doctoral Dissertation Series No. 2010:13

School of Management

On the Economic Return of a Software Investment
Managing Cost, Benefit and Uncertainty

Emil Numminen

Blekinge Institute of Technology Doctoral Dissertation Series
No 2010:13

On the Economic Return of a Software Investment

Managing Cost, Benefit and Uncertainty

Emil Numminen

Department of Business and Economics
School of Management
Blekinge Institute of Technology
SWEDEN

Department of Business and Economics
School of Management
Publisher: Blekinge Institute of Technology
Printed by Printfabriken, Karlskrona, Sweden 2010
ISBN: 978-91-7295-194-5
Blekinge Institute of Technology Doctoral Dissertation Series
ISSN 1653-2090
urn:nbn:se:bth-00480

Caroline and Leila: This one's for you!

"Ideas are seldom born fully clothed, but are rather dressed by a slow and arduous process of accretion" (Rubinstein, 2002, p. 1).

ABSTRACT

The purpose of this dissertation is to explore how the economic return of a software investment can be assessed and managed. This topic has been studied in research and has been a concern for firms making software investments. In order to study this we need a model of the underlying factors affecting the economic return. Assessing and managing the return of a software investment is been argued to be difficult due to specific economic characteristics of a software investment, i.e. high degree of intangible consequences and uncertainty about the total investment cost. Given these characteristics it is has been concluded that it is difficult to derive a return function.

In this dissertation we question this conclusion and propose a comprehensive model to assess and manage the intangibles and the underlying uncertainty. The model is deduced from general assumptions of the economic behavior of the firm. To develop the model we analyze the relevance of intangibles in relation to the economic purpose of making a software investment. Based on this a new way of deriving a cash flow function for a software investments is defined.

Further it is analyzed how the underlying uncertainty of a software investment can be managed. The analysis uses a quantitative approach and methods from financial economics. It includes how the application of a real option and portfolio approach can reduce the uncertainty in a software investment and the role of efficient software platforms. The relation between software platforms and the opportunity to create different types of real options for future development is inferred from empirical studies. The studies in this dissertation show how a managerial view on a software investment corresponds with the overall economic goal of the firm. They also show how a strategic value of a software investment can be created, assessed and managed.

Keywords

Economic Return, Managerial Decisions, Real Option, Software Investment, Uncertainty

ACKNOWLEDGE-MENTS

Finally a long journey comes to an end! I owe my gratitude to the below people for taking a part in it (un)voluntarily…

First and foremost I want to thank my supervisors Birger Rapp and Anders Hederstierna. Birger: Thank you for making me write those extra five sentences more or less everywhere. In hindsight you were right in the nitpicking. Anders: Thank you for persuading me to do a Ph. D. and for all the help along the way. You were also the one who hired me way back; I hope you have not regretted it…

I would also like to thank Peter Jennergren, who was the discussant at my final seminar, for making me clarify and argue for what I had taken for granted.

I have had the fortune to be associated and partly funded by the Swedish Research School of Management and Information Technology (MIT). Thank you, senior and junior, colleges for comments and constructive discussions on my material through out the years. Thank you Petter Ahlström and Pontys Fryk for adding the social side at the meetings this has been much needed after long and intensive days.

On the empirical parts of this study I would like to thank the undisclosed companies for agreeing to participate. Special thanks to Magnus, Mårten, Rehan, Patrik and Tobbe for your time and input.

I also want to thank the participants at our local seminars at the School of Management at BTH for comments and for providing an environment out of which this dissertation is derived. Especially thank you Anders Wrenne (if for nothing else for writing two of the papers with me), Bo Schenkman, Camilla Wernersson, Eva Lövstål, Henrik Sällberg, Marie Aurell, Marie Hemming, Stefan Hellmer and the ones I have forgotten at the moment. Also, thank you Ihsan Zakri for all the "fikas" and good times.

Las but not least, the greatest gratitude goes to my family for constant support and encouragement, I promise to be more present now when this is over… Thank you Caroline, Markku, Leila, Pia, Eva, Michael and Marcus with families. I would also like to thank the Lundgren family for the help and support. Michael: Thank you for proof-reading the math. Äiti: Kiitos avusta! Caroline: This would not have doable without you.

Karlskrona 2010-11-11

Emil Numminen

TABLE OF CONTENT

CHAPTER 1
SETTING

Many have argued that valuing a software investment is more difficult than valuing a traditional investment (e.g. Rehesaar and Mead, 2005; Irani, 2002; Banker et al, 1993; Ashford et al, 1988). The purpose of this dissertation is to explore how the economic return of a software investment can be assessed and managed. To support the purpose this chapter presents the research questions for this dissertation. In order to fulfill the purpose a model for assessing the return of a software investment is deduced for this study from general economic conditions. It is also discussed how the 6 papers of this dissertation relates to this model.

BACKGROUND

The focus of this dissertation is the economic return of a software investment, and how this return can be managed to create value for the firm. The economic return of a software investments stems from the difference between the economic benefit of selling or using software and the cost for developing or buying software. This residual is the economic return of a software investment. To clarify the meaning of the above, an investment is defined as "the sacrifice of an immediate and certain satisfaction in exchange for a future expectation whose security lies in the capital invested" (Massé, 1962 in Rapp, 1974 p. I-11). This definition is general enough for the purpose of this dissertation and has the same underlying meaning as definitions by e.g. Baddeley (2003). An immediate sacrifice of a certain satisfaction refers to the investments cost, which may occur over time but is valued at the present. The expected uncertain future refers to the cash flows that are expected from the software investment (Numminen, 2008a). The cash flow of an investment is a measure of the difference between the cash that is earned by making the investment and the cash that is consumed by the investment, i.e. the difference between the pure cash in and cash out of the firm due to the investment (Berk and DeMarzo, 2008). The motivation of cash flow as a relevant measure will be discussed below as the goal function for the firm is motivated when the assessment model for the dissertation is developed.

To further clarify the focus of the dissertation we define software, based on the discussion in von Eisenhardt, 2008, as a complex and recombinant system of commands and instructions for data processing which is an intangible information good that can only be consumed in discrete units. A software creates an output given a specific input and can only be used in fixed units, most often one, and is constructed using a programming language to then be compiled into ones and zeros (binary code). The different characteristics and economic implications of software are discussed more thoroughly below in Chapter 3. On an application level software investments made by firms ranges from different types of office software to advanced enterprise resource planning (ERP) systems (as studied by e.g. Aral et al., 2006), supply chain management (SCM) systems (as studied by e.g. Dehning et al., 2007) and flexible manufacturing systems (as studied by e.g. Kumar, 1995), as well as operating systems to facilitate the use of the above applications (as studied by e.g. Fichman, 2004) just to name a few of the applications that exists.

THE PROBLEM

What is the economic return of a software investment? This question has been a topic in research for several decades now but has also been an issue for the firms making these investments (e.g. Irani et al., 2007; Borenstein and Betencort, 2005; Milis and Mercken, 2004; Seafemidis and Smithson, 2000). There is still inconclusive evidence of the economic return of a software investment and therefore the question is still a topic for research after more than five decades (Shuurman et al., 2009), see e.g. Gunasekaran et al. (2006) for an extensive review of the evaluation literature concerning software investments.

There have been several aspects to this problem that has evoked the interest from various academic fields. Banker and Kauffman (2004) reviews the evolution of the previous 55 years of research in the field and the problem has been studied from a wide range of academic disciplines from psychology to decision theory using all types of methodology ranging from case studies to mathematical programming. Although the extensive research conducted of the question it is still debated whether or not a software investment do yield a positive economic return for the investing firms (e.g. Aral, et. al, 2006) or if it just money down the drain.

From an economic point of view the challenge of assessing the return of a software investment can be argued to be the following three relevant issues (Kim and Kim, 2005). 1. Determine the economic benefit from the software investment. 2. Determine the cost for the software investment. 3. Determine the uncertainty associated with the cost and the benefit. The first two issues is the problem of deriving a relevant cash flow for the software from an acquiring point of view due to the consequences that the firm faces from using software (Numminen, 2008a; 2008c). From a developers point of view the first two issues relates to the difficulty of assessing the size of the investments it takes to develop the software and assessing the expected net cash flows from selling it. Given these difficulties there will be uncertainty associated to the economic benefit and cost that needs to be assessed and managed. All of these issues will be further elaborated on below to motivate this study.

Consequences from software investment

Assessing the economic return of a software investment has been and is still a puzzle in the software economics literature as discussed above and further in e.g. Lefley (2004); Smithson and Hirschheim (1998); Renkema and Berghout (1996). As already noted above, many have argued that assessing the economic return of a software investment is more difficult than valuing a traditional investment. This point is not agreed upon by everyone; Powell (1992) analyzes the literature on software assessment and the characteristics of a software investment to conclude that assessing the economic return of a software investment is difficult, but there are models for overcoming this that are not widely used, e.g. dynamic models such option based valuation models. Similar results can also be found in e.g. Lesjak and Vehovar (2005); Ballentine and Stray (1999); Hochstrasser and Griffiths (1991).

Much of the discussion of the problems of assessing the economic return of a software investment has been accounted to the characteristics of the consequences from a software investment (Gunasekaran et al., 2001). A consequence is defined as an event that occurs due to a software investment and can be financial or non-financial (Renkema and Berghout, 1996). It is due to these consequences, i.e. the non-financial consequences, that a software investment is been argued to be more difficult to assess the economic return from than a traditional investment (e.g. Fichman, 2004; Halikainen et al., 2002). The difficulty to assess the consequences from a software investment is mainly due to their economic characteristics, e.g. that they mainly lead to intangible consequences and due to time lags between costs and revenues (e.g. Lee and Kim, 2006; Love et al., 2000).

The categorization of consequences from a software investment has been a focus in previous research. This research is summarized in Numminen (2008a), and partly in Chapter 4, where the economic characteristics of these consequences are analyzed. These economic characteristics make it difficult, according to previous research, to derive a cash flow function for the investment which is needed for the valuation of a software investment (Im et al., 2001). The problem is studied in Chapter 4 where the economic characteristics of the argued consequences from a software investment are analyzed to see how a cash flow function can be defined that takes them into account. The analysis is focused on how and when the consequences will affect the cash flow function from a software investment. This issue will be further discussed in Chapter 10 to motivate a different view of how intangible consequences can be transformed into cash flows.

An issue that further complicates the derivation of a cash flow function for a software investment is determining the actual usage of the software (e.g. Venkatesh and Davis, 2000; Venkatesh et al., 2003). The usage is a necessary variable since it is not the amount of money invested in software as such that affects the economic return but the extent to which the software is used (Devaraj and Kohli, 2003). But, increasing the amount of money invested in making the software easy to use has shown positive effects on level of software usage (Davis, 1989; Szajna, 1996; van der Heijden, 2004). So it is usage that leads to the consequences that the cash flow function needs to take into account. However, research, e.g. Davidson and Heslinga, 2007; Fichman and Kemerer, 1999, show that adoption of new technology do not follow the predicted adoption curves due to assimilations gaps. Given these issues some research, e.g. Orlikowski

(2009; 2007); Orlikowski and Scott (2008), state that it may not be meaningful to separate the effect from the software from the overall organizational output where the software is implemented since we observe the co-effect of both.

The problem concerning the economic characteristics of software investments have even lead some to suggest that it is pointless to use formal methods for assessing the economic return from a software investments since they do not capture the true return of the investment (e.g. Milis and Mercken, 2004; Lubbe and Remenyi, 1999; Anandarajan and Wen, 1999; Bellantine and Stray, 1999; 1998); Falk and Olve, 1996; Lefley, 1996; Small and Chen, 1995; Lefley, 1994). Drawing this conclusion is the same as saying that investment decision can be arbitrarily made without any consideration of how scarce resources are used. Or, from a financial economics point of view, we do not care if we earn the return on invested capital that is motivated by the market-based opportunity cost of capital.

An investment under uncertainty

Another aspect that makes the economic return of a software investment difficult to assess is the cost for the development process of the software. (Dey at al., 2007) state that software development projects often suffers from time and cost overruns as well as from quality underachievement. This is a source of major problems since software is expensive to develop (Liu and Mintram, 2005). The above is also validated empirically by Jeffery and Leliveld (2004) where it is estimated that 68% of corporate software investments do not meet their budget or are not completed in time, and projects for $100 billion to $150 billion has failed totally for US firms alone between 2002 and 2003. Similar results are given in Wilcocks and Lester (1994) were it is shown that over 22% of the invested money in software are wasted and in the range of 34% to 40% of all projects yield no net economic return regardless of how it is measured. For more similar results see e.g. Baccarini et al. (2004); Barki et al. (1993) and see Standing et al. (2006) for an analysis of the attributes of success and failure in software projects.

Given the uncertainty involved in a software development project, as suggested by e.g. Dey et al. (2007); Baccarini et al. (2004); Schwalbe (2002), it may be beneficial to treat the development as an investment under uncertainty instead of using naïve cost based methods (Erdomus et al., 2006 in Biffl et al., 2006). An advantage of doing so is that the focus in on the economic return created rather than the costs saved during the development (Boehm, 2003). Viewing a software investment as an investment under uncertainty may be part of the answer to the above arguments of the impossibility of software valuation. Changing the view of the investment also allows us to use more general models and methods when assessing the economic return investment since more aspects of the economic characteristics of a software investment can be included (Numminen, 2008a; 2008c).

An advantage of acknowledging the uncertainty is the flexibility, i.e. real options, which can be included when assessing the economic return of the software investment. The point of including options is that decisions are made based on the time-dependent information, i.e. today's decision are made with today's information and tomorrow's decisions are made based on tomorrow's information (e.g. Trigeorigis, 1996; Dixit and Pindyck, 1994). This results in asymmetric cash flows that require different valuation models for the economic value of the flexibility to be included. There are different types of options that can be created in most software development projects to manage the uncertainty in the project. Under what circumstances and how these options can be created is analyzed in more detail in Chapter 5 and Chapter 6.

Since the seminal works of the importance of real options in strategic decision-making (Kester, 1984; Myers, 1984; 1977) a wide range of disciplines has adopted the idea of including flexibility in decision-making, see e.g. Li et al. (2007); Reuer and Tong (2007) or Schwartz and Trigeorgis (2001) for extensive literature reviews. Although the advantages offered by an option-based management and valuation of investments the approach has also been criticized. Borison (2005) analyzes the applicability, assumptions and mechanics of the common used real option valuation models used in research and by firms to conclude that they all have shortcomings. This view is also shared by Steffens and Douglas (2007) who agrees with the underlying thought of the approach but do not agree with the applicability and mechanics of the approach. However, e.g. McDonald (2006), Copeland and Antikarov (2001), Trigeorgis (1996), Dixit and Pindyck (1994) show that traditional static discounted cash flow (DCF) valuation is based on more restrictive assumptions than real option valuation even though it is the most used valuation method in the corporate world (Ryan and Ryan, 2002; Graham and Harvey, 2001).

As another way of dealing with the uncertainty in software development, research has focused on software product line engineering in which previous software components are reused in new software (Pohl et al., 2005). For firms to optimally benefit from this strategy there are two issues that must be taken into account. Firstly, what functionalities should be included in the software platform that is reused to reduce the uncertainty the most. And secondly, what functionalities shall be included in the platform and what functionalities shall be included in the specific software to create as many options as possible for future development, i.e. how shall a software platform be designed to maximize the number of software that can be derived from it.

PURPOSE AND RESEARCH QUESTIONS

The aim of the above discussion is to show that there is need for research in the field of software valuation. More specifically, there is a need for more research on a more comprehensive level that analyzes how the different parts of the value creation process of software investment interrelate and on the economic nature of this relation. The research needs to take into account how to manage the risk and uncertainty during the development of the software and the resulting economic effect on the investment. It should also take into account how the economic return of a software investment can be assessed as well as how this return can be managed to maximize it.

By doing the above this dissertation adds to the research by taking a more comprehensive approach and not only focuses on one aspect of the software investment economical value puzzle.

Purpose

Based on the discussion above we state the purpose of this dissertation as:

- Explore how the economic return of a software investment can be assessed and managed.

Research questions

The below research questions are studied in this dissertation. The research questions are motivated by the above purpose and by the discussion of the different economic aspects of a software investment valuation above. In order to fulfill the above stated purpose the following research questions will be answered.

1. How can a cash flow function be defined for a software investment that takes into account the uncertainty in expected usage and the resulting consequences?
2. How can the economic return of a software investment be modeled in the case of no managerial flexibility?

These question are motivated to study because it is argued in previous research, as discussed above, that a cash flow function cannot be defined that takes into account the economic characteristics of software investment. The next problem is to develop a valuation model that can assess the economic return of a software investment. We assume no flexibility, i.e. no real options are included, in the simplified setting. These questions are analyzed in Paper 1 in Chapter 4 below.

3. How can managerial flexibility be incorporated in a software investment?
4. How does managerial flexibility affect the economic return of a software investment?
5. How does managerial flexibility reduce the uncertainty in a software investment?

To make the valuation of a software investment more general and under less strict assumptions we analyze how managerial flexibility, i.e. real options, can be incorporated in the valuation. The main issue here is how can these real options be created and what economic value will it add to the software investment as well as analyze how this affects the uncertainty of the investment. These questions will be analyzed in Paper 2 in Chapter 5 and Paper 3 in Chapter 6 below.

6. How can the uncertainty in software development be managed by software platforms?

7. What type of managerial flexibility is created by software platforms?
8. What constitutes an efficient software platform for uncertainty management in software development?

To reduce the uncertainty during software development previous research have argued for a component-based development methodology of software. To implement a strategy like this, it is important to analyze what components shall be combined into software platforms given the type of uncertainty that the developer aims to manage. To do so, we need to analyze what constitutes an efficient software platform and the role played by software platforms to create managerial flexibility. These issues are analyzed in Paper 4 in Chapter 7, Paper 5 in Chapter 8 and Paper 6 in Chapter 9 below.

A MODEL FOR THE ECONOMIC RETURN OF A SOFTWARE INVESTMENTS

Given the discussion above of the economic characteristics of a software investment and the purpose of this dissertation we need model for assessing the economic return of a software investment. The model needs to take into account that the ability to manage the return to maximize it.

For the model to be useful it needs to guide the firm on which software investments to undertake and which to pass. Before we can state such criterion we must determine why firms make software investments. This issue has been greatly analyzed in previous research; in e.g. Lin (2007) and Dedrick et al. (2003) the main desired outcome of a software investment is argued to be increased economic performance in the form of economic growth, increased labor productivity, increased profitability and consumer welfare. Mitra (2005) adds long-term decrease of operating costs and thereby and increase of the firm's cash flow as a desired outcome. For the software to do the above in an economically justified way it is implicitly or explicitly assumed that the software must add more value than it consumes. We therefore pose the following criterion for making a software investment:

$$MAX[V(S),0], \tag{1}$$

where $V(S)$ denotes the economic return of a software investment.

Equation 1 states that a software investment shall be made if, and only if, the economic return of the investment is greater than zero. This decision rule is directly related to the discussion above of the reasons why firms make software investments. But it is also justified from a financial point of view that an investment should add more value to the firm than it costs which will be further elaborated on below. In Equation 1 $V(S)$ is measured as the difference between the present value of the expected cash flow and the investment cost of the software.

The goal function of the firm

On a general level the decision rule in Equation 1 is linked to the assumed goal function of the firm, stated as maximizing the long-run market value of the firm (e.g. Brealey and Myers, 2003; Ross et al., 2007). This goal function is referred to as the value maximization proposition by Jensen (2001a; 2001b). A firm's goal function is the way that the principals (owners of the firm) assigns how the agents (managers) of the firms shall behave, i.e. run the firm. The rationale for the value maximization proposition is that we only seek to maximize one variable, and the most relevant one for the owners, instead of many as suggested by other goal functions, e.g. the stakeholder theory (Freeman, 1984). This makes the delegation easier and more precise for the principals of how their capital should be managed but also gives a better guidance for the managers when e.g. choosing between mutually exclusive investments alternatives (Rose, 1999).

The theoretical foundation for the value maximization be found in the accountability principle (Cools and van Prag, 2003), which states that a limited number of communicated quantified measures and targets will increase accountability and lower costs for effective designed incentives to create shareholder value. This is the total opposite to the informativeness principle (Holmstrom, 1979) stating that a maximum number of measures would be optimal, as implied by the stakeholder theory.

The value maximization proposition stems from the set-of-contracts-view of the firm (Alcian and Demetz, 1972; Jensen and Meckling, 1976) which is based on principal-agent theory, see e.g. Bolton and Dewatripont (2005), Laffont and Martimort (2002), Rapp and Thortsenson (1994) for an extensive coverage of the theory or Eisenhardt (1989) for a discussion of the seminal papers. It is due to these pre-negotiated contracts that firms can earn excess returns on investments in comparison to the security market line (SML) developed by Sharpe (1964), Lintner (1965), Mossin (1966), i.e. make investments with positive economic returns and create wealth in comparison to investments in the financial market.

Now when the boundary condition for which investments the firm shall accept we can pose conditions on the valuation model. Since we make the initial decision of undertaking the investment today, i.e. at $t=0$, the valuation model must translate the future uncertain cash flows into today's equivalent time-value, i.e. it must be a discounted cash flow model for it to be in line with the value maximization proposition since expected utility must grow as function of time and uncertainty for investors to be willing to postpone consumption (Lancaster 1963; Hirshleifer, 1958). The result is derived in e.g. De Matos (2001) and critically reviewed in Frederick et al. (2000).

We also require that the principals are compensated for the uncertainty in the software investment in accordance with similar opportunities in the financial market. We do so by either using a risk-adjusted discount rate in the valuation model or by using risk-neutral measures to govern the probability distribution of the realization of the cash flow function. This is further elaborated on below. The key underlying assumption here is that an investor strictly prefers higher utility over lower utility which the valuation model must take into account by using a justified discount rate in the present value calculations.

Managerial flexibility

Finally, the valuation model must explicitly take into account the asymmetric structure of the resulting cash flows that are created by managerial flexibility, i.e. by exercising the real options incorporated in the investment decision (Copeland and Antikarov, 2003; Trigeorgis, 1996; Dixit and Pindyck, 1994). The reason for incorporating real options into the investment analysis is to maximize the return of the software investment by hedging against bad outcomes and maximizing good outcomes. By not taking into account this flexibility, we would underestimate the return of the software investment (Dixit and Pindyck, 1995a; 1995b).

From and technical point of view the incorporation of options also enables us to slack the assumptions that the net present value (NPV) valuation operates under (Dixit and Pindyck, 1994), i.e. there is no uncertainty about the future, it is a now-or-never decision and managers cannot or will not take actions to steer away from bad outcomes or maximize good outcomes, and finally the NPV calculation assumes that the investment is fully reversible. These assumptions are not valid for a software investment or a rational manager.

Stating the model

Given the above discussion we can state the assessment model for the economic return of a software as

$$V(S) = \sum_{t=0}^{T} \frac{E[CF(t)]}{(1+k)^t} + \sum_{t=0}^{T} \frac{E[O(t)]}{(1+k)^t} - \sum_{t=0}^{T} \frac{E[I(t)]}{(1+k)^t}, \qquad (2)$$

s.t. Equation 1,

where:

V(S) previously defined in Equation 1,
E denotes the expectation operator,
I(t) denotes the time-dependent investment cost function,
k denotes the discount rate,
CF(t) denotes the time-dependent cash flow function of the investment, and
O(t) denotes the value of managerial flexibility, i.e. the option values.

Equation 2 summarizes the discounted cash flow and subtracts the discounted investment cost for the software, and adds the present value created by the managerial flexibility. The assessment model is a special case of a NPV function in discrete time. Equation 2 can be seen as general version of the expanded NPV in Trigeorgis (1996) in the sense that it explicitly allows for the investment cost to be time-dependent with or without real options included.

Further, Equation 2 has two mathematical features which make it suitable for assessing the economic return of a software investment and how to manage this return. The function is separable in time and space which makes it general enough to differentiate the known present from the uncertain future and to separate the effect of the different input variables, i.e. the different sums in Equation 2 (Rapp, 1974). For the return function to be complete we assume that the financial decision can be separated from the investment decision as a result of (Modigliani and Miller, 1958; Fisher, 1930), see e.g. Rubinstein (2003); DeMatos (2001) or the seminal papers for a more technical discussion and derivation of the result. These two features can also be derived by using the arbitrage theorem developed by Ross (1976). For a greater technical discussion about the functional form of Equation 2 see Numminen (2008a).

Taking uncertainty into account

A final aspect about Equation 2 must be commented on. *E* in Equation 2 denotes the expectations operator which is included to handle the uncertainty in cash flow from a software investment as discussed above. We do so by assigning probabilities to the different cash flows and calculate the expected value, i.e. the weighted value. This transforms them into an expected cash flow which takes into account the uncertainty that is assumed for the decision setting. For these probabilities $0 < p_i < 1, \sum_i p_i = 1$ apply, see e.g. Falmagne (2003) or Papoulis and Pillai (2002) for a discussion of the properties of probabilities and e.g. Aggoun and Elliont (2004) for a discussion about the measure theoretical constructs of probability measures. Note that we require the probability to between zero and one for an outcome, i.e. we do not allow a probability of zero or one. This will be further elaborated on below and is motivated by the view we have on the setting for the investment.

Depending on the assumptions that we make about the world, i.e. the setting where the investment takes place, we can distinguish between decisions under three types of settings; decision under certainty, decision under risk and decision under uncertainty (Knight, 1921; Luce and Raiffa, 1957). We say that a decision is under certainty when it is know that the decision leads invariably to one, and only one, outcome, i.e. the true outcome is known *a priori*. However, as discussed above, a software investment is made under uncertainty and therefore does not invariable yield an economic return known *a priori*. Therefore we cannot assume that software investment is a decision under certainty as done in the previous research discussed above. The difference between a decision under risk or uncertainty is whether it is possible, or if it is meaningful, to assign probabilities to a specific outcome. In the case of risk we assume that probabilities can be assigned and under uncertainty they cannot or it is not meaningful to do so. This type of uncertainty is often referred to Knightian uncertainty since it makes strong assumption about the lack of knowledge about the world, i.e. the setting in which the decision is made, or that the world is inherently uncertain which other scholars do not agree upon.

However, since decision making in a world under Knightian uncertainty is less precise we want to move the decision making into a world under Knightian risk instead. To do so we interpret the Knightian uncertainty as a decision setting where we cannot assign any probabilities and Knightian risk as a decision setting where probabilities can be assigned objectively or by other means, e.g. subjective probabilities or risk-neutral probabilities, i.e. martingale measures, see below for a discussion. This type of situation is what is normally discussed as uncertainty in financial economics and will be used here as well. By doing so we can use a larger, and a more relevant, set of models for the decision making and valuation of the investment. On a general level we can view risk as a way to operationalize the uncertainty by using probability theory.

Since this dissertation uses stochastic processes, i.e. uncertain cash flows over time modeled as a sequence of random variables (Aggoun and Elliot, 2004; Papoulis and Pillai, 2002), the above distinction and transformation of the setting becomes important in order to know how to handle the realization of the process. We motivate the use of stochastic processes since the realization of the different cash flows from e.g. developing software or using software are uncertain, as argued above. To be able to manage this randomness we assign values for the distribution of the various cash flows over time based on our degree of belief or by other means. The use of Knightian risk, i.e. a probabilistic approach, is also implied by the definition of an investment above.

A risk-neutral world

Before concluding the discussion about, and the setting for, Equation 2 above a note must be made to clarify the meaning of the terminology in the discussion of probability measures above. As noted above a probability is a measure of the degree of belief or the frequency of occurrence of an outcome in general terms. And since the cash flows of the software investment are uncertain we need to assign probabilities to different possible cash flows to take this uncertainty into account in the return calculation and when assigning the discount factor.

This is the catch 22 of valuation since assigning a correct discount factor, i.e. the risk-premium for taking on uncertainty, we basically need to know the cash flows from the investment in order to find a correct benchmark portfolio in an equilibrium model, i.e. avoiding arbitrage. So if we can assign a correct discount rate the valuation problem is solved. This catch can be resolved by transforming the setting into a risk-neutral world where the discount factor is set to the risk-free rate of return. To do so we have to assign risk-neutral probabilities to govern the uncertainty of the decision setting instead of the real world probabilities used otherwise. It shall be noted that risk-neutral probabilities are not real probabilities in the sense that they measure a degree of belief but merely a mathematical technique to overcome the risk-premium problem. The two techniques will give the same answers so choosing one before the other do not result in a different economic return for the software investment. This technique will be used when suitable later on in this dissertation since it handles the underlying problem of assessing the economic return of a software investment. For a more theoretical discussion of the method the reader is referred to e.g. Chapter 5 below; Schreve (2004a; 2004b); Neftci (2000)

The assessment model and the research questions

The six papers that make up the main part of this dissertation are related to Equation 2 in the following way:

Paper 1: Modeling the Economic Return of a Software Investments, based on (Numminen, 2008c), analyzes how a cash flow function can be defined for a software investment and how a software investment can be valued based on the usage of the software. The assessment model incorporates the economic characteristics argued in previous research for the consequences of a software investment in the setting of no managerial flexibility. This paper analyses how cash flow from a software investment affects the value of the software investment, i.e. the first summation in Equation 2. Paper 1 is presented in Chapter 4 below.

The above is the essence of research questions 1 and 2 previously stated as:

- How can a cash flow function be defined for a software investment that takes into account the uncertainty in expected usage and the resulting consequences?

- How can the economic value of a software investment be modeled in the case of no managerial flexibility?

Paper 2: Hedging Uncertainty in Software Development – A Real Option Modeling Approach, based on (Numminen, 2010a), develops a model for how the most arbitrarily real options can be included in software development projects. The paper analyzes what value is added from this and what uncertainties are reduced. The analysis is generalized to also discuss how these options can be incorporated by an acquiring firm as well. Therefore, this paper also analyses how all three summation in Equation 2 affects the economic return of a software investment Paper 2 is presented in Chapter 5 below.

Paper 3: Valuation of a Software Investment When you Have the Option to Customize it – The Case of Open Source Software, based on (Numminen, 2008b), addresses the issue of how much a software investment should be customized and how this is related to the cash flow function when the option to customize is included in the software. This paper analyses how all three summations in Equation 2 affects the value of a software investment. Paper 3 is presented in Chapter 6 below.

The above is the essence of research questions 3 to 5 previously stated as:

- How can managerial flexibility be incorporated in the software investments?

- How does managerial flexibility affect the economic return of a to a software investment?

- How does managerial flexibility reduce the uncertainty in software investments?

Paper 4: Uncertainty Reduction in Software Development by the use of a Platform-Based Development Strategy, based on (Numminen and

Wrenne 2009), analyzes the usefulness of a platform-based development strategy to create options for future software development and how this strategy can be used to manage uncertainty during development. This paper analyzes the relation between created options and the cost of developing software, i.e. the ratio between the second and third summation in Equation 2. Paper 4 is presented in Chapter 7 below.

Paper 5: The Role of Platforms in Software Development – Creating Real Options to Manage Uncertainty, based on (Numminen and Wrenne 2010), further develops paper three and empirically validates the developed model. This paper is therefore also analyzing the relation between the second and third summation in Equation 2. Paper 5 is presented in Chapter 8 below.

Paper 6: Why Software Platforms Make Sense in Risk Reduction in Software Development – A Portfolio Theory Approach, based on (Numminen, 2010b), develops the underlying mathematical model for Paper 3 and Paper 4. This paper is therefore also analyzing the relation between the second and third summation in Equation 2. Paper 6 is presented in Chapter 9 below.

The above is the essence of research questions 6 to 8 previously stated as:

- How can the uncertainty in software development be managed by software platforms?
- What type of managerial flexibility is created by software platforms?

- What constitutes an efficient software platform for uncertainty management in software development?

Chapter 10 will extend the analysis and conclusions of the six papers into a general analysis based on the problem motivated in this chapter. The analysis will focus on how the various parts of the return function of a software investment, i.e. Equation 2, affects the economic return of a software investment in a general setting

The six papers will be more thoroughly discussed in Chapter 3.

THE SIX PAPERS

The six papers that constitute the main body of this dissertation are based on the below papers. The papers will, as above noted, be more thoroughly discussed in Chapter 3.

Paper 1: Numminen, E (2008) "Modeling the Return of Information System Investments", *Proceedings of the 2nd European Conference on Information Management and Evaluation*, pp. 325-334

Paper 2: Numminen, E. (2010) "Hedging Uncertainty in Software Development – A Real Option Modeling Approach", Presented at the 24th European Conference for Operations Research in Lisbon, Portugal, unpublished.

Paper: 3: Numminen, E. (2008) "Valuation of a Software Investment when you have the Option to Customize it – The case of open Source Software", presented at the 18th Triennial Conference of the International Federation of Operational Research Societies in Johannesburg, South Africa, unpublished

Paper 4: Numminen, E., Wrenne, A (2009) "Uncertainty Reduction in Software Development by the use of a Platform Based Development Strategy", *Proceedings of the 3rd European Conference on Information Management and Evaluation*, pp. 356-362

Paper 5: Numminen, E., Wrenne, A. (2010) "The Role of Platforms in Software Development – Planting Real Options to Manage Uncertainties", *Proceedings of the 1st International Conference on Information Management and Evaluation*, pp. 289-295

Paper 6: Numminen, E. (2010) "Why Software Platforms Make Sense in Risk Reduction in Software Development – A Portfolio Theory Approach", *Proceedings of the 4th European Conference on Information Management and Evaluation*, pp. 282-290

CHAPTER 2 ECONOMIC ASPECTS OF SOFTWARE

In this chapter we will explore the different aspects of a software asserted by the definition of a software stated in the previously in Chapter 1. This is done to better understand the underlying economic properties of software from which the economic value is derived. The discussion in this chapter is mainly based on von Eisenhardt (2008) from which the definition of software in this dissertation also is from. The chapter concludes with a discussion about different types of software on a general level and their economic characteristics to suggest a more comprehensive view of what software can be.

PROPERTIES OF A SOFTWARE

This section will discuss software based on the different economic issues that can be derived from the definition of software stated in the previous chapter.

Data processing

The fundamental functionality of software is data processing. This is captured in the IPO-principle stating input-processing-output for the exchange of commands in the interaction between the different parties involved. Depending on the setting, the exchange may occur between user and software, software and software or user and user. The value of this exchange in general terms is depending on compatibility between different parts of the system, i.e. hardware, software and user, where the software is implemented. This compatibility may results in network effects (Shapiro and Varian, 1999) which can explain the increase in demand for popular software as we see in real-life.

System of commands or instructions

Software is a logic construct of algorithms and instructions for data processing, performing one or several tasks in creating information. Software can thus be seen as an information good (Shapiro and Varian 1999). However, depending on the type of software analyzed we may end up with the information paradox (Thorp, 2003). In the case of proprietary software the buyer only gets the binary code and not the source code. This does not enable the buyer to validate the value of the information before the transaction. If the information, i.e. the source code, is available for evaluation then no one would buy the information. This constitutes the paradox. The same situation may not occur in the case of open source software (OSS) since the source code is distributed. The user can therefore validate the information prior to the exchange. This explains why there is no market, from an economic point of view, for OSS as such but for complimentary products and services instead. The user also has the option to make changes to the source code to have software that better fit hers or his needs if desired (Numminen, 2008b).

A recombinant system

Software is a digital good consisting of ones and zeros. These ones and zeros combined in designated ways becomes the algorithms and instructions for data processing, i.e. the exchange of data. In this sense these ones and zeros can be recombined into new components and be used when developing new software. The resulting uncertainty reduction and added economic value of this development strategy is studied in Numminen (2010a) and Numminen and Wrenne (2009; 2010).

Discrete unit consumption

Software represents from a standard economics point of view a non-ordinary type of a product. A software x is consumed in discrete units and most often in only one unit, i.e. $x \in [0,1]$ in formal terms. This poses challenges when pricing software since the firm is only expecting to sell once to a customer in comparison to e.g. a grocery seller. So in order to gain the maximum from the customers the firm may try to price discriminate and sell complementary products (Shapiro and Varian, 1999).

Firms may also try to get the consumer to buy a new version by implementing a revolutionary development strategy (Beinhocker, 1999; Shapiro and Varian, 1999). An example of a firm using this strategy is Microsoft where backwards compatibility is limited when sufficient structural changes are made in new versions. It then becomes important not only to analyze what type of functionalities to include in new versions, as studied by e.g. Mukherji et al. (2006), but also when to release new versions, as analyzed by e.g. Kauffman and Wu (2006).

A complex system

Software can be viewed as a complex system of code based on logic operations. Modern software is made up of tens of millions of lines of source code. It is estimated that on average there is 0.51 errors in per one thousand lines of codes (von Eisenhardt, 2008). Given this, it is not surprisingly that there are errors in various types in software releases since not all of these errors are found and fixed before release. The cost of maintenance of a software is substantial (Zhang and Windsor, 2003)

Some research suggests it even accounts for 50%-80% of the total system life-cycle cost (Krishnan et al., 2004).

Since the quality of software is discovered by usage we say that software is an experience good (Shapiro and Varian, 1999). Being an experience good, the consumer does not know the quality of the software prior to the purchase and can therefore not be sure of the utility he or she will get from the software as discussed above. We therefore say that the decision is made under uncertainty. The problem of determining *a priori* utility from software on a general level is analyzed in Numminen (2008a; 2008b; 2008c).

A intangible product

By being made up of ones and zeros, software can be seen as an intangible good that is dependent on a carrying medium to be distributed and consumed. Being an intangible product gives software very specific economic characteristics. When software is duplicated the second copy becomes a perfect copy of the first one and there is in theory no natural limit to how many copies can be made. This means that there is no rivalry in consumption of software but only for the carrying medium, e.g. a CD-disk or a flash memory. In this sense software resembles a public good more than a private good.

DIFFERENT TYPES OF SOFTWARE

There are several ways different types of software can be categorized. The below discussion aims at differentiating types of software based on their general economic characteristics. The description is in no way complete but shows the various economic characteristics depending on the software type.

The legal issue

We can differentiate between different types of software based on their characteristics or based on the dimension of the software by which we would like to separate different types of software. From a legal point of view we can have a OSS, e.g. Linux, or an closed form of software (proprietary software), e.g. Microsoft Windows.

If the software is of proprietary type or OSS has economic implication for the acquirer. There is an economic advantage of the OSS in two ways (Numminen, 2008b). Firstly; generally an OSS can be downloaded for free and therefore there are no up-front costs for acquiring the software. Secondly; by acquiring an OSS you not only get the software but also the underlying source code which you do not have access to in proprietary software. This means that the firm also has the possibility to customize the software to better fit the needs of the organization and therefore increase the usage of the software. It shall though be noted that this option may not included in all OSS type of software products.

Investment cost

We can also differentiate software on the basis of the investment cost for buying the software, i.e. free-ware, e.g. Linux, versus buy-ware, e.g. Microsoft Windows. An issue that combines the legal and the acquiring matter is whether the consumer buys or licenses the software. This decision changes the timing of the cash flow between the parties and could also change the size of the cash flow between the parties. If we buy the software we pay the total cost up-front whereas if we license the software we pay over time for the software.

Another issue concerning the acquiring of the software is the uniqueness of the software. An off-the-shelf-software is the same software who ever buys it whereas bespoke software is tailored made based on the needs of the individual buyer. The bespoke software is therefore unique whereas the off-the-shelf-software is general software aimed for a mass-market. We can also separate on the basis of whether this development is done in-house or paid for. A middle type in this differentiation is mass-market customized software where changes are made on general software to better fit the individual needs and represent a more cost-effective development (Kreuger, 2000).

One way of accomplishing the mass-market differentiation is to sell module-based software instead of complete feature software. Based on their needs the customers will decided which modules to buy, e.g. toolboxes in MatLab or different module of an ERP-system. This differentiation of the product can also be made on the software level where some software is sold in bundles and other not, e.g. Microsoft Office versus MatLab. A bundling strategy instead of a tying strategy is beneficial if it reduces the variation in the willingness to pay among the customers (Shapiro and Varian, 1999)

Functionality of the software

We can also differentiate different types of software based on the tasks they perform. Underlying software, e.g. an operating system, performs the underlying tasks such as communicating with the hardware. An application performs the tasks that the user instructs.

Software can also be categorized based on independence, i.e. embedded software versus independent software. The former is when the software is part of a system of goods and only constitutes a part of the functionality of the system, e.g. software used in cars. The latter refers to when the software is independent, e.g. an ERP software. In a strict sense all software are embedded since they will not function without a medium by which the software can be utilized. The same distinction can be made between the number of functionalities the software performs. Some software is designed for a single use whereas other software is designed in the line of general purpose technology.

Operation of the software

Finally, we can distinguish between software that are self-operating and those that require manual operation, e.g. an autopilot that flies an airplane by itself in comparison to Microsoft Word where the user controls the operations by commands. However there is in most cases the possibility to self-operating features in manually operated software as well, e.g. automatic save function in Microsoft Word.

The above discussion aims at showing the many economic dimensions of software. Depending on the type of software discussed different economic issues are relevant to take into account to determine the economic value. However, the assessment model developed in this dissertation is general enough to handle these differences on a general level. This issue will be returned to in Chapter 10.

CHAPTER 3 DISCUSSION OF THE SIX PAPERS

In this chapter we will introduce the six papers and discuss their main content. Each paper will also be discussed in the light of the previous research for the problem domain of the papers.

PAPER 1

Background

Previous research has argued that it is difficult to value a software investment (e.g. Irani, 2008; Gunnasekaren et al., 2006; Murphy and Simon, 2002). Some researchers have even argued that it is impossible or pointless to do so (e.g. Milis and Mercken, 2004, Lubbe and Remenyi, 1999; Anandarajan and Wen, 1999). The main argument has been that the usage of a software leads to economic consequences that cannot be included in formal valuation models. The aim of this paper is to analyze why some consequences pose problems from an investment point of view and develop a valuation model that incorporates these consequences.

Aim of the paper

This paper analyses the consequences argued to be difficult to assess. By defining a goal function for the investing firm and analyzing how and when various consequences have an effect on the cash flow from the investment, a cash flow function and a valuation model is developed for the software investment.

Due to the characteristics of the consequences this paper argues for the use of a stochastic cash flow function to capture the effects on the cash flow from the investment. Due to the stochastic cash flow function, this paper proposes a valuation model based on a stochastic differential equation to value software investments.

Validation of the results

The empirical validation of the deducted results in the paper is suggested for future research. This can be done in either of two ways. The first way is to analyze relevant data from a software investment to see if it has the characteristics that the models assumes. The problem posed by validation in this way is that it is difficult to find the detailed data about what consequences did occur and the cash flow it resulted in. The second way is to empirically validate the assumptions underlying the model to see whether they hold in an empirical context.

PAPER 2

Background

Real option theory is not new in the field of software economics and was introduced due to the shortcomings of traditional valuation methods to incorporate managerial flexibility in the valuation, see e.g. Benaroch et al. (2007); Benaroch et al. (2005); Fichman (2004) or the paper refered to in the paper for a discussion of the literature on real option theory in software economics.

The common denominator for most of the research on real option theory in software economics is the focus on how to value options using continuous-time models.

Aim of the paper

The aim of this paper is analyze the value-added to a software investment from the flexibility created by the three most arbitrarily real options. A focus in the paper is not just on the valuation of these options but also on how these options can be created in software development projects.

Further, a discrete-time model is developed for the analysis which is based on fewer assumptions in comparison to the continuous-time models typically used to value the same three real options. The three real options that are analyzed in this paper is the option to defer, the option to expand and the option to abandon. In the modeling a risk-neutral discount rate is derived to overcome the risk-premium puzzle.

Validation of the results

The deducted results in this paper are validated in the empirical discussion under which these options can be included in software investments.

PAPER 3

Background

The economic research on open source software (OSS) has mainly focused on three issues. Firstly; why individuals, and to some extent firms, participate in open source software projects (e.g. Roberts et al., 2006; Johnson, 2001; Lerner and Tirole, 2001). Secondly; what types of business models can be defined for firms to profit from open source software products (e.g. Mustonen, 2001; Lerner and Tirole, 2000; Raymond, 2000). Thirdly, organizational adoption issues of OSS (e.g. Aija and Wu, 2007; Katsamakas and Xin, 2005; Glass, 2004).

Aim of the paper

Previous research has focused on the cost advantage of open source software as the economic motivation for a firm to choose OSS. In this paper a different point of economic value which is argued for. The aim of the paper is to analyze value-added of OSS from the option to customize the software. Customized software is argued to increase the usage and thereby increase the cash flow from the software. The argument underlying the analysis is in accordance with the software usage literature; see e.g. Taylor and Todd (1995) and Davis (1989) for a discussion on this.

This paper analyses under what circumstances an option to customize software shall be exercised and how much resources should be invested in customizing the software, i.e. what is the cap on the exercise price for the option, in order to maximize the economic value of it. It is also shown in the paper how to the valuation model changes depending on how the implementation cost are dealt with by the firm. The main part of the paper is developing valuation models for these decisions and discussing their validity. The 4 decision settings analyzed in the paper are: 1). When the firm does not make any adjustments to the software and there is no investment cost. 2). When the firm does not make any adjustments to the software but there is an investment cost. 3). When there is uncertainty about the future level of usage for a given level of customization costs. 4). When the customization cost is uncertain for the future level of usage. Finally, it is also discussed how the models can be generalized for proprietary software.

Validation of the results

The empirical validation of the deducted results in this paper is suggested for future research. The results can be validated by studying if firms have exercised the option to change the software in accordance to better fit their needs. Empirical result on this can be found in Shirky (2008) that partly validates the results.

PAPER 4

Background

Previous research has argued that developing a software product is an uncertain and costly venture (e.g. Dey et al., 2007; Tüysüz & Kahraman, 2006; Erdogmus, 2002). This paper focuses on how to reduce these uncertainties and costs during the development cycle of software products.

This paper uses the videogame industry as a case for analyzing how platforms have been used to create real options for developing new products. The paper is differentiated from previous research by taking a more general view of what constitutes a software platform in comparison to e.g. Fichman (2004) or Taudes et al. (2000) where the software platform is defined as the operating system or specific parts of the source code. This gives a technical perspective on what a software platform can be and may neglect important factors that may not be captured by the above. In order to get a more comprehensive view of what can be a software platform this paper uses the concept of a service platform instead, see e.g. Goldstein et al. (2002); Edvardsson et al. (2000) for a definition and discussion.

Aim of the paper

The aim of this paper is to analyze what role a software platform plays in reducing uncertainty in software development and what options are created by the platforms. For the analysis of this paper the video game industry was used as a case. The video game industry was chosen because it has been argued that the success of a videogame is more uncertain than a software product. Three specific examples of video games were chosen to analyze the role played by different types of platforms in reducing uncertainty.

In the paper, a model for the platform-based development strategy is proposed and it is also shown what type of uncertainty is reduced depending on the type of real option that is planted in the software development.

Validation of results

The empirical validation of the induced results of this paper is carried out in Numminen and Wrenne (2010).

PAPER 5

Background

This paper builds on the findings from Numminen and Wrenne (2009) and further develops the model presented there by incorporating a stronger supply side view on software innovations. This is done by developing an operationalized version of a technological innovation systems which is a part of the experimentally organized economy framework, see e.g. Carlsson and Stankiewicz (1991); Carlsson (1995); (1997) (2002); (2004) and Eliasson and Eliasson (1996) for a extensive discussion of the models.

Aim of the paper

The aim of this paper is to empirically validate the further developed model. The empirical study was conducted at a Swedish firm which develops and sells software solutions for the integration of IT and telecommunications.

The conclusion of the study is that software platforms are used and are a prerequisite for offering software at the speed and cost that the market demands. The study also shows that the term software platform has a diversified meaning. Depending on the setting and on the role of the person defining software platform there is a difference in the definition which is in line with viewing a software platform as something more than just source code.

PAPER 6

Background

The model and the analysis in this paper can be seen as an extension of the ideas of using portfolio theory in software development as presented in Butler et al. (1999); Kazman et al. (2001); Asundi and Kazman (2001). It provides a more general modeling based on fewer assumptions than the model presented in Sivzattian and Nuseibeh (2001). The approach of using portfolio theory to guide software development investments is not without critique. Verhoef (2002) criticizes the approach on the basis of the great dissimilarity between the characteristics of a financial asset and software. It shall be noted that this paper is analyzing what functionalities shall be combined into a software platform based on the added returns from these functionalities and the reduction in market risk faced by the firm. So it not the characteristics of the functionality per say that are analyzed but the consumer's preferences to pay for them. Therefore the basis of the critique is not valid for the aim and context of this paper.

Aim of the paper

The aim of the paper is to develop an underlying model for the approach used in Numminen and Wrenne (2009; 2010). It shows how a software platform reduces the market risk of a software investment project and what constitutes an efficient software platform. The model in this paper is developed is using a portfolio theory like approach.

Validation of the results

The deducted approach and supporting model is not validated in the paper but see Berinato (2001); Numminen and Wrenne (2009); (2010) for empirical results on the approach.

CONCLUDING NOTES

The above discussion aims at introducing the six papers that represents the main part of this dissertation. The introduction intends to show the various aspects that are taken into account in assessing the economic return of a software investments and how the return can be managed to maximize it. The point of doing this is to motivate that the study has a more comprehensive approach than previous research.

The contributions of the papers will be discussed in Chapter 10.

CHAPTER 4
PAPER 1

Based on Numminen, E (2008) "Modeling the Return of Information System Investments", Proceedings of the 2nd European Conference on Information Management and Evaluation, pp. 325-334

MODELING THE RETURN OF A SOFTWARE INVESTMENT

Emil Numminen
School of Management, Blekinge Institute of Technology, Ronneby, Sweden
emil.numminen@bth.se

ABSTRACT

The task of *a priori* valuation of software investments has attracted a lot of research for a long time. One of the main themes of this research has been which types of consequences software investments result in and how these consequences can be incorporated in the *a priori* valuation of that investment. Much of this research has stated the problem as how to incorporate intangible consequences in the valuation since intangible costs and benefits are assumed to represent a large part of the conse-

quences from software investment. These consequences are therefore highly relevant in the assessment of software investments.

This paper is concerned with the question of how intangible consequences can be incorporated in the *a priori* valuation of software investments. To answer this question, the paper presents a theoretical model for the valuation of software investments based upon a continuous time discounted cash flow model. The general model argued for in this paper is that usage results in consequences which must be into cash flows to be incorporated in a discounted cash flow model. Usage is chosen as the underlying value creating function since it is the basic underlying function that creates all consequences specific to software investment. Some of these consequences can be measured and valued and thus expressed in cash flows and do therefore not cause any valuation problems. Intangible consequences on the other hand cannot be measured or valued when they occur. If these consequences never affect the cash flow they do not pose a valuation problem. It is more likely however that they will affect the cash flow but at a later time.

In this paper a stochastic cash flow model is developed to incorporate the uncertainty and characteristics of when the intangible consequences affect the cash flow. This is done by using a Brownian motion in the assessment model of the economic return from a software investment. The expectations of the future cash flows are transformed into risk-neutral expectations so a risk-free rate of return can be used as a discount factor.

KEYWORDS

Software investment valuation, software usage, consequences, goal function of the firm, discounted cash flow model, stochastic process.

1. INTRODUCTION

From an a priori point of view the difficulty of software investment valuation is to assess the consequences that the investment results in. Hallikainen, et al. (2002), Fichman (2004) argues that the consequences from software investment are more uncertain and harder to estimate than consequences from a traditional investment. A stated reason for this is that the consequences from software investment are mainly intangible and with time lags, see e.g. Love et al. (2005), Lee and Kim (2006). This result in a valuation difficulty when the most common used methods for evaluating investments do not take intangible consequences into account, see e.g. Bacon (1994).

If the decision of the software investment is based only on traditional investment valuation models this could lead to the approval of the wrong software investments, as indicated above, because not all consequences may be correctly incorporated (Lubbe and Remenyi, 1999). This view is shared by Milis and Mercken (2004) who argue that software investments have special characteristics as high risk, long-time returns and large portion of intangible/hidden benefits and costs. Other researchers have argued that it is harder to estimate the benefits as well the total expense for the investment than in the case of traditional investments (Anandarajan and Wen, 1999). Some researchers (e.g. Bellantine and Stray 1998; 1999; Lefley, 1994; 1996 Small and Chen, 1995) even conclude that investment valuation techniques are not applicable on software investments as a result of the above.

This paper develops a discounted cash flow model for the assessment of the economic return a software investment. To do so, a simplified categorization over the consequences from a software investment is developed to determine which consequences are relevant to include in the assessment. The type of software analyzed in this paper is mainly software that is acquired and not developed in-house. The rest of the paper will be structured as follows: Section 2 will present the setting for the paper by discussing and defining the goal function of the firm. The goal function is needed to determine which consequences are relevant to include in the assessment of a software investment. Section 3 will discuss previously used categorizations of consequences from a software investment and motivate a simplified categorization of consequences. In section 4 a suitable cash flow function for a software investment will be derived based upon the discussion in section 3. Section 5 will deal with the development of the analytic model for the assessment of a software investment based on the cash flow function from section 4. The paper ends with conclusions and suggestions for further research in section 6.

2. THE SETTING FOR INVESTMENT ASSESSMENT

Since the firm can have several goal functions we must define what we mean with a goal function to determine a useful goal function. For this paper a goal function is defined as the purpose of existence for the firm. For the goal function to be useful it must assist the managers of the firm running the firm and help them deciding which investments shall be undertaken and not. In this paper we set the goal function to be the maximization of the long run market value of the firm (cf. Brealey and Myers, 2003; Ross et al., 2007). This goal function is called the value maximization proposition (Jensen, 2001). Total value in this sense does not only include equity but debt, preferred stocks and warrants, i.e. the total value of the firm's long-run financial capital. The firm maximizes the long run market value of the firm by maximizing the cash flow to the firm through activities with positive returns.

2.1 CASH FLOW BASED MODELS

Firms make investments in order to increase the long-run market value of the firm. In order for the firm to be able to know whether the investment increases or decreases the long run market value of the firm, the assessment of the investment must be based upon the incremental cash flows from the investment. A cash flow can be defined as the residual between the inflow and outflow of money for the firm due to the investment. More formally the cash flow can be defined as

$$CF_t = OCF_t - \Delta NWC - \Delta Cap \quad (1)$$

where

$$OCF_t = EBIT_t + A_t - Tc_t \quad (2)$$

and ΔNWC denotes the change in the net working capital during the period. $\Delta Cap = NFA_t - NFA_{t-1} + A_t$ where NFA denotes the net fixed asset, Tc_t denotes the corporate tax for the period and A_t denotes the depreciation for the period. OCF_t denotes the operating cash flow at time t and $EBIT_t$ denotes the earnings before interest and taxes. It can be seen from Equation 1 and 2 that the cash flow is a residual measure of the relevant benefits and costs given the goal function of the firm. This is the reason why cash flow is used in the modeling of the economic return from software investment instead of other measures.

2.2 Investment criterion

In order to assess a software investment, a rule must be defined for when to accept and when to reject the investment. To support the value maximization proposition the firm the firm shall only undertake investments with positive return, i.e.

$$NPV = -I + \int_0^T \left[CF(t) e^{-kt} \right] dt > 0 \,. \qquad (3)$$

Equation 3 states that the firm shall only undertake a software investment if the Net Present Value (*NPV*) from software exceed the cost of the investment (*I*). If this criterion is met, the firm will act in accordance with its goal function. If NPV is 0 the firm shall be indifferent between undertaking and not undertaking the investment given the goal function of the firm.

3. DERIVING A SIMPLIFYING CATEGORIZATION OF CONSEQUENCES

In accordance with Renkema and Berghout (1997) a consequence is defined as an event that occurs because of the introduction of software. Since the event can represent a cost or a benefit, It is also assumed if a certain type of cost can be defined then the same type of benefit can also be defined and vice versa. Table 1 below summarizes the common categorizations of consequences from a software investment. Consequences in a parenthesis indicates that this type of consequences include the same type characteristics as the above mentioned consequence.

Common categorizations of consequences from a software investment	
Direct consequences	Indirect consequences (Human and organizational consequences) (Social subsystem consequences) (Hidden consequences)
Quantitative consequences	Qualitative consequences
Financial consequences	Non-financial consequences
Tangible consequences	Intangible consequences
Strategic consequence – Tactic consequences – Operational consequences	

Table 1. Common categorizations of consequences from a software investment

The distinction between direct and indirect consequences is made to differentiate the consequences that are directly related to a software investment and the consequences on the organization as well as the people within the organization, see Irani and Love (2001); Irani (2002). The distinction between direct and indirect consequences is related to whether the consequences are due to the implementation of software or not, see Love et al. (2005) for exemplification. These consequences share the same characteristics as quantitative and qualitative consequences discussed by Gunasekaran et al. (2001). This applies to social subsystem consequences discussed by Ryan and Harrison (2000) as well.

Whereas the above categorizations have all been made about the object affected by the consequence, the distinction between financial and non-financial consequences differentiate according to the activity inducing the consequence. Kusters and Renkema (1996) differentiate between consequences that can be directly traced to specific activities and those that cannot be directly traced. It is the casual relation that determines whether the consequence is financial or non-financial. A non-financial consequence thus represents a consequence for which we cannot determine the cash flow driving activity.

The distinction between tangible and intangible consequences is the most common distinction made, se e.g. Love et al. (2006); Borenstein and Betencourt (2005); Gunasekaran et al. (2001). An intangible consequence is a consequence that cannot be measured or valued and a tangible consequence is thus a consequence easily measured and valued according to Gunasekaran et al. (2001). The intangible consequences are often discussed as strategic consequences as in e.g. Gunasekaran et al. (2001); Irani and Love (2001) where consequences are categorized according to if they have a strategic, tactical or operational affect on the firm. Irani and Love (2001) further differentiate if the consequence is financial or non-financial. It is therefore no real difference between tangible consequences and direct consequences whereas the intangible consequences resemblance the qualitative consequences. For a more thorough discussion concerning different categorizations of consequences the reader is referred to Irani et al. (2006).

Although the above discussion is concerned with one type of consequences at the time, a consequence must be discussed using several of the categorizations to be fully understood. An intangible consequence can be direct and financial as an example of this. To overcome this, the simplified categorization introduced below will be used instead in deriving the cash flow function for a software investment.

3.1 A simplified categorization of consequences

None of the above categorizations are meant to be mutually exclusive since a consequence can be differentiated according to several of the different categorizations since they focus on different perspectives of the consequences and the firm that makes a software investment. In this paper a simpler and a more manageable categorization is motivated since it is directly related to the goal function of the firm. A question can be raised of why the categorization has to be linked to the goal function? The answer to this question is that since the goal function dictates the terms of existence for the firm we must know which consequences are relevant to take into account when assessing the economic return of a software investment. Given the NPV criterion in Equation 3, we need to know whether the consequences affect the cash flow from the investment.

To value the consequence, i.e. transform it into a cash flow, we need to be able to measure the consequence. The measuring do not have to done economically as long as we can assign a value on a different scale, e.g. frequencies, amount of time saved. The consequence can then be indirectly valued.

This gives the two types of unique consequences as presented in Table 1 below.

Consequence	Effect on the net economic benefit
Tangible at time *t*	Tangible at time *t*
Intangible at time *t*	Tangible At time *t+s*

Table 2. The unique feature of the different consequences and their effect on the net economic benefit

We differentiate between consequences that can be measured at time *t* and consequences that cannot be measured at time *t*. To be able to measure the consequence at time *t* we need to be able to determine an expected value of the consequence expressed in the time value of when the assessment of a software investment is made. If we can measure the consequence at time *t* we can also express as a cash flow from the software investment, i.e. the consequence can be valued at time *t*.

For the consequences that cannot be measured at time *t* we need to distinguish between two different types of consequences; the consequences that will never affect the cash flow from the software investment and the consequences that will affect the cash flow but at a later point in time, at time *t+s*. The time lag variable *s* is a random variable since we cannot tell *a priori* when the consequences will affect the cash flow. The first types of consequences can be ignored in the assessment model since they do not have an affect on the long term market value of the firm. The second type of consequences will affect the cash flow but we not know when or to what extent. Figure 1 below presents the simplified categorization the consequences from a software investment. The dashed arrows denote a binary relation, i.e. a choice between two paths, and solid arrows denote paths where no choice is given.

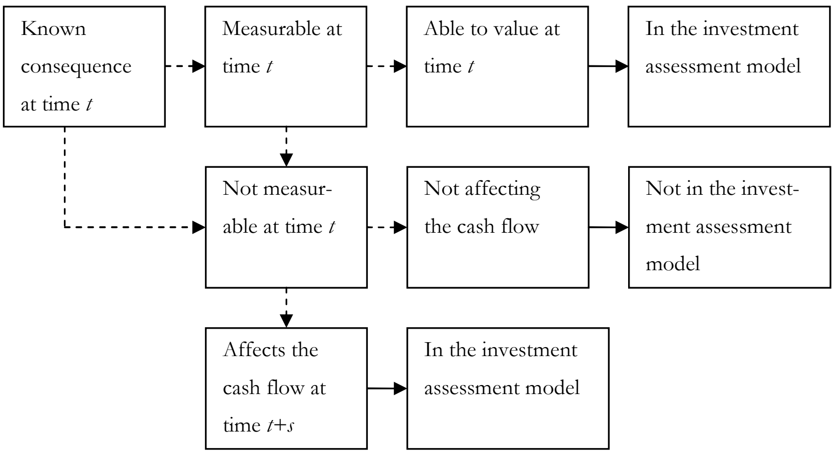

Figure 1. The simplified categorization of consequences from a software investment

To cope with consequences that are not measurable at time *t*, the cash flow function for the software investment must be able to handle the time lag and the lagged effect of these consequences, which is denoted by ε.

4. DERIVING THE CASH FLOW FUNCTION FOR A SOFTWARE INVESTMENT

To find the explicit behavior of the cash flow function for a software investment we need to find what drives the value for a software investment. In this paper we set the usage of software as an underlying process for the cash flow function, see Devaraj and Kohli (2003). The usage of software creates the consequences discussed above which have to be translated into cash flows. These cash flows are then to be used in the investment assessment model. In accordance with diffusion theory in IS/IT research, see e.g. Meade and Islam (2006) we expect the usage to increase over time, i.e. $E\lfloor \xi_{t_n} - \xi_{t_{n-i}} \rfloor > 0, \quad 0 \le t \le n \le i \le T$, where E denotes the expectation and ξ denotes the software usage. The usage is therefore viewed as a time-dependant variable, i.e. the amount of usage is dependant on the time that the software investment has been implemented. If the usage of the software investment is expected to increase over time, so is the cash flow from the investment, i.e. $E\lfloor CF_{t_n} - CF_{t_{n-i}} \rfloor > 0, \quad 0 \le t \le n \le i \le T$. The expected growth in the cash flow will govern by μ.

Given the above the cash flow function will be decomposed so it functional dependency can be expressed explicitly. Let X_t denote the consequences that can be measured and valued at time *t* and ε_{t+s} the effect of the consequences that cannot be measured at time *t* but affects the cash flow at time *t+s*. The cash flow function can then be written as a function of these two variables plus the growth factor, i.e. $CF = f\lfloor \mu(X_t, \varepsilon_{t+s}) \rfloor$. Since usage is the underlying function, we expect that both types of consequences will increase as the usage of software increases, and the same will therefore happen to the cash flow. Since the behavioral of ε is not known, the affect on the cash flow has to be imposed on the cash flow function. We do so by introducing the Brownian motion.

4.1 The cash flow function as a stochastic process

Given a probability space, there is a continuous process *B(t)* of $t \geq 0$ for each $w \in W$ that satisfies *B(0) = 0* and depends on *w*. The process *B(t)*, $t \geq 0$, is a Brownian motion if for all $0 = t_1 < \cdots < t_m$ the increments

$$B(t_1) = B(t_1) - B(t_0),\ B(t_2) - B(t_1), \cdots,\ B(t_m) - B(t_{m-1})$$

are independent and each of these increments are normally distributed with

$$\mathrm{E}\left[B\left(t_{i+1}\right) - B\left(t_i\right)\right] = 0 \qquad (4)$$

and

$$Var\left[B\left(t_{i+1}\right) - B\left(t_i\right)\right] = t_{i+1} - t_i \qquad (5).$$

The Brownian motion is the continuous time version of a scaled symmetric random walk process with the difference that it is normally distributed and does not exhibit linearity between time steps, i.e. the trajectory is nowhere linear and has no natural time step. This is referred to as the self-similarity of the Brownian motion. Intuitively, the self-similarity means that no matter how small part of a trajectory of a Brownian motion is magnified, it still exhibits the same irregularity as the entire trajectory.

An example of a trajectory of a symmetric random walk and a Brownian motion are plotted below for 5 years (T)[1] in Figure 2 and Figure 3. The random walk is generated with one increment per time period and for the Brownian motion 100 increments have been used. The expected value of each increment is 0 since the up and down ticks is equal in size, on average, due to the distribution and since they have the same probability. The variance, given by the above formula, is thus 1 for the random walk and $1/100$ for the Brownian motion.

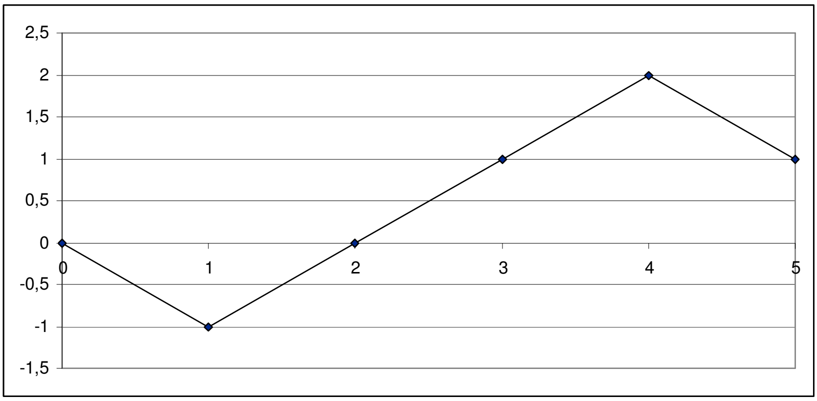

Figure 2. The trajectory of a symmetric random walk

In Figure 2 we can see the same type of behavior as for the characteristics of how ε has an effect on the cash flow from the software investment that was motivated above. Both examples show non-expected irregularities over time. The limitation of the symmetric random walk is that it is a discrete time process with only *1* increment per period of time. Since usage is assumed to be best approximated by a continuous process we will use the continuous time version of a scaled symmetric random walk instead, which is the Brownian motion seen in Figure 3

1 For methods on how to generate Brownian motions using different software the reader is referred to e.g. Black (2005); Stojanovic (2002).

below. The scaled symmetric random walk is generated by scaling down the size of the steps taken by the symmetric random walk while speeding up the time in the function, i.e. adding more increments.

We shall make one technical note on the differences between the scaled symmetric random walk and the Brownian motion. The trajectories shown in the figures represent merely an example of both. For the case of the random walk it is one case of a finite set of trajectories. This is not true for the Brownian motion since the probability space for a continuous process is infinite. If we were to generate two examples of both processes, chances are that the second trajectory of the Brownian motion would differ more that the second trajectory of the scaled symmetric random walk due to the fact that it is a continuous process.

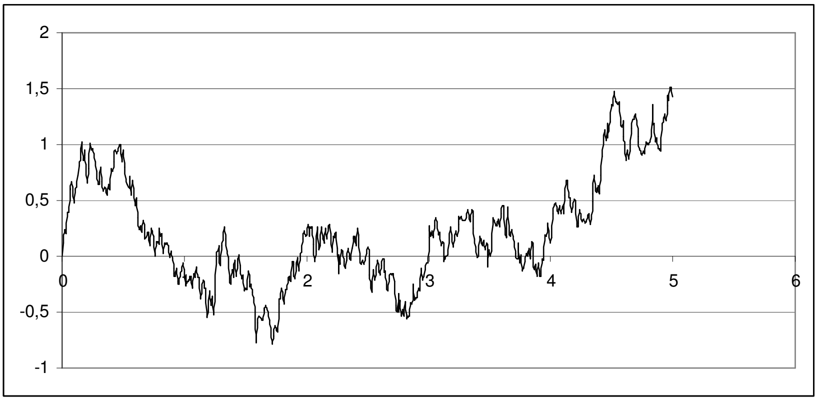

Figure 3. The trajectory of a Brownian motion with *T=5* and *N=500*

Due to being a continuous process that is generated by letting $N \to \infty$ the Brownian motion provides us with a greater flexibility concerning when the effect of the consequences that cannot be measured at time t can affect the cash flow. The trajectory in Figure 3 has a greater resemblance with the effect of ε on the cash flow from the software investment then the trajectory in Figure 2. The effect of these consequences with the time lag s can now be modeled as a stochastic process, i.e. a sequence of random variables, on the probability space.

The Brownian motion is defined over a probability space which shall be defined rather intuitively. A probability space consists of the triple (W, F, P) where W denotes the sample space which is a set describing all possible outcomes of the cash flow. In a continuous time setting the sample space is infinite. The variable F is the σ-algebra of subsets of W, $F \in W$, for which we assign measures, i.e. these are the events that are of interest for us. In continuous time F will play the role of a filtration process. Finally, P is a function assigning probability measures that assigns values from the set $[0,1]$, i.e. $0 \leq P(A) \leq 1$ to the event A which is a subset of F. We also have $\sum P = 1$, $P(W) = 1$, $P(O) = 0$, i.e. the sum of all probabilities equal 1 and the probability for the sample space is equal to 1 while the probability for the null set (denoted O) equals 0.

We will not go into more technical details here. The reader is referred to e.g. Aggoun & Elliot (2004); Jacod & Protter (2004); Papoulis & Pillai (2002); Shreve (2004) for a more complete discussion of probability theory and the underlying measure theory.

4.2 The explicit cash flow function for the software investment

Since the return is dependent on the consequences that can be measured and valued at time t and time as a variable for expected usage which we allow uncertainty in, we must develop some methods for calculating the return over time. If the case of a deterministic function we would use total differentiation for expressing the total dynamics of the return and the chain rule for the rate of change in the chain reaction. Since stochastic functions are unpredictable, rules of deterministic calculus cannot be used. Rules of calculus for stochastic processes must therefore be used. These rules will be introduced after the stochastic cash flow function is defined. Let the change, i.e. the derivative, of the cash flow function of the software investment be given by

$$dCF_t = \mu(X_t, t)dt + \sigma(X_t, t)dB_t \tag{6}$$

given $t \geq 0$. In the above function $\mu(X_t, t)$ is the drift of the process and $\sigma(X_t, t)$ represents the diffusion of the process. The function states that the consequences that can be measured and valued increase over time. Time is included as a measure of usage which will increase over time. The function takes into account the uncertainty in the consequences that can be measured and valued at time *t*, i.e. the time-independent consequences, as well as in the usage. The Brownian motion is included to capture the effect on the cash flow from the consequences that cannot be measured at time *t*, i.e. the time-dependent consequences. We will use this stochastic differential equation (SDE) to model the cash flow.

The SDE above is suitable for the purpose of this thesis since it differentiates the changes in the deterministic trend, i.e. the drift, from the uncertain effect of ε which is captured by the diffusion of the process. In this way the time-independent consequences are treated separately from the uncertain effects from the time-dependent consequences that have an effect on the cash flow with a time lag *s*.

4.3 The dynamics of the cash flow function

In order to express the dynamics of the above SDE we apply the stochastic counterpart of the chain rule, i.e. Ito's lemma, on the SDE. Under the assumption that the cash flow function is twice differentiable with respect to X_t and once with respect to *t*, the dynamics of the cash flow function is given by:

$$dCF = \frac{\partial CF}{\partial t} dt + \frac{\partial CF}{\partial X_t} dX + \frac{1}{2} \frac{\partial^2 CF}{\partial X_t^2} (dX^2) \tag{7}$$

where

$$(dX_t)^2 = (\mu_t dt + \sigma_t dB_t)^2 = \mu_t^2 (dt)^2 + 2\mu_t \sigma_t dt dB_t + \sigma_t^2 (dB_t)^2 = \sigma_t^2 dt$$

given

$$(dt)^2 = dt dB_t = 0, (dB_t)^2 = dt .$$

This is the equivalent to:

$$X(t) = X(0) + \int_0^t \mu(u) du + \int_0^t \sigma(u) dB(u) \qquad (8).$$

There is no formal difference between the integral form and the differentiated form of a SDE since the integral only sums up the increments in the differentiation. Since we have the explicit form of the function we can rewrite the dynamics from the Ito formula into the form of our SDE and get

$$dCF = \left[\frac{\partial CF}{\partial t} + \frac{\partial CF}{\partial X_t} \mu(X_t, t) + \frac{1}{2} \frac{\partial^2 CF}{\partial X_t^2} \sigma^2(X_t, t) \right] dt + \frac{\partial CF}{\partial X_t} \sigma(X_t, t) dB_t . \tag{9}$$

The reader is referred to e.g. Malliaris (1983); Merton (1992); Øksendal (2003); Protter (2005) for the derivation and a more thorough discussion of Ito's lemma and Protter & Jarrow (2004) for the history of stochastic calculus in finance.

Before solving the task of determining the return of the software investment we must define a martingale. We say that an adapted stochastic process is a martingale, $M(t),\, 0 \le t \le T$, given a probability space if

$$\mathrm{E}\big[M(t)\big|F(s)\big] = M(s) \tag{10}$$

for all $0 \le s \le t \le T$. This process has no tendency to rise or fall over time. If the process have a tendency to rise it is called it a submartingale and if the process have a tendency to fall it is referred to as a supermartingale.

Given the above we see that the Brownian motion is a martingale and so is our cash flow function if $\mu(X_t, t) = 0$, i.e. the cash flow function does not increase over time. Since we assume an increase in the cash flow function over time we need to transform the cash flow function into a martingale in order to find an explicit solution for return of the software investment. An intuitive interpretation of the above is to say that the martingale process maps the present value of the function. For a more thorough discussion of the above see e.g. Björk (2004); Karatzas & Shreve (2001; 2004); Musiela & Rutkowski (2005).

Given the above we can now develop the model which estimates the return of a software investment.

5. THE MODEL FOR ASSESSING THE ECONOMIC RETURN OF A SOFTWARE INVESTMENT

By applying Itos lemma, see e.g. Øksendal (2003); Shreve (2004), to explore the dynamics of the logarithm of Equation 6 we get the cash flow function for which the present value shall be calculated stated as

$$dCF = \left[\mu(X_t, t) - \frac{1}{2}\sigma^2(X_t, t) \right] dt + \sigma(X_t, t) dB_t \qquad (10)$$

in differential form. The expected value and the variance of Equation 10 are given by

$$E[CF_t] = CF_0 e^{\mu t} \qquad (11)$$

and

$$Var[CF_t] = CF_t^2 e^{2\mu t} \left(e^{\sigma^2 t} - 1 \right) \qquad (12)$$

with simplified notation, see Dixit and Pindyck (1994). By stating Equation 10 in integral form it becomes more obvious that it becomes the present value function for a software investment if a discount rate is added. Equation 10 can be written in integral form as

$$CF_t = CF_0 \exp\left\{ \int_0^t \sigma_s dB_s + \int_0^t \left(\mu_s - \frac{1}{2}\sigma_s^2 \right) ds \right\} \qquad (11)$$

with simplified notation where *exp* denotes the exponential *e*. The above formula shows how the continuous cash flows from the software investment are summed. In order to get the present value of the cash flows we introduce an adapted interest rate process *R*(*t*) and a discount process

$$D_t = \exp - \left\{ \int_0^t R_s ds \right\} \qquad (12)$$

with

$$dD_t = -\mathrm{R}_t D_t dt\,. \qquad (13)$$

The discounted cash flow function can now be stated as

$$D_t CF_t = CF_0 \exp\left\{\int_0^t \sigma_s dB_s + \int_0^t \left(\mu_s - \mathrm{R}_s - \frac{1}{2}\sigma_s^2\right) ds\right\} \qquad (14)$$

and in differential form as

$$d(D_t S_t) = \sigma_t D_t CF_t \left[k_t dt + dB_t\right] \qquad (15)$$

where k_t is the market risk used as discount factor from Equation 3, which is defined as

$$k_t = \frac{\mu_t - \mathrm{R}_t}{\sigma_t}\,. \qquad (16)$$

For Equation 15 to be the final solution to the analytic model we must explicitly determine k. This can be done by CAPM, see Sharpe (1964); Lintner (1965) and Mossin (1966). To use CAPM we need to predict the future return of the investment to be able to calculate the beta coefficient used in CAPM. In order to do so the return issue of a software investment is already solved. To overcome this problem we shall instead change the probability distribution for the future cash flows of a software investment. By doing so the cash flow function becomes a risk neutral process. Given a risk neutral process we can instead use a risk-free rate as a discount factor and still obey the law of one price.

5.1 The risk neutral solution

To change the cash flow process into a risk-free process, the Radon-Nikodým derivative, *Z*, is introduced. The Radon-Nikodým derivative is defined as:

$$Z = \frac{dQ}{dP}. \tag{17}$$

The Radon-Nikodým derivative derives the *Q*-probability measure, i.e. the risk neutral probability measure, from the *P*-probability measure for which the Brownian motion was defined. For it to be relevant to change measure form *P* to *Q*, the two measures must be equivalent, i.e. $Q(dZ) > 0$ if and only if $P(dZ) > 0$. To ensure this we impose that $Q(W) = \int_W Z(w)dP = EZ = 1$ and by rules of complement the zero set must have a zero probability under either measure. For *Z* to be useful we state the process that is follows as:

$$Z_t = \exp\left\{-\int_0^t k_u dB_u - \frac{1}{2}\int_0^t k_u^2 du\right\}, \tag{18}$$

and

$$\widetilde{B}_t = B_t + \int_0^t k_u du \tag{19}$$

given

$$E\int_0^T k_u Z^2 du < \infty. \tag{20}$$

The above is known as the Girsanov theorem, see e.g. Shreve (2004), which can be understood as Radon-Nikodým derivative for continuous time processes. Equation 20 is known as Novikov's condition which is imposed so that the variation of the Radon-Nikodým derivative stays finite. Equation 16 defines a new Brownian motion which is defined under the *Q*-probability measure. The point of using the Girsanov thorem is to change the mean of the process while leaving the uncertainty structure, i.e. the variance, unchanged. In that way the risk-free rate of return can be used as the discount factor instead of the market risk calculated by CAPM.

The new stochastic process for Z_t is a martingale that we will use to change the mean of the cash flow function for the software investment while leaving the variance structure unchanged, i.e. we subtract the risk premium and make the process risk-free. We see that the process of Z_t will do so by observing the way that k_t is a part of the process. The last formula above (the Novikov condition) is imposed so that the Radon-Nikodým derivative process does not vary too much i.e. stays finite.

The process $\widetilde{B}_t$ is a new Brownian motion which is a martingale defined over a new probability space (*Q, F, W*) instead of (*P, F, W*). This new probability measure *Q* is a risk-neutral probability measure and the process thus becomes a martingale. The reason why we will use the risk-neutral measure is that we can then use the risk-free rate of return as a discount factor and thereby we do not have to determine the "correct" risk premium for the investment.

Given the above we can now state the final analytic model for assessing the economic return of a software investment. We do so by first by re-writing Equation 11, given the Q-probability measure, as

$$D_t CF_t = CF_0 + \int_0^t \sigma_u D_u CF_u d\widetilde{B}_u \ . \qquad (21)$$

The cash flow function for the software investment has now a mean rate of return that equals the interest rate process under *Q*. This can be verified by substituting $dW_t = -kdt + d\widetilde{B}t$ into Equation 12 which then becomes

$$dCF_t = \mathrm{R}_t CF_t dt + \sigma_t CF_t d\widetilde{B}t \,. \tag{22}$$

We can now solve for the analytic solution for the economic return of a software investment. We do so by first replacing the integrals in Equation 8 by their equivalents under the *Q*-probability measure, i.e. $\int_0^t \sigma_s d\widetilde{B}_s - \int_0^t (\mu_s - \mathrm{R}_s) ds$. The final step is to subtract the investment (*I*) from the function and by doing so we get

$$CF_t = CF_0 \exp\left\{ \int_0^t \sigma_s d\widetilde{B}_s + \int_0^t \left(\mathrm{R}_s - \frac{1}{2}\sigma_s^2 \right) ds \right\} - I_0 \,. \tag{23}$$

Equation (23) is the NPV function that by which the economic return of a software investment is assessed. For more about the used approach, see e.g. Shreve (2004).

6. CONCLUSIONS

This paper it has shown how the economic return of a software investment can be assessed. It has shown how the cash flow function for a discounted cash flow model can be developed and implemented. The discounted cash flow model is chosen since the value maximization proposition is seen as the goal function of the firm and the NPV is aligned with the value maximization proposition.

The assumption underlying the approach used in this paper is that usage leads to consequences which have to be transformed into cash flows. One class of consequences (X_t) does not pose any problems since we can measure them and value them, i.e. transform them into cash flows when they occur. The second class of consequences ε is more problematic to incorporate using traditional (naive) DCF models, i.e. DCF models based on deterministic cash flow functions, due to their characteristics. They are argued to be non-measurable at time t and therefore we cannot value them at the same time t, i.e. the do not affect the cash flow at that time period. It is argued that the effect of ε on the cash flow is lagged with s periods, where s is a random variable.

To incorporate ε in the cash flow function a Brownian motion is used to impose the characteristics of ε. The final step in deriving Equation 23 is the change of probability measure on which a new Brownian motion is defined to govern the effect of ε on the cash flows. By doing so the risk-free rate of return can be incorporated into the cash flow function as a discount factor and the economic return of a software investment can be assessed.

6.1 Suggestions for further research

Little is known the realization of the cash flow from different types of software investments. Do all types of software investments produce consequences that fit the categorization made in section 3? Can all consequences be mapped into cash flows using the developed model in this paper? Although being a categorization on a general level there ought to be differences between the characteristics in cash flows from different types of software investments.

REFERENCES

Anandarajan, A. and Wen, H. J. (1999) "Evaluation of Information Technology Investment", *Management Decision*, Vol. 37, No. 4, pp. 329-337

Bacon, C. J. (1994) "Why Companies Invest in Information Technology" in Willcocks, L. (Ed.) (1994) "Information Management: The evaluation of Information Technology Investments", *Chapman Hall*, London

Borenstein, D and Betencourt, P. R. B. (2005) "A Multi-Criteria Model for the Justification of IT Investments", *Infor*, Vol. 43, No. 1, pp. 1-21

Bellatine, J. and Stray, S. (1998) "Financial Appraisal and the IS/IT Investment Decision Making Process", *The Journal of Information Technology*, Vol. 13, No. 1, pp. 3-14

Bellatine, J. and Stray, S. (1999) "Information System and Other Capital Investments: Evaluation Practice Compared", *Logistics Information Management*, Vol. 12 No. 1-2, pp. 78-93

Brealey, R. and Myers, S. (2003) "Corporate Finance", *McGraw-Hill*, New York

Devaraj, S., Kohli, R. (2003) "Performance Impacts of Information Technology: Is Actual Usage the Missing Link?", *Management Science*. Vol. 49, No. 3, pp. 273-289

Dixit, A. K, Pindyck, R. S. (1994) "Investment Under Uncertainty", *Princeton University Press*, Princeton, New Jersey

Fichman, R. G. and Kemerer, C.F. (1999) "The Illusory Diffusion of Innovation: An Examination of Assimilation Gaps", *Information Systems Research*, Vol. 10, No. 3, pp. 255-275

Gunasekaran, A., Love, P. E. D., Rahimi, F. and Miele, R (2001) "A Model for Investment Justification in Information Technology Projects", *International Journal of Information Management*, Vol. 21, No. 5, pp. 349-364

Hallikainen, P, Kivijärvi, H and Nurmimäki, K. (2002) "Evaluating Strategic IT Investment: An Assessment of Investment Alternatives for a Web Management System", *Proceedings of the 35th Hawaii International Conference on System Science*

Irani, Z. and Love, P. E. D. (2001) "The Propagation of Technology Management Taxonomies for Evaluating Investments in Information Systems", *Journal of Management Information Systems*, Vol. 17, No. 3, pp. 161-177

Irani, Z. (2002) "Information System Evaluation: Navigating Through the Problem Domain", *Information & Management*, Vol. 40, No.1, pp. 11-24

Jensen, M. C. (2001) "Value Maximization, Stakeholder Theory, and the Corporate Objective Function", *European Financial Management*, Vol. 7, No. 3, pp. 297-317

Hitt, L. M. and Brynjolfsson, E. (1996) "Productivity, Business Profitability, and Consumer Surplus: Three Different Measures of Technology Value", *MIS Quarterly*, Jun, Vol. 20, No. 2, pp. 121-143

Kusters, R. J and Renkema, T. J. W (1996), *Proceedings of the third European Conference for IT Evaluation*, Bath University School of Management, Bath University, Bath, pp. 133-141

Lee, S. and Kim, S. H. (2006) "A Lag Effect of IT Investments on Firm Performance", *Information Resources Management Journal*, Vol. 19, No. 1, pp. 43- 69

Lefley, F. (1994) "Capital Investment Appraisal of Manufacturing Technology", *International Journal of Production Research*, Vol. 32, No. 12, pp. 2751-2776

Lefley, F. (1996) "Strategic Methodologies of Investment Appraisal of AMT Projects: A review and Synthesis", *The Engineering Economist*, Vol. 41, No. 4, pp. 345-363

Love, P. E., Irani, Z., Standing, C., Lin, C. and Burn, J. M. (2005) "The Enigma of Evaluation: Benefits, Costs and Risks of IT in Australian Small-Medium-Sized Enterprises", *Information and Management*, Vol. 42, No. 7, pp. 947-964

Love, P. E. D., Irani, Z., Standing, C., Lin, C. and Burn, J. M. (2006) "The Enigma of Evaluation: Benefits, Costs and Risks of IT in Australian Small-Medium-Sized Enterprises", *Information & Management*, Vol. 42, No. 7, pp. 947-964

Lubbe, S. and Remenyi, D. (1999) "Management of Information Technology Evaluation – The Development of a Managerial Thesis", *Logistics Information Management*, Vol. 12, No. 1/2, pp. 145-156

Meade, N., Islam, T. (2006) "Modelling and Forecasting the Diffusion of Innovation – A 25-year Review", *International Journal of Forecasting*, Vol. 22, No. 3, pp. 519-545

Milis, K. and Mercken, R. (2004) "The Use of Balanced Scorecard for the Evaluation of Information and Communication Technology Projects", *International Journal of Project Management*, Vol. 22, No. 2, pp. 87-97

Mossin, J. (1966) "Equilibrium in a Capital Asset Market", *Econometrica*, Vol. 34, No. 4, pp. 768-783

Lintner, J. (1965) "Security Prices, Risk, and Maximal Gains from Diversification", *Journal of Finance*, Vol. 20, No. 4, pp. 587-615

Renkema, Th. J. W and Berghout, E.W. (1997) "Methodologies for Information Systems Evaluation ant the Proposal Stage: A Comparative Study", *Information and Software Technology*, Vol. 39, No.1, pp. 1-13

Ross, S. A., Westerfield, R. and Jaffey, J. (2007) "Corporate Finance", *McGraw-Hill*, New York

Ryan, S. D., Harrison, D. A. (2000) "Considering Social Subsystem Costs and Benefits in Information Technology Investment Decisions: A View from the Field on Anticipated Payoffs", *Journal of Management Information Systems*, Vol. 16, No, 4, pp. 11-40

Sharpe, W. F. (1964) "Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk", *Journal of Finance*, Vol. 19, No. 3. pp. 425-442

Shreve. S. E. (2004) "Stochastic Calculus for Finance II – Continuous-Time Models", *Springer Science*, New York

Small, M. H. and Chen, J. (1995) "Investment Justification of Advanced Manufacturing Technology: An Empirical Analysis", *Journal of Engineering and Technology Management*, Vol. 12, No. 1-2, pp. 27-55

Wilcocks, L. and Lester, S. (1994) "Evaluating the Feasibility of Information Technology Investments: Recent UK Evidence and New Approaches" in: Wilcocks, L. (Ed.) (1994) "Information Management: The Evaluation of Information Technology Investments", *Chapman & Hall*, New York

Øksendal, B. K. (2003) "Stochastic Differential Equations – An Introduction with Applications", *Springer-Verlag*, Berlin, Heidelberg

CHAPTER 5
PAPER 2

Based on Numminen, E. (2010) "Hedging Uncertainty in Software Development – A Real Option Modeling Approach", Presented at the 24th European Conference for Operations Research in Lisbon, Portugal, unpublished

HEDGING UNCERTAINTY IN SOFTWARE DEVELOPMENT – A REAL OPTION MODELING APPROACH

Emil Numminen
School of Management at Blekinge Institute of Technology, Ronneby, Sweden
Emil.Numminen@bth.se

ABSTRACT

This paper shows how real option theory is useful to manage uncertainties in software development projects. The paper show how the option to defer, the option to scale up and the option to abandon create flexibility and add value while decreasing the project uncertainty. The numerical modeling in this paper is done in a two-period discrete time setting using equivalent martingale measures to enforce a risk-neutral expectation. The

cash flow uncertainty is modeled via a multiplicative binomial process. The analysis is concluded with an empirical validation of the circumstances under which these options can be created in most software development projects. The paper ends with suggestions for further research.

KEYWORDS

Software Development, Uncertainty, Real Option

1. INTRODUCTION

The valuation of a software investment has been argued to be an difficult task (e.g. Irani et al., 2007; Borenstein and Betencort, 2005; Milis and Mercken, 2004; Serafemidis and Smithson, 2000). To a large part this problem is argued to stem from the fact that software investment is inherently more uncertain than traditional investments (e.g. Fichman, 2004; Halikainen et al., 2002).

As a way to overcome the difficulty of software investment valuation, a great deal of attention has been devoted to real option theory, see below for literature review. Most of the papers that constitute this research apply continuous-time models to analyze the value of an option in the software investment. This paper adds to that research developing a discrete-time model that is based on fewer assumptions to show the added value form the three most arbitrarily real options. The paper also discusses how these real options can be incorporated in more or less all software investment projects.

The remainder of the paper is organized as follows. Section 2 introduces real option theory. Section 3 discusses how real option theory has been used previously in software economics. Section 4 develops the model and shows the value added by the three options to a software investment decision. Section 5 draws the conclusion of this paper.

2. A BACKGROUND TO REAL OPTIONS

A real option can according to e.g. Amran and Kulatilaka (1999); Dixit and Pindyck (1994); Copeland and Antikarov (2003) be defined as the right to undertake an action for the future at a predetermined price for a predetermined period of time. From a valuation or point of view there is a great difference between static e.g. discounted cash flow (DCF) based models such as the net present value (NPV) calculation and dynamic valuation model. The difference is that the latter types of models can take into account the managerial flexibility in the sense that decisions can be altered which results in added economic value. This managerial flexibility is found in the possibility to adapt and revise previously made decisions when additional information becomes accessible. This results in reduced uncertainty being incorporated into the chain of decisions. The value of the investment therefore changes from value calculated by traditional NPV,

$$NPV = -I + \sum_{t=0}^{T} \frac{E(CF_t)}{(1+k)^t} \qquad (1)$$

to

$$ENPV = NPV + O\,, \qquad (2)$$

which will be denoted the expanded NPV (Trigeorgis, 1996) and where O denotes the extra value that the new opportunities, i.e. real options, add. In the NPV calculation, Equation 1, I denote the cost for the initial, CF denotes the cash flow that the firm receives at time t and k denotes the opportunity cost of capital. E is the expectation operator that takes into account the uncertainty of the future cash flows when it is assigned values below one. If there is no uncertainty about the future cash flows $E = 1$ and Equation 1 is the traditional NPV criterion. More formally E is the probability measure for the state that yields the cash flow and is govern by $0 < p_i < 1, \sum_i p_i = 1$ where p denotes a probability measure.

When E is assigned values below one as required above, Equation 1 assesses the investment value using decision three analysis (DTA), see e.g. De Reyck et al. (2008) or Smith and Nau (1995) for a discussion of the differences between NPV and DTA. Equation 1 is subjected to the constraint $MAX[NPV,0]$, which makes sure that decision maker acts according to the value maximization proposition as defined in e.g. Jensen (2001), i.e. only undertakes investments that creates more money than is consumed by the investment. This constraint is also valid for the expanded NPV decision for obvious reasons, i.e. Equation 2.

To benefit from the extra value added by real options there are three underlying assumptions that must hold true for the decision setting according to Dixit and Pindyck (1994; 1995a; 1995b). Firstly, we must be able to defer the investment if market conditions require so, i.e. it cannot be a now or a never investment decision. This assumption would not be controversial for most investments since most investments can be timed by the firm and started when most appropriate given a time interval. This would hold true for most software development projects.

Secondly, the investment must be fully, or at least partly, irreversible, i.e. it must involve a sunk cost. This assumption is likely to be true for most types of investments since if a project is abandoned or sold, the salvage value is not likely to cover the total cost spent on the project or the cost of purchase due to depreciation, in a wider sense, of the product. This assumption implies that if it is a now or never type of investment then the investment must be fully reversible for the NPV criterion to make sense. Given that this study deals with software investments this is not a valid assumption since the secondary market for software is not an efficient market and we cannot capitalize the total value of the software.

Thirdly, there must be an uncertainty regarding one or several of the variables that determines the value of the investment, e.g. output prices or input costs. This assumption can be seen as a consequence of the assumption made about the world and more concretely implies that we do not posses full knowledge about the future. This assumption is valid for software investments as seen later in this paper.

2.1 Option terminology

As indicated by the term "option" it provides the right to undertake the action but not the obligation to do so. Given the definition of a real option it becomes clear that a real option plays the same role for real investments as the more known financial option does for financial investments. A financial call option gives given the holder the right to by the underlying asset at a predetermined price, the exercise price (*K*), whereas a financial put option gives the holder the right to sell the asset for the exercise price. If the option is an European style the transaction takes place at a predetermined date, i.e. the exercise date and if it is an American style option the transaction can take place up until the exercise date.

Since the option provides the flexibility of whether to exercise the option or not we will only exercise if the option is in the money. In the case of a call option this means that the spot price (*S*) of the underlying assets exceeds the exercise price of the option. For a put option we have the opposite relationship, i.e. the *K* should exceed *S* for option to be exercised. If we have the opposite relationship for the call and the put options the option is out of money and should not be exercised. Option pricing, i.e.

valuation, is therefore based on the optimal choice where we prefer more over less. The boundary condition for the value option value can be more formally stated as

$$MAX[S-K,0] \tag{3}$$

for a call option and as

$$MAX[K-S,0] \tag{4}$$

for a put option. The valuation of the option is based on a MAX-formula to capture the managerial flexibility. The reader is referred to Hull (2008) or McDonalds (2005) for a more thorough discussion concerning financial option pricing.

2.2 Different types of real options

Since real options define opportunities for the future we can distinguish between different types real options depending on the type of situation we analyze and the type of uncertainty we are trying to hedge. What is common to them all is that they can be defined with the help of their American financial counterparts. The following types of real options are discussed by Copeland & Antikarov (2003):

- Deferral option
  The option is the right to delay the undertaking of the investment, i.e. an American call option.
- Option to abandon
  The option to abandon the project at a predetermined price, i.e. an American put option.
- Option to contract (scale back option)

An option to sell of a part of the project, i.e. an American put option. The option to scale back can be viewed as special kind of a option to abandon where only a part of the of underlying investment is abandoned.

- Option to expand

  An option to increase the scale of the project, i.e. an American call option. This type of option is also referred to as a growth option.

- Option to extend

  An option to extend the life-span of the project, i.e. an American call option.

- Switching option

  A portfolio of call and put options depending on the type of strategies for the project that can be switched among at a predetermined price, i.e. input or output is changed for a predetermined price.

- Compound option

  Options on options, the existence of a sequential option is dependent on exercising a previous option, i.e. American call and put options depending on the underlying option.

- Rainbow option

  An option which value is driven by several sources of uncertainty.

- Compound rainbow option

  Options on options that derives their values from several sources of uncertainty.

In the above description the reference has been given to American options because it is more realistic that the decision must not be made on a single point in time but instead during a fixed interval of time. For an in-depth discussion about the seminal papers behind the different types of real options the reader is referred to Schwarz and Trigeorgis (2004) or

Trigeorgis (1996). The below section intends only to give a short introduction to the history of real options.

The first time the term real option occurred in a paper was 1977 when Stewart Myers discussed growth options in Myers (1977) as mean to describe the difficulty to use DCF based models to assess the value of sequential investments where the positive value comes from the subsequent investment. DCF based models do not capture this flexibility in the decision setting since they were developed to value static financial assets, e.g. fixed income assets and residual income assets. A similar analysis was also made by Kester (1984). Although these papers formalized the critique of DCF based models and argued for a different view on real investments due to the embedded options in many investments the term options and its relation to business owners was already discussed in Fisher (1930).

2.3 What drives the value of a real option?

The following six variables and their impact on the value of a real option are discussed by Copeland et al. (2000).

- The value of the underlying asset, i.e. the real investment
  Given the two boundary conditions for the call and put value above it is clear that the value of call based option is positively correlated with the value of the underlying investment. For put based option the opposite relationship is true.
- The exercise price
  Given the two boundary conditions for the call and put value above the vale of a call based option is negatively correlated with the exercise price. For put based options the opposite relationship holds true.
- The time to expiration

As the time to expiration increases so does the value of the option regardless of whether it is a call or a put based option. This is because the time for entering the in-the-money zone increases and therefore also the value of the option.

- The uncertainty of the value of the underlying investment

  Both call and put based options values are positively correlated with uncertainty of the underlying asset since it provides a setting for managerial flexibility. The uncertainty referred to can be of two general types; inner and outer. Inner uncertainty relates to the technical uncertainty in the production of the goods or services and therefore drives the cost uncertainty. Outer uncertainty denotes the market uncertainty concerning the demand and prices that the goods and services are (expected to be) sold for. In most types of real options we only allow for one specific source of uncertainty but when valuing rainbow options we explicitly take into account the different sources of uncertainty and their effect on the value of the investment.

- The risk-free rate of return

  Call based option will be positively correlated with the risk-free rate of return. This is because the risk-free rate of return is underlying all returns and if it increases, the compounding of the funds needed to finance the exercise price increases with an increasing risk-free rate of return. For a put based option the relation becomes opposite, with the same line of reasoning.

- Dividends paid by the underlying asset

  Call based options will be negatively correlated with cash flows from the underlying investment. This is can be understood given the definition and usage of real options; if the option is not called, we do not own the underlying investment and therefore do not receive the cash flows. In the case of put based options the relation is the opposite since we receive the cash flows before we sell of parts of the investment or the entire investment.

3. REAL OPTIONS IN THE SOFTWARE ECONOMICS LITERATURE

Real option techniques have been used for some time now in the software economics literature and were introduced given the same critique of standard DCF based models in the corporate finance literature. However theses DCF based models are not widely used by companies when software investments are valued, see e.g. Klecun and Cornford (2005); Milis and Mercken (2004); Irani (2002); Ballantine and Stray (1998) for literature reviews and empirical evidence on the topic, and Walter and Spitta (2004) for critical review of *a priori* valuation techniques.

The use of DCF based models have been argued to less suitable, or even not suitable at al, for software investments due to the characteristics from these investments, see discussion and literature review in McGrath (1997); Lefley (1994) and see Powell (1992); Ashford et al. (1988) for an opposite view of the issue. The consequences form a software investment are said to be more uncertain than the consequences an "ordinary" investment in the sense to what extent and when they have an effect on the cash flow, see e.g. Numminen (2008a; 2008b) for a more thorough discussion about the consequences from a software investment and their effect on the cash flow.

Real options have been used in the software economics literature as one way to overcome the above problems and as a way to include flexibility in the development of software to decrease the uncertainty in the projects. Early works include Clemmons (1991), that discusses strategic opportunities in software investments, Dos Santos (1991) discusses embedded options in software for future investments based on the same technology, i.e. compound options, using Magrabe's (1978) model. Kambil et al. (1993) illustrate how different real options create value by future growth or cost saving by studying whether or not a pilot project shall be undertaken using Cox et al's (1979) model. Common for the latter two papers is that they study the value dependency between a pilot project and a follow-up project that is created by exercising the embedded option. This methodology is also proposed by Chalasani et al. (1998) and Erdogmus (2002) when it comes to software development decisions instead of Bayesian decision theory, as pioneered by Boehm (1984), for settings when the prototyping decision can be deferred.

Other researchers have looked at how different models for real option valuation can be used to assess the value of IT and software investments. Benaroch and Kauffman (1999; 2000) studied how Black and Scholes (1973) model can be used to assess the flexibility value of a investment in a electronic banking system and how to manage the uncertainty in the investments. Panayi and Trigerorgis (1998) used the same model to analyze a two-stage decision in a telecom investment in Cyprus and modeled the growth option as an European call option. Shcwartz and Zozaya-Gorostiza (2000; 2003) used Pindyck's (1993) model and the model developed by Schwartz and Moon (2000) to evaluate under what circumstances a software shall be developed and when it shall be acquired. Benaroch (2002) and Benaroch et al. (2006) studied what types of real options to plant into software development project to manage different types of risks in software development.

Other problems that have been studied is the inter-uncertainties between two projects and value of exchange one project for another. Kumar (1996; 1999) used Magrabe's (1978) model to evaluate the option to exchange an uncertain cash flow for another uncertain cash flow by studying the value of a decision support system in several settings.

Other studies in software economics using real options includes Favaro et al. (1998) where the strategic value to reuse software is studied based on contingent claim analysis, i.e. option analysis, to show the value of flexibility. Taudes (1998) and Taudes et al. (2000) studied how software platforms provide growth options for future applications and thus incorporating flexibility in the value creating process. A similar problem is studied in Fichman (2004) where a model is developed to analyze when a firm shall take the leading role in emergent technologies, i.e. software platforms.

The above discussion of the previous research of real options in software economics is provided to give a brief introduction to the different ways real options have been used in previous software economics literature. The main point is that due to the economic characteristics of a software investment, a real option approach to the investment valuation has been able to handle the present uncertainties, by incorporating flexibility, better than traditional static approaches. For more comprehensive literature reviews the reader is referred to e.g. Benaroch et al. (2005); or Fichman (2004).

4. HOW FLEXIBILITY ADDS VALUE IN A SOFTWARE INVESTMENT

This section will show how the three most basic real options add value to a software development project as well as how they can be used to manage different uncertainties for the software development project. The common feature for all three of them is that they can, more or less, easily be planted into most software developments projects. We will also show the resulting asymmetry in the payout of the option and why the NPV rule does not lead to optimal decisions under these circumstances.

Assume that a firm develops a software that will in one year either provide a cash flow of 90 million (CF^{+}) or 30 million (CF^{-}) depending on the market situation. Assume further that in two years the cash flow from the software will grow by 80% into 162 million (CF^{++}) or further decrease by 60% to 18 million (CF^{--}). The cash flow function is thus a stochastic process following a multiplicative binomial process with $CF_{t=0} = 50$ that moves up with $u = 1.8$ and down with $d = 0.6$ given a risk-free rate of return of 8% (*r*). Should the firm make the investment if the software costs 103 millions (*I*) to develop given that the risk-adjusted cost of capital is 20% (*k*), i.e.

$k = \frac{E_{t=0}(CF_{t=1})}{CF_{t=0}} - 1 = \frac{0.5 \times 90 + 0.5 \times 30}{50} - 1 = 0.2$, and the probability for up (*p*) and down movement (*1-p*) in the economy is 0.5? See Figure 1 below for the complete graphical illustration of the cash flows for this decision setting.

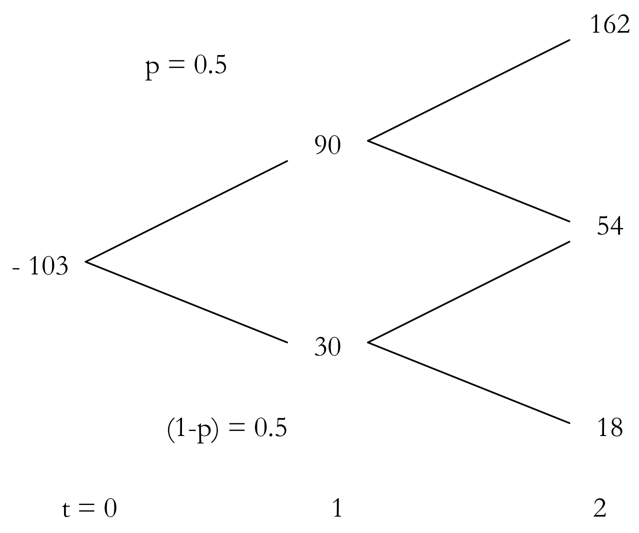

Figure 1. General cash flow setting for the software investment decision

If behaving according to the static NPV criterion, i.e. Equation 1, the value of the software project is

$$
\begin{aligned}
NPV &= -I + \frac{pCF^{+} + (1-p)CF^{-}}{(1+k)} + \frac{p^2 CF^{++} + (1-p)^2 CF^{--} + 2p(1-p)CF^{+-}}{(1+k)^2} \\
&= -103 + \frac{0.5 \times 90 + 0.5 \times 30}{1.2} + \frac{0.5^2 \times 162 + 0.5^2 \times 18 + 2 \times 0.5^2 \times 54}{1.2^2} \\
&= -103 + 50 + 50 = -3
\end{aligned}
$$

and the firm should not proceed with the investment since the expected present value of the future cash flows is less than the cost for the development of the software. By going forward with the investment anyway would result in erosion of the value of the firm and is not inline with the value maximization proposition, i.e. the goal function of the firm.

In the above example we have used the marketed asset disclaimer (MAD) in the calculation of k instead of the a market-based replicating portfolio in accordance to the arguments given by e.g. Copeland and Antikarov (2003).

4.1 The option to defer the investment

The above calculations are based on the assumption that the firm can make the investment now or never. This may not always be the case. If market conditions, e.g. competition, copyrights, etc, may allow the firm to wait until the market uncertainty has been reduced before the investment is made. Instead of making the investment at $t = 0$, the firm can choose to wait one year before developing and releasing the software and by that base the decision on the realized information of the market condition. The question is whether this would make a difference for the value of the software development project.

To simplify the valuation of the new mutually exclusive opportunity we are going to introduce risk-neutral valuation. The reason for doing so is that by introducing the option to the decision setting we change the cash flows yielded given the investment. The cash flows now becomes asymmetrical and require that *k* is not a general measure for the opportunity cost of capital but must be determined for each node in the decision tree that is effected by the option. Introducing risk-neutral valuation makes the valuation more general and allows us to use the risk-free rate of return as a discount rate, which is not determined by the uncertainty inherent in the individual investment. The risk-neutral probabilities are determined by Equation 5 below,

$$q = \frac{(1+r)-d}{u-d} \tag{5}$$

given $d < (1+r) < u$ for an up movement and $(1-q)$ for a down movement, see e.g. Copeland and Antikarov (2003) or Trigeorgis (1996) for a derivation of Equation 5 or the seminal papers by Harrison and Kreps (1979); Harrison and Pliska (1981) for a more technical discussion on the mater. This gives us $q = \frac{(1+0.08)-0.6}{1.8-0.6} = 0.4$ and $(1-q) = 1-0.4 = 0.6$ for the present setting.

Note that the value of the investment is the same under the two measures given the same setting, i.e.

$$\begin{aligned} NPV &= -I + \frac{qCF^{+} + (1-q)CF^{-}}{(1+r)} + \frac{q^2 CF^{++} + (1-q)^2 CF^{--} + 2q(1-q)CF^{+-}}{(1+r)^2} \\ &= -103 + \frac{0.4\times 90 + 0.6\times 30}{1.08} + \frac{0.4^2\times 162 + 0.6^2\times 18 + 2\times 0.4\times 0.6\times 54}{1.8^2} \\ &= -103 + 50 + 50 = -3. \end{aligned}$$

since the risk-neutral probability for higher cash flows are lower whereas the probability for lower cash flows are higher. It must be stated, however, that risk-neutral probabilities are not a measure for the degree of belief that an event will occur but simply mathematical measures that allows risk-free valuation. Risk-neutral probabilities simplify the valuation since fewer variables must be estimated empirically.

We can now value the software project when we have the option to defer the investment one year. If investment is not made at $t=0$ but instead at $t=1$ we can instead deposit the I for a year in a bank account earning the risk-free rate of return and therefore the true cost of the investment at $t=1$ is $103 \times 1.08 = 111.24$ in a risk-free world. The cost for making the investment at $t=1$ will be denoted I_D. The setting for the new investment decision can be depicted as shown in Figure 2 on the following page.

The difference between this valuation setting and the original setting is that there may occur a negative cash flow at $t=1$ and that we are using the risk-neutral probabilities and thereby the risk-free rate of return as a discount factor.

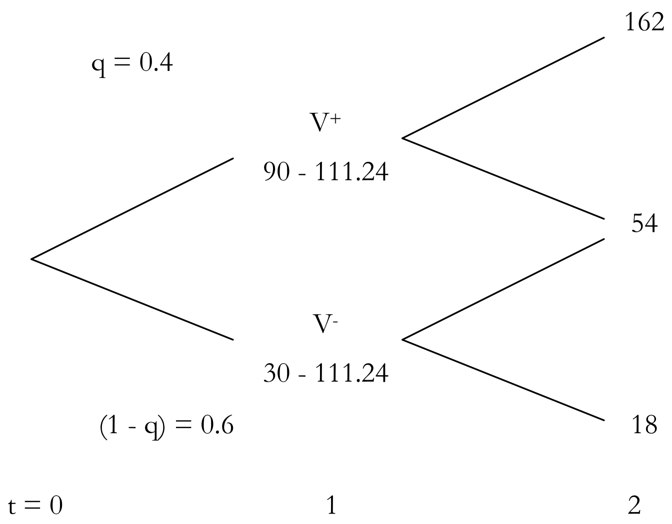

Figure 2. The cash flow setting for the option to defer

The keyword in the above paragraph is *may* since the firm has the option to take the action but not the obligation. The firm will therefore only exercise the option if it is in line with the value maximization proposition. To value the decision lets call the value of the first up state V^+ and the value of the first down state V^-, and let $^+$ and $^-$ denote up and down state generally. Then,

$$ENPV^+ = MAX(V^+ - I_D, 0) \tag{6}$$

and

$$ENPV^- = MAX(V^- - I_D, 0). \tag{7}$$

The intuition behind Equation 6 and 7 is that the firm will only undertake the development if the responding value is greater, if not; the firm chooses not to exercise the option and get 0 in cash flows from the option. This makes sense according to the value maximization proposition since the value of the node consists of the cash flow of the period plus the present value at $t = 1$ of the expected future cash flow, i.e.

$$ENPV^{+} = MAX\left[90 + (0.4 \times 162 + 0.6 \times 54)/1.08 - 111.24,\ 0\right] = 68.76$$

and

$$ENPV^{-} = MAX\left[30 + (0.4 \times 54 + 0.6 \times 18)/1.08 - 111.24,\ 0\right] = 0 .$$

The value of the investment at $t = 0$ is then

$$ENPV = \frac{qENPV^{+} + (1-q)ENPV^{-}}{1+r} = \frac{0.4 \times 68.76 + 0.6 \times 0}{1.08} = 25.47$$

and the value created by the flexibility in the deferral option is worth according to Equation 3.2

$$O = ENPV - NPV = 25.47 - (-3) = 28.47 .$$

The numerical example above shows that the value of the investment increases substantially if we take into the ability to time the investment. The ability to time the investment turned an investment with a -\$3 million NPV into an investment worth undertaking, *ENPV = \$25.47* million, when the market uncertainty was reduced by deferring the development one year. In the above case, the option premium accounted for more than 27% of the investments gross value. The flexibility to defer a software investment could be planted in most software development projects where there is no contractual release date, or when breaking the contract is more beneficial due to value from the option. There could exist market conditions that make planting options pointless, e.g. no future uncertainty over critical variables or first mover advantages. The reader is referred to e.g. Shapiro and Varian (1999) for a more thorough discussion on the latter point.

4.2 The option to expand

Assume that the firm goes ahead with the investment based on the static NPV calculation anyway although the negative NPV as indicated by the software economics literature on evaluation, see e.g. Irani (2002) or Small and Chen (1995) for discussion of this. Could this be justified from an economic stand point of view?

Assume that the firm can choose to invest an additional $50 million ($I_E$) to develop complementary software in $t = 1$ that reuses vital components from the initial developed software. The development of the new software will influence the cash flow for the firm by doubling the cash flow in $t = 2$ but will otherwise, apart from the additional investment, resemble the original decision setting, i.e. the static NPV case. The additional investment requires that the initial investment is made can be viewed as an sequential investment in the form of a growth option, i.e. an option to expand the business. The growth option enables the firm to expand the portfolio of software for a lower investment cost by reusing components from previous software. This is more cost effective than starting from scratch every time new software is developed.

How will the growth option change the value initial decision setting? The first thing to note is that the additional investment is an opportunity for the firm and must not be undertaken. So, for the decision setting to change, the firm must choose to exercise the option. The firm should only do so if it provides a greater value than otherwise in accordance with the value maximization proposition. The cash flows for the new decision setting can be depicted as below in Figure 3

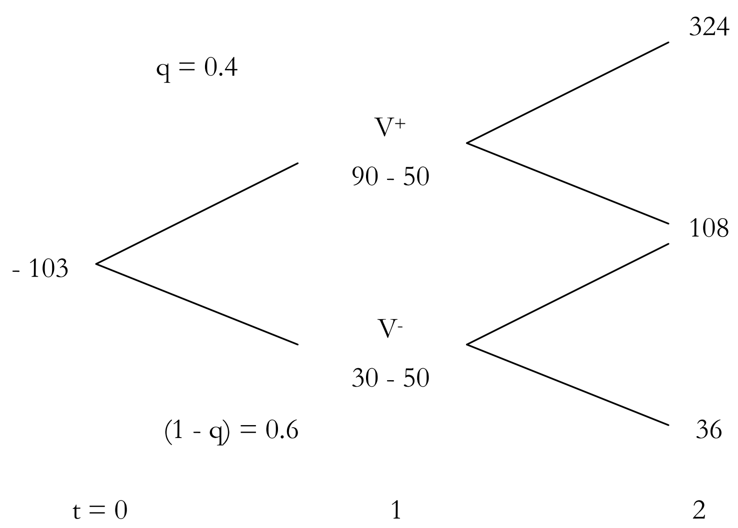

Figure 3. The cash flow setting for the option to expand

The value of the investment can then be determined by assessing the value of V^+ and V^- by modifying Equations 6 and 7 to see if the firm should undertake the additional investment i.e. exercise the option to expand the software project. The value of the decision in the up and down state in $t = 1$ can now be stated as

$$ENPV^+ = MAX(V^+ - I_E, 0) \tag{8}$$

and

$$ENPV^- = MAX(V^- - I_E, 0). \tag{9}$$

The intuition behind Equations 8 and 9 is the same as in valuation of the option to deferral; the firm will only undertake the additional investment if incremental value is created. This is implemented in the above equations only looking at the maximum of the two choices. So, zero in the equations refers to getting no value from the decision and the node is valued as in the initial investment setting. The value of the up and down state is then

$$ENPV^{+} = MAX\begin{bmatrix} 90 + (0.4\times 324 + 0.6\times 108)/1.08 - 50, \\ 90 + (0.4\times 162 + 0.6\times 54)/1.08 \end{bmatrix} = 220$$

and

$$ENPV^{-} = MAX\begin{bmatrix} 30 + (0.4\times 108 + 0.6\times 36)/1.08 - 50, \\ 30 + (0.4\times 54 + 0.6\times 18)/1.08 \end{bmatrix} = 60\,.$$

The above calculations show that if the market conditions are favorable, the firm shall go ahead the sequential investment. But, if market conditions turn unfavorable, the firm shall not go ahead with the sequential investment since the discounted expected cash flow does not exceed the investment cost for the additional software.

The value of the software development project at $t = 0$ with the option to expand is

$$ENPV = \frac{qENPV^{+} + (1-q)ENPV^{-}}{1+r} - I_0 = \frac{0.4\times 220 + 0.6\times 60}{1.08} - 103 = 11.81$$

and according to Equation 2 the flexibility to expand the investment is worth

$$O = ENPV - NPV = 11.81 - (-3) = 14.81\,.$$

The above example shows that the option to expand an investment can turn an investment that should be rejected into an investment that should be undertaken.

The option to expand the software or parts of it in other software projects is common in many types of software project. Examples of this is platform based software development where the same platform is used as an underlying component for more than one software, see e.g. Nummien and Wrenne (2009; 2010). Another type of development strategy where this line of reasoning can be seen as the underlying value creation is software product line engineering. A software product line is a portfolio of software systems that are created from a shared set of software resources and components see e.g. Clements and Northrop (2001) for further discussion. The most simple and general setting for the option to expand is the possibility to launch a subsequent version or add-ons to already existing software. By the same line of reasoning, investments in the software architecture can be motivated as options to expand rather than just viewing then as costs as suggested by e.g. (Kazman et al., 2001).

4.3 The option to abandon

This section will discuss another setting under which the software project could be justified even though the firm chooses to go ahead with the investment in $t=0$, i.e. the initial setting with the standard NPV-valuation.

Assume that the firm can sell of the software or parts of it if the firm cannot capitalize it due to market conditions as discussed by e.g. Brereton et al. (2002); Traas and van Hillegersberg (2000). The term option to abandon is instead of option to abandon or scale back to simplify to notation without loss of generality.

Assume that the abandonment will yield \$40million if the firm chooses to do so in $t = 1$. The firm will only do this if it is profitable for the firm to do so. The firm will only exercise the option to abandon the software project if this increases the value of the investment. The cash flow for software development project can be depicted as shown in Figure 3.4 below when the abandonment option is included.

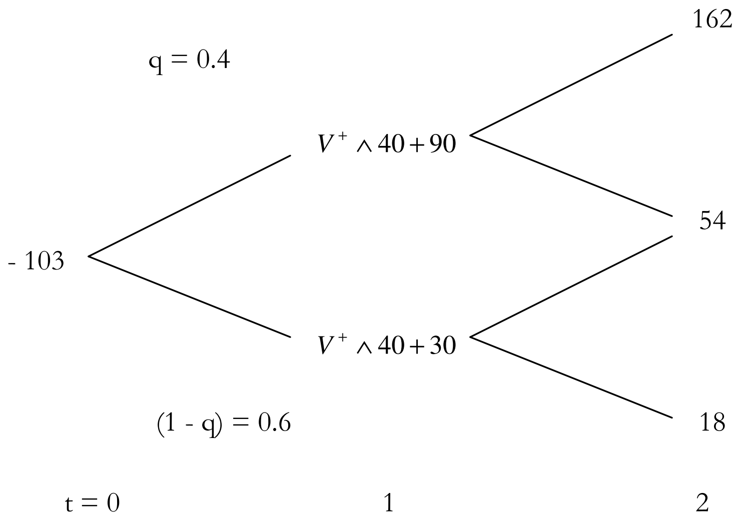

Figure 4. The cash flow setting for the option to abandon

The intuition behind the cash flows in Figure 4 is that the firm will choose the highest cash flow it can get in $t = 1$. The choice is to keep the investment and get the cash flows from $t = 2$ or to sell of the investment and get the pre-contracted or known salvage value of \$40 million. The choice of the firm is based on the value in the up and the down state which can be stated as

$$ENPV^+ = MAX\left(A + CF^+, 0\right) \tag{10}$$

and

$$ENPV^{-} = MAX\left(A + CF^{-}, 0\right), \tag{11}$$

where A denotes the value of the abandonment value of the software and CF^{+} and CF^{-} denotes the cash flow in the up and down state. The 0 denotes as previously the value of the option if it is not exercised.

The value of the of the up and down state is then

$$ENPV^{+} = MAX\left[90 + 40, 90 + \left(0.4 \times 162 + 0.6 \times 54\right) / 1.08\right] = 180$$

and

$$ENPV^{-} = MAX\left[30 + 40, 30 + \left(0.4 \times 54 + 0.6 \times 18\right) / 1.08\right] = 70 .$$

Based on the above calculations, the firm will choose to exercise the option to abandon if the market turns down since the abandonment value of the option is greater than $E\left[CF_{t=2}\right] / \left(1 + r\right)$, i.e. the expected cash flows at $t = 2$ discounted back to $t = 1$.

The value of the software development project with the option to abandon it is then

$$ENPV = \frac{qENPV^{+} + \left(1 - q\right)ENPV^{-}}{1 + r} - I_{0} = \frac{0.4 \times 180 + 0.6 \times 70}{1.08} - 103$$
$$= 2.56$$

and the value of the flexibility that the option has added is

$$O = ENPV - NPV = 2.56 - \left(-3\right) = 5.56 .$$

The possibility to abandon the software project has turned negative NPV investment which should not be undertaken into an investment which adds value to firm instead and should be undertaken. This will hold true as long as the firm can resell the software project above to a minimum price which can be determined for this example by solving

$$ENPV = \frac{qENPV^{+} + (1-q)ENPV^{-}}{1+r} - I_0 = 0 = \frac{0.4 \times 180 + 0.6 \times A_{MIN}}{1.08}$$
$$-103 = 0 = 35.4$$

where $A_{MIN} = CF^{+} + A$. As long as the firm can resell the software project for a higher price than \$35.4 million the investment is worth undertaking from an economical stand point of view in accordance to the value maximization proposition.

In this example the abandonment option is discussed in the case where the firm can resell the software or parts of it. This opportunity is only created if there is someone else who can use the software or the components that is sold. The value of the exercise price can therefore be argued to be negatively correlated with the degree of asset specificity since low degree of asset specificity increases the potential area of use for the software or the components of the software that is being sold, see Zaher and Venkatraman (1994); Malone et al. (1987); Williamson (1985) for a discussion of the different dimensions of asset specificity. For software components, this would imply that more general components such as e.g. software architectures and software platforms would yield a higher exercise price than specific applications as implied by Numminen and Wrenne (2009).

5. DISCUSSION

The fundamental conclusion from the examples in this paper is that real options decrease uncertainty by creating flexibility and therefore adds value to the investment. The value added creates a asymmetry in the cash flows from the investment which makes it cumbersome to use the standard NPV-calculation in the valuation. To use the NPV-calculation the marginal cost of capital (*k*) must be recalculated throughout the cash flow nodes to cover the correct risk. From a theoretical point of view, this paper has shown that a software development project meets the conditions previously discussed for option valuation to better capture the true value of the investment.

The three basic real options exemplified in this chapter, the option to defer, the option to expand and the option to abandon, each have added value to the investment and turned an investment not worth undertaking into a profitable investment worth undertaking. The options have not only added value but have also hedged the underlying uncertainty present in the setting for the decision, i.e. the cash flow uncertainty.

The valuations of the three different types of real options in this chapter are all based the general valuation of financial call and put options. The options to defer and to expand is a call option for the entire investment or for parts of the investment whereas the option to abandon the investment is put option for the underlying asset or a part of it. This similarity can be seen by comparing the boundary conditions used in the valuation of the different options with Equation 3 and Equation 4. More specifically, the types of options used in this chapter resemble European style options based on dividend paying underlying assets. This makes the Black and Scholes option pricing model not applicable. On a general level, most software developing projects would resemble American style options, as argued before, but since the exercise time has been predetermined in the numerical examples discussed in this chapter the options are of European style.

By creating different portfolios of the basic real option exemplified in this chapter, all real options discussed earlier can be derived. Depending on the uncertainties present in the decision setting, different types of real options become relevant to plant by redefining the decision setting. The portfolios of these options would in most cases be as easy to plant into software investments and could hedge more kinds of uncertainties than exemplified in the examples presented here. By adopting the methodology presented in this chapter, more complicated and realistic investment decisions can thereby be analyzed.

Finally, the examples in this paper analyze the investment from the developer's point of view. However, the methodology is general and same type of examples could be set up for an acquiring firm's point of view as well to hedge the uncertainty in those decision settings. The option to expand exists every time when the firm is buying module-based software. Buy buying the basic module the option to expand this software exists and should be exercised when needs for additional modules occur. The firm is at the same time using an option to defer the acquisition of modules that are not needed in the present. When the firm has acquired the software it always has the option to abandon by stopping the usage of the software. If it is leased software the firm gains the savings on not having to pay for the leasing. If the firm has the ownership over the software the firm will exercise the option if cost of usage exceeds the gains from usage.

5.1 Suggestions for further research

This paper has shown the value that is added by three different types of real options and how they can be incorporated in most software development projects. Most papers using real option theory in software economics is still theoretical papers and there thus exists a need for empirical studies on the topic.

There is a need for studies that analyses the effect on software developments project given an option based development strategy. To be able to further develop the theoretical basis for option theory in software economics it is important to understand the setting of the decisions that are analyzed.

Further, there is need for a reality check on if and how option theory is used by software developing firms so that appropriate models can be developed. If option theory is not used to the extent it can to help managers in making these decisions under uncertainty due to the complexity of using some of the real option models, there is an even greater need for empirical research on how to overcome this. There may also be need for theoretical work on developing simplified models that still captures the essence of option theory but are easier to implement.

REFERENCES

Amran, M., Kulatilaka, N. (1999) “Real Options: Managing Strategic Investments in an Uncertain World”, *Harvard Business School Press*, Boston

Ashford, R. W., Dyson, R. G., Hodges, S. D. (1998) “The Capital-Investment Appraisal of new Technology: Problems, Misconceptions and Research Directions”, *The Journal of Operational Research Society*, Vol. 39, No. 7, pp. 637-642

Ballantine, J., Stray, S. (1998) “Financial Appraisal and the IS/IT Investment Decision Making Process”, *Journal of Information Technology,* Vol. 13, No. 1, pp. 3-14

Benaroch, M. (2002) “Managing Information Technology Risk: A Real Options Perspective”, *Journal of Management Information Systems*, Vol. 19, No. 2, pp. 43-84

Benaroch, M., Kauffman, R. J. (1999) ”A Case for Using Real Option Pricing Analysis to Evaluate Information Technology Project Investments”, *Information Systems Research*, Vol. 10, No. 1, pp. 70-86

Benaroch, M., Kauffman, R. J. (2000) ”Jusitfying Electronic Banking Network Expansion Using Real Options Analysis”, *MIS Quarterly*, Vol. 24, No. 2, pp. 197-225

Benaroch, M., Lichtenstein, Y., Robinson, K. (2006) “Real Options in Information Technology Risk Management: An Empirical Validation Risk Option Relationship”, *MIS Quarterly*, Vol. 30, No. 4, pp. 827-864

Black, F., Scholes, M. (1973) "The Pricing of Options and Corporate Liabilities", *Journal of Political Economy*, Vol. 83, No. 3, pp. 637-654

Boehm, B. W. (1984) "Software Engineering Economics", IEEE Transactions on Software Engineering (TSE), Vol. 10, No. 6, pp. 290-302

Borenstein, D., Betencourt, P. R. B. (2005) "A Multi-Criteria Model for IT Investments", *INFOR*, Vol. 43, No. 1, pp. 1-21

Brereon, P., Linkman, S., Thomas, N., Bøegh, J., De Panfilis, S. (2002) "Software Components – Enabling a Mass Market*", Proceedings of the 10th International Workshop on Software Technology and Engineering Practice*, pp. 169-176

Chalasani, P., Jha, S., Sullivan, K. (1998) "An Options Approach to Software Prototyping", Technical Report

Clements, P., Northrop, L. (2001) "Software Product Lines: Practices and Patterns", *Addison-Wesley Professional*, New York

Clemmons, E. (1991) "Evaluation of Strategic Investments in Information Technology", *Communication of the ACM*, Vol. 34, No. 1, pp. 22-36

Copeland, T., Koller, T. Murrin, J. (2001) "Valuation: Measuring and Managing the Value of Companies", *John Wiley and Sons*, New York

Copeland, T., Antikarov, V. (2003) "Real Options – A Practitioner's Guide", *Texere*, New York

Cox, J. C., Ross, S. A., Rubinstein, M. (1979) "Option Pricing: A Simplified Approach", *Journal of Financial Economics*, Vol. 7, No. 3, pp. 229-263

De Reyck, B., Degraeve, Z. Vandenborre, R. (2008) "Project Option Valuation with net Present Value and Decision Tree analysis", *European Journal of Operational Research*, Vol. 184, No. 1, pp. 341-355

Dixit, A., Pindyck, R. S. (1994) "Investment under Uncertainty", *Princeton University Press*, Princeton

Dixit, A., Pindyck, R. S. (1995a) "The Option Approach to Capital Investment", Harvard Business Review, May/June, pp. 105-115

Dixit, A., Pindyck, R. S. (1995b) "The Option Approach to Capital Investment" *Long Range Planning*, Vol. 28, No.4, pp. 129-129

Dos Santos, B. L. (1991) "Justifying Investments in new Technologies", *Journal of Management Information Systems*, Vol. 7, No.4, pp. 71-90

Erdogmus, H. (2002) "Valuation of Learning Options in Software Development Under Private and Market Risk", *The Engineering Economist*, Vol. 47, No. 3, pp. 308-353

Favaro, J. M., Favaro, K. J., Favaro, P. F. (1998) "Value Based Software Reuse Investment", *Annals of Software Engineering*, Vol. 5, No. 1, pp. 5-52

Fichman, R. G. (2004) "Real Option and IT Platform Adoption: Implication for Theory and Practice", *Information Systems Research*, Vol. 15, No. 2, pp. 132-154

Fisher, I. (1930) "The Theory of Interest", *The Macmillian Co.*, New York

Hallikainen, P, Kivijärvi, H, Nurmimäki, K. (2002) "Evaluating Strategic IT Investment: An Assessment of Investment Alternatives for a Web Management System", *Proceedings of the 35th Hawaii International Conference on System Science*, pp. X-XX

Harrison, M. J., Kreps, D. M. (1979) "Martingales and Arbitrage in Multiperiod Securities Markets" *Journal of Economic Theory*, Vol. 20, No. 3, pp. 381-408

Harrison, M. J., Pliska, S. J. (1981) "Martingales and Stochastic Integrals in the Theory of Continuous Trading", Stochastic Processes and Their Applications, Vol. 11, No. 3, pp. 215-260

Hull, J. C. (2008) "Options, Futures, and other Derivatives", *Prentice Hall*, New York

Irani, Z. (2008) "Information Systems Evaluation: What Does it Mean?", *Construction Innovation*, Vol. 8, No. 2, pp. 88-91

Irani, Z. (2002) "Information Systems Evaluation: Navigating Through the Problem Domain", *Information and Management*, Vol. 40, No. 1, pp. 11-24

Jensen, M. C. (2001) "Value Maximization, Stakeholder Theory, and the Corporate Objective Function", *European Financial Management*, Vol. 7, No. 3, pp. 297-317

Kambil, A., Henderson, J. C., Mohsensadeh, H. (1993) "Strategic Management of Information Technology: an Options Perspective", *CISR WP No. 222 – Sloan WP No. 3319*, available at:

http://dspace.mit.edu/bitstream/handle/1721.1/49166/strategicmanagem00kamb.pdf?sequence=1
2009-11-03

Kazman, R.,.Assunsi, J., Klein, M. (2001) "Quantifying the Costs and Benefits of Architectural Decisions", *Proceedings of the 23rd International Conference on Software Engineering (ICSE 23)*, Toronto, pp. 297-306

Kester, W. C. (1984) "Today's Options for Tomorrow's Growth", *Harvard Business Review*, Vol. 62. No. March/April, pp. 153-161

Klecun, E., Cornford, T. (2005) "A Critical Approach to Evaluation", *European Journal of Information Systems*, Vol. 14, No. 3, pp. 229-243

Kumar, R. L. (1996) "A Note on Project Risks and Option Values of Investments in Information Technologies", *Journal of Information Management Systems*, Vol. 13, No. 1, pp. 187-193

Kumar, R. L. (1999) "Understanding DSS Value: An Options Perspective", *Omega*, Vol. 27, No. e, pp. 295-304

Lefley, F. (1994) "Capital Investment Appraisal of Advanced Manufacturing Technology", *International Journal of Production Resources*, Vol. 32, No.12, pp. 2751-2776

Magrabe, W. (1978) "The Value of an Option to Exchange One Asset for Another", *Journal of Finance*, Vol. 33, No. 1, pp. 177-186

Malone, T. W., Yates, J., Benjamin, R. I. (1987) "Electronic Markets and Market Hierarchies", Communications of the ACM, Vol. 30, No. 6, pp. 484-497

McDonald, R. L. (2005) "*Derivatives Market*", Addison Wesley, New York

McGrath, R. G. (1997) "A Real Option Logic for Initiating Technology Positioning Investments", *Academy of Management Review,* Vol. 22, No. 4, pp. 974-996

Milis, K., Mercken, R. (2004) "The use of Balanced Scorecard for the Evaluation of Information and Communication Technology Projects", *International Journal of Project Management*, Vol. 22, No. 2, pp. 87-97

Myers, S. C. (1977) "The Determinants of Corporate Borrowing", *Journal of Financial Economics*, Vol. 5, No. 2, pp- 147-176

Panayi, S., Trigerorgis, L. (1998) "Multi-Stage Real Options: The Case of an IT Infrastructure and International Bank Expansion", *The Quarterly Review of Economics and Finance*, Vol. 38, No. 3, pp. 675-692

Pindyck, R. S. (1993) "Investments of Uncertain Costs", *Journal of Financial Economics*, Vol. 34, pp. 53-76

Powell, P. (1992) "Information Technology Evaluation: Is it Really Different?", *The Journal of Operational Research Society*, Vol. 43, No. 1, pp. 29-42

Smith, J. E., Nau, R. F. (1995) "Valuing Risky Projects: Option Pricing Theory and Decision Tree Analysis", *Management Science*, Vol. 41, No. 5, pp. 795-816

Schwarz, E. S., Moon, M (2000) "Evaluating Research and Development Investments", in *Project Flexibility, Agency and Competition*, Brennan, M. J., Trigerorgis, L. (Eds.) (2000), *Oxford University Press*, New York

Schwarz, E. S., Trigeorigis, L. (2004) (Ed) "Real Options and Investment Analysis under Uncertainty – Classic Readings and Contributions", *The MIT Press*, Cambridge

Schwartz, E. S., Zozaya-Gorostiza, C. (2000) "Valuation of Information Technology Investments as Real Options", *AFA 2001 New Orleans Meeting*, available at:
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=246576
2009-11-03

Schwartz, E. S., Zozaya-Gorostiza, C. (2003) "Investment Under Uncertainty in Information Technology: Acquisition and Development Projects", *Management Science*, Vol. 49, No. 1, pp. 57-70

Shapiro, C., Varian, H. (1999) "Information Rules – A Strategic Guide to the Network Economy", *Harvard Business School Press*, Boston

Serafemidis, V., Smithson, S. (2000) "Information System Evaluation in Practice: A Case Study of Organizational Change", *Journal of Information Technology*, Vol. 15, No. 2, pp. 93-106

Small, M. H., Chen, J. (1995) "Investment Justification of Advanced Manufacturing Technology: An Empirical Analysis", *Journal of Engineering and Technology Management*, Vol. 12, No. 1-2, pp. 27-55

Taudes, A. (1998) "Software Growth Options", *Journal of Management Information Systems*, Vol. 15, No. 1, pp. 165-185

Taudes, A., Feurstein, M., Mild, A. (2000) "Option Analysis of Software Platform Decisions: A Case Study", *MIS Quarterly*, Vol. 24, No.2, pp. 227-243

Traas, V., van Hillegersberg, J. (2000) "The Software Component Market on the Internet - Current Status and Conditions for Growth", *Software Engineering Notes*, Vol. 25, No. 1, pp. 114-117

Trigeorgis, L. (1996) "Real Options – Managerial Flexibility and Strategy in Resource Allocation", *The MIT Press*, Cambridge

Walter, S. G., Spitta, T. (2004) "Approaches to ex-ante Evaluation of Investments into Information Systems", *Wirtschaftsinformatik*, Vol. 46, No. 3, pp. 171-180

Williamson, O. E. (1985) "The Economic Institutions of Capitalism", *Free Press*, New York

Zaheer, A., Venkatraman, N. (1994) "Determinants of Electronic Integration in the Insurance Business", *Management Science*, Vol. 40, No. 5, pp. 549-566

CHAPTER 6
PAPER 3

Based on Numminen, E. (2008) "Valuation of a Software Investment when you have the Option to Customize it – The case of open Source Software", presented at the 18th Triennial Conference of the International Federation of Operational Research Societies (IFORS) in Johannesburg, South Africa, unpublished

VALUATION OF A SOFTWARE INVESTMENT WHEN YOU HAVE THE OPTION TO CUSTOMIZE IT – THE CASE OF OPEN SOURCE SOFTWARE

Emil Numminen
School of Management, Blekinge Institute of Technology, Ronneby, Sweden
emil.numminen@bth.se

ABSTRACT

The research about open source (OSS) software has mainly been focusing on three issues: Firstly, why do individuals, and to some extent firms, participate in the developing these software. Secondly, what types of business models can be derived for firms from OSS:s, and thirdly, some

research has focused on the adoption issues of OSS. This paper takes a different view on the third issue in relation to previous research by viewing OSS as a software investment where the option to customize the software is included. The main part of this paper analyses the value that is created by that option and how much the software shall be customized.

KEYWORDS

Open source software, Real Option, Customize, Usage

1. INTRODUCTION

The economics of software has taken the full loop to some extent. In the early time of the personal computer (the 1950:s) software was freely distributed with the computer that you bought and it was not until later that you had to pay for the software that you wanted for your computer (Glass, 2004), see Glabel (1991) for a historical discussion on the matter. Today software has to some extent become free again via open source software. Some even predict that OSS will lead to the end of proprietary software (Raymond, 2000). If OSS will lead to the end of proprietary software is for the future to show it will have some impact on the market for proprietary software since it changes the market structure and the equilibrium under which the software market is functioned under. Some general evidence for this can be found by studying other markets where a low-price alternative has entered, see e.g. the effect on the European aviation industry when Ryanair and other price-cutters entered the market.

Under what circumstances is software OSS? According to the open source initiative (www.opensource.org) there are 10 criteria that must be met for the software to be classified as an OSS. The 10 criteria for the license are:

1. Free redistribution of the software
2. The software must include the source code
3. You are allowed to modify the source code
4. Must respect the integrity of the authors source code
5. May not discriminate against person or group
6. May not discriminate against field or endeavor
7. The rights to software is freely redistributed
8. The rights to program is not attached to an eventual distribution that the software is a part of
9. The license may not restrict use of other software
10. The license must technology neutral

For more on the specific meaning of the different criteria the reader is referred to www.opensource.org or sees Riehle (2007) for the implications or Krishnamurthy (2003) for discussion about different types of OSS licenses. Open source software can be found in many of the major types of software applications; Linux among operating systems, MySQL among database applications, Open Office among office software packages and Firefox among web browsers just to name a few. For a greater discussion concerning the history and origin of OSS see Lerner & Tirole (2002; 2005).

2. PREVIOUS RESEARCH ON OSS

Research about the economics of OSS has been conducted for some years now. As with any research, several streams of research have evolved within the field. The larger scope of this research has been focuses on incentives of individuals to participate in OSS development. Lerner & Tirole (2001) discusses career concern incentives and ego gratification as examples of benefits from participation. Johnson (2001) develops a game theoretical model to show why an purely self-interested programmer would want to participate in an OSS project. Engelbrecht (2006) adds on to this research by analyzing how trust and the average happiness in a country correlates with OSS. Roberts et al. (2006) studied the Apache project to see how intrinsic and extrinsic motivation affects participation and performance.

Other research focus on the business models for profit seeking firms participating in OSS projects. Raymond (2000) indentifies seven different types of business models that a firm can apply involving OSS and also discuss real-life firms who have done so. Mustonen (2001) develops a model to analyze the economic interaction between a proprietary software monopolist and the open source community. Lerner & Tirole (2000) also looks at the interaction between closed form software and OSS and the reasons for why the firm encourages their employees to participate in OSS projects.

Other research has focus on the adoption of OSS, Katsamakas & Xin (2005) develops a model to analyze under what conditions a firm will adopt an OSS. They conclude that the decision is based upon network effects, the fit with existing applications used by the firm and the IT capability of the firm. Aija & Wu (2007) empirically studied the issue of reusing components of OSS and found that companies can get productivity increases and quality increases given reuse of code. Another stream of research has focused on at the overall success of OSS by its parts, Bonaccorsi & Rossi (2003) apples Schumpeter's three-phase innovation axis (invention, innovation and diffusion) to study the motivation to participating in free development and how this development is coordinated with our proprietary rights, and finally the diffusion of OSS in a proprietary surrounding. Glass (2004) looked at the general economic circumstances for OSS.

This literature review is in no way complete and for more about the different research conducted concerning on OSS see the literature reviews in Schiff (2002) and Katsamakas & Xin (2005).

2.1 Aim of the paper

Although the above research, there is however little research conducted concerning the decision making in the individual firm that faces the choice between OSS and proprietary software and the investment it involves. This paper looks at four different adoption decision as investment decisions. These four decision situations are differentiated based upon the firms approach towards the software. In this paper we see usage of the software as the underlying value creation process. Given this, the firm has an option to increase the usage by customizing the software for the users and specific needs of the organization. Further, since the firm does not have the same total cost for the software as they would have for a strict proprietary similar software on the market, the level of usage do not need to be as high for the investment to be justified.

This paper will analyze four different situations: 1). When the firm does not make any adjustments to the software and there is no investment cost. 2). When the firm does not make any adjustments to the software but there is a non-zero investment cost. 3). When there is uncertainty about the future level of usage for a given level of customization costs. 4). When the customization cost is uncertain for the future level of usage.

To show the value of the option, a one period model will be developed for both situations. The underlying assumption for the modeling is that the uncertainty is resolved after one period. The second situation will be used as a base case for the analysis. This situation will use a traditional Net Present Value (*NPV*) valuation to assess the value of the decision and will be used as a benchmark for situation 3 and situation 4.

The rest of paper will be organized as follows. Section 3 will lay the basis for the assumed economic behavior of the firm and Section 4 will develop the underlying theoretical tools used in the analysis of the decision settings. Section 5 will present the analysis of the four different decision situations analyzed in this paper and section 6 concludes the paper with discussion and conclusions as well as suggestions for further research.

3. THE SETTING FOR THE DECISION

Since this paper examines the economics behind various situations concerning the adoption of an OSS we need to assign the economic view of the firm. We need to do this so an decision rule can be designed for the firm to act in accordance with. Therefore, throughout this paper we therefore make the assumption that the firm is acting in a rational manner. To do so the firm will only make decision for which the outcome will create more benefits in comparison to the cost associated with that decision. This behavior is referred to as the value maximization principle by e.g. Jensen (2001a; 2001b) or Rose (1999) and is a somewhat standard assumption about the firms behavior in financial economics.

4. Underlying theories for the modeling

In this paper, the cash flow function *CF* is modeled as a function of the usage of the software:

$$CF = f(u) \tag{1}$$

where u denotes the usage of the software. A cash flow is defined as the residual between cash in to the firm and cash out from the firm, see e.g. (Berk & DeMarzo, 2007) for a more precise definition and discussion. The intuition behind the functional form of the cash flow function comes from the IS/IT literature, see e.g. Straub et al. (1995), Devaraj &

Kohli (2003) or Venkatesh et al. (2003) for a discussion concerning software usage and the effect on firm performance. The actual usage of the software is uncertain in most cases and hence also the expected usage. However, in accordance with the software economics literature, in order to affect the cash flow the usage must be affected. Davis (1989) and Taylor & Todd (1995), among others, argue that perceived ease of use and perceived usefulness is positively correlated with software usage. This is the core of what creates the need for adjusting the software for the individual needs of the organization. If the firm adopts an OSS they will have the option to do so because of the access to the source code of the software, which according to above must be distributed with the software. In this way the firm has acquired a real option to increase the usage of the software and hence increase the cash flow. The firm can thereby decrease the uncertainty of the expected usage.

In this paper, a real option is defined as a right, but not an obligation, to take some action in the future and in most cases the right would be for a predetermined price and for a predetermined time, see e.g. Dixit & Pindyck (1994; 1995), Trigeorgis (1996) or Copeland & Antikorov (2003). Real options have clear resemblance to financial options and any real option can be designed by using standard American call options and put options, see Hull (2006) for more on financial options.

4. THE FOUR DIFFERENT DECISION SITUATIONS

This part of the paper will present the four different situations that the firm faces concerning the OSS. For each situation we will provide the conditions under which the firm shall undertake the investment that the decision concerns and also the value of the OSS.

4.1 Situation 1

Under this situation the firm does not plan to make any changes to the software and hence chooses not to exercise the option embedded in the software. The value of the software can then be assessed by calculating the NPV as follows:

$$NPV = -I + \sum_{t=0}^{T} \frac{CF_t}{(1+k)^t} = \sum_{t=0}^{T} \frac{CF_t}{(1+k)^t} \qquad (2)$$

where k denotes the opportunity cost for the capital used for the investment, t denotes time and other variables as above defined. Although the software is free of charge for the firm there still exists an opportunity cost for the firm. The opportunity cost of capital is the measure for the cost of forgoing the best use of these resources. The resources used in this situation are captured by I which is assumed to be zero.

According to the decision rule

$$MAX[NPV, 0] \qquad (3)$$

the investment shall be made iff, i.e. if and only if, the discounted cash flows exceed the resources used. This result is under the assumption that the firm uses its funds in a rational way.

Situation 1 would capture the situation if the firm is adopting the OSS from a strict cost saving perspective compared to proprietary software. Under this situation the firm shall adopt the software under any level of usage however low it may be since it results in a positive CF. By doing so the firm would act in accordance with Equation 3. Since the firm is not customizing the software it is assumed that it will start receiving cash flows immediately.

4.2 Situation 2

In situation 2 the firm is still not considering making any changes to the software but *I* is non-zero. The value of the software can then be assessed by:

$$NPV = -I + \sum_{t=0}^{T} \frac{CF_t}{(1+k)^k} \tag{4}$$

where variables as above defined and the firm is expected to decide according to Equation 3.

Even though the software is an OSS we assume that *I* in Equation 2 is non-zero because of e.g. implementation costs for software. The firm shall therefore not adopt the software under any level of usage because it provides a positive *CF* as in situation 1 given the type of software discussed. The result in situation 2 is true even under no uncertainty because of the non-zero *I* since the investment is irreversible, i.e. *I* is a sunk cost. The implication of going from situation 1 to situation 2 is that the investment is no longer fully reversible since the non-zero *I*. The firm cannot disinvest at a zero cost at a later time period if the usage of the software turns out to be lower for any reason. Since the firm is not customizing the software it is again assumed that it will start receiving cash flows immediately.

4.3 Situation 3

In situation three we will make the decision setting more realistic by introducing uncertainty in the future usage resulting in uncertainty in the expected future cash flow. That expected usage is an uncertain variable is a key factor underlying much of the research in the software economics field (e.g. Taylor & Todd, 1995) and much research is conducted to develop models to predict and explain individual usage of software, see e.g. Venkatesh et al. (2003) for a review of these models.

But as we will show below, it may be rational for the firm to spend additional resource, i.e. increase *I*, in adjusting the software under the assumption that it will increase usage and therefore increase the expected cash flow from the software. This, on the other hand, makes the investment more irreversible compared to situation 2, since additional resources have been spent that the firm cannot recover if the investment turns out to be a bad investment.

In order to incorporate the uncertainty in the usage and hence the cash flow we assume the cash flow can either increase by *u* or decrease with *d* and that the uncertainty is resolved after one period. This means that if the cash flow goes up, it stays up, i.e. usage has increased and has stabilized, or if the cash flow decreases, it stays down, i.e. usage decreases and stabilizes. This period can be seen as an adoption period for the software where we learn the actual stable usage over time, i.e. the diffusion curve of the software, see e.g. Fichman & Kemerer (1997; 1999). Since there is an uncertainty to whether the usage will increase or decrease, we will incorporate this uncertainty by stating that there is a *q* probability that *u* will occur and a *1-q* probability that *d* will occur. We can now express the value of the software as

$$NPV_{DTA} = -I + p_u \sum_{t=1}^{T} \frac{CF_t^u}{(1+k)^t} + p_d \sum_{t=1}^{T} \frac{CF_t^d}{(1+k)^t}. \qquad (5)$$

We apply the same decision rule to Equation 5 as to the above situations, e.g. Equation 3. The main difference between Equation 5 and Equation 4 is that Equation 5 for takes into account the change in the cash flow and the likelihood of that change; we therefore call it NPV_{DTA} where *DTA* stands for Decision Tree Analysis.

If the firm chooses to exercise the option to customize software Equation (4) will no longer capture the value of the software. We assume that it takes one period to adjust the software for the organization and that the total investment cost for the software is then I^{+} including the costs for adjustments. Not that the firm will not receive any cash flows under the first period since the software is not in use. This means that the firm will have to take that into account that the customized software must earn additional long time cash flows to compensate for the loss of first year cash flow. We will denote this long term level of cash flow CF_{BE} and define p^{*} as the probability that the break even cash flow will be the true state. Given this, it will be rational for the firm to go ahead with the investment if

$$NPV_{ROA} = -I^{+} + p^{*}\sum_{t=1}^{T}\frac{CF_t^{*}}{(1+k)^{t}} \qquad (6)$$

s.t. $CF_t^{*} > CF_{BE}$ where CF^{*} denotes the cash flow from the customized software. The decision rule for Equation 6 is same before, i.e. Equation 3. The value of the option (OV) will then be

$$OV = NPV_{ROA} - NPV\ . \qquad (7)$$

The intuition behind Equation 7 comes from flexibility that is included in Equation 6 but not in Equation 4 because Equation 6 includes the value created by the fact that the firm has an opportunity to make a choice to customize the software to increase the perceived ease of use and perceived usefulness of the software. Note that according to the constraint for Equation 6 states that the option only adds value of the customization makes results in cash flow exceeding the break even cash flow level. If not, the customization only increases the investment cost for the firm and yields no added cash flow.

4.4 Situation 4

The last situation to be analyzed is how much money is rational for the firm to invest in customizing the software. In order for the firm to act in accordance with Equation 3 the firm can either maximize the cash flows for a given amount of invested funds, or the firm can minimize the invested funds for a given level of cash flows. Given situation 3 the firm faces a threshold level for the cash flow that must be exceeded for the firm to exercise the option to customize. This threshold level for the customized software is the break even cash flow. If the cash flow does not exceed CF_{BE} the additional costs for the customization will be a sunk cost for the firm. So for the firm to minimize the investment for a cash flow above the break even cash flow the shall go ahead with the investment if

$$NPV_{ROA} = -I^{+}_{MIN} + p^{*}\sum_{t=1}^{T}\frac{CF_t^{*}}{(1+k)^{t}} \qquad (8)$$

s.t. $CF_t^{*} > CF_{BE}$. The decision rule for Equation 8 is the same as previously, i.e. Equation 3 and the value of the option can be assessed by Equation 7.

The intuition behind Equation 8 is that the firm wants to minimize the total cost of the software for a given level of cash flow above the break even cash flow. The firm will therefore only invest at the minimal level for each marginal increase of usage given the restriction to the optimization problem.

5. DISCUSSION AND CONCLUSIONS

This paper has shown that cost savings may not be the only reason why a firm will adopt an OSS. The greater value from an OSS may come from the option to customize the software in an effort to maximize the cash flows from the usage. Given an OSS the firm has the flexibility to decide upon how much to customize the software and therefore different firms may use the same software to a different extent. This flexibility is not included in proprietary software or at least not at the same cost. For the option to be included in proprietary software it would most likely have to be bespoke software. The decision to order bespoke software would change the decision setting entirely because of the increase in the cost that the firm faces.

The main conclusions of this paper are if the firm is only adopting OSS for cost reasons it shall do so if it results in usage and therefore positive cash flow. By doing so, however, the firm misses out on the value from the option to customize the software and increase the cash flow. The latter is due to increased usage because of if perceived ease of use and perceived useful of the software. The firms shall however only invest in customization as long as it provides marginal increase in usage that motivates the cost of customization.

As a final note about the analysis in this paper it should be noted that the analysis is not limited to the case of OSS. The analysis is general and also applies for proprietary software as well. The difference is that we would have a non-zero investment for the software and therefore, C. P., it would require higher usage for the investment to break-even. Therefore it could be argued that customization is even more important to increase the adoption of the software although the firm must pay for this option.

The managerial implication of this paper is that the firm has to decide upon how the OSS is going to be adopted based upon the expected result. An OSS is from an economic stand point a resource as any other asset in the firm and shall therefore also be treated under the value maximization proposition. The basic underlying message to management from this paper is that an OSS provides an opportunity not included in standard proprietary software and could increase the value of the investment if dealt with correctly.

Since the economic circumstances concerning OSS from a firm point of view is still rather unknown, there needs to be more research conducted concerning how firms reason about adopting OSS. The common view is that adopting OSS is a cost saving strategy but it instead be a benefit increasing strategy. It is therefore proposed that more empirical studies are conducted concerning the adoption of OSS and especially at firms that have made modifications of the OSS to study the economic outcome of it.

REFERENCES

Aija, S. A., W, D. (2007) "Empirical Study of the Effects of Open Source Adoption on Software Development Economics", *The Journal of Systems and Software*, Vol. 80, No. 9, pp. 1517-1529

Berk, J., DeMarzo, P. (2007) "Corporate Finance", *Prentice Hall*, New York

Bonaccorsi, A., Rossi, C. (2003) "Why Open Source Software Can Succeed", *Research Policy*, Vol. 32, No. 7, pp. 1243-1248

Copeland, T., Antikarov, V. (2003) "Real Options: A Practitioner's Guide", *Texere*, New York

Davis, F. D. (1989) "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology", *MIS Quarterly*, Vol. 13, No. 3, pp. 319-340

Devaraj, S., Kohli, R. (2003) "Performance Impact of Information Technology: Is Actual Usage the Missing Link?", *Management Science*, Vol. 49, No. 3, pp. 273-289

Dixit, A. K., Pindyck, R. S. (1994) ”Investment under Uncertainty”, *Princeton University Press*, Princeton, New Jersey

Dixit, A. K., Pindyck, R. S. (1995) ”The Option Approach to Capital Investment”, *Harvard Business Review*, Vol. 28, No. 4, pp. 105-115

Engelbrecht, H-J. (2006) “Can Happiness Economics Help Explain Voluntary Open Source Activities? Some Exploratory Cross-Country Emperices.”
Available at:
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=957174
2009-02-17

Fichman, R. G., Kemerer, C. F. (1997) ”The Assimilation of Software Process Innovation: An Organizational Learning Perspective”, *Management Science*, Vol. 43, No. 10, pp. 1345-1363

Fichman, R. G., Kemerer, C. F. (1999) ”An Illusory Diffusion of Innovation: An Examination of Assimilation Gaps”, *Information Systems Research*, Vol. 10, No. 3, pp. 255-275

Gabel, L. (1991) “Competitive Strategies for Product Standards”, *McGraw-Hill*, London

Glass, R. L. (2004) “A Look at the Economics of Open Source – Is Open Software the Future of the Software Field of a Passing Fad?”, *Communications of the ATM*, Vol. 47, No. 2, pp. 25-27

Hull, J. C. (2006) “Options, Futures and Other Derivatives”, *Prentice Hall*, New York

Jensen, M. C. (2001a) “Value Maximization, Stakeholder Theory, and the Corporate Objective Function”, *European Financial Management*, Vol. 7, No. 3, pp. 297-317

Jensen, M. C. (2001b) “Value Maximization, Stakeholder Theory, and the Corporate Objective Function” in Chew, Jr., D. H., Gillian, S. L. (2005)

"Corporate Governance at the Crossroads: A book of Readings", *McGraw-Hill*, New York

Johnson, J. P. (2001) "Economics of Open Source Software"
Available at:
http://opensource.mit.edu/papers/johnsonopensource.pdf
2009-02-18

Katsamakas, E., Xin, M. (2005) "An Economic Analysis of Enterprise Adoption of Open Source Software", *NET Institute Working Paper Series 05-29*
Available at:
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=851306
2009-02-18

Krishnamurthy, S. (2003) "A Managerial Overview of Open Source Software", *Business Horizons*, Vol. 46, No. 5, pp. 47-56

Lerner & Tirole (2000) "The Simple Economics of Open Source", *HBS Finance Working Paper*, No. 00-059
Available at:
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=224008
2009-02-18

Lerner & Tirole (2001) "The open source movement: Key research questions", *European Economic Review*, Vol. 45, No. 4, pp. 819-826.

Lerner & Tirole (2002) "Some Simple Economics of Open Source", *Journal of Industrial Economics*, Vol. 52, No. 2, pp. 197-234

Lerner & Tirole (2005) "The Economics of Technology Sharing: Open Source and Beyond", *The Journal of Economic Perspectives*, Vol. 19, No. 2, pp. 99-120

Mustonen (2001) "Copyleft - The Economics of Linux and Other Open Source Software", *Information Economics and Policy*, Vol. 15, No. 1, pp. 99-121

Raymond, E. S. (2000) "The Cathedral and the Bazaar", *Thyrsus Enterprises*
Available at:
http://www.catb.org/~esr/writings/cathedral-bazaar/
2009-02-18

Riehle, D. (2007) "The Economic Motivation of Open Source Software: Stakeholders Perspective", *Computer*, Vol. 40, No. 4, pp. 25-32

Roberts, J., Hann, I-H., Slaughter, S. (2006) "Understanding the Motivation, Participation and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects", *Marshall School of Business Working Paper Series No. IOM 01-01*
Available at:
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=918518
2009-02-18

Rose, D. (1999) "Teams, Firms, and Evolution of Profit Seeking Behavior", *Working Paper Department of Economics at the University of Missouri-St. Luis*
Available at:
http://papers.ssrn.com/paper=224438
2009-02-18

Schiff, A. (2002) "The Economics of Open Source Software: A Survey of the Early Literature", *Review of Network Economics*, Vol. 1, No. 1, pp. 66-74

Straub, D., Limayem, M., Karahanna-Evaristo, E. (1995) "Measuring System Usage: Implications for IS Theory Testing", *Management Science*, Vol. 41, No. 8, pp. 1328-1342

Taylor, S., Todd, P. A. (1995) "Understanding Information Technology Usage: A Test of Competing Models", *Information System Research*, Vol. 6, No. 2, pp. 144-176

Trigeorgis, L. (1996) "Real Options: Managerial Flexibility and Strategy in Resource Allocation", *The MIT Press*,

Venkatesh, V., Morris, M. G., Davis, G. B., Davis, F. D. (2003) "User Acceptance of Information Technology: Towards a Unified View", *MIS Quarterly*, Vol. 27, No.3, pp 424-478

CHAPTER 7
PAPER 4

Numminen, E., Wrenne, A (2009) "Uncertainty Reduction in Software Development by the use of a Platform Based Development Strategy", *Proceedings of the 3rd European Conference on Information Management and Evaluation*, pp. 356-362

UNCERTAINTY REDUCTION IN SOFTWARE DEVELOPMENT BY THE USE OF A PLATFORM BASED DEVELOPMENT STRATEGY

Emil Numminen[1], Anders Wrenne[2]

[1,2] School of Management at Blekinge Institute of Technology, Ronneby, Sweden

Emil.numminen@bth.se

Anders.wrenne@bth.se

ABSTRACT

The process of developing software requires a substantial investment and the project has several sources of uncertainty. One part of this uncertainty is the inner uncertainty, i.e. the uncertainty related to the cost and time for developing the software. The second part of the uncertainty

is the outer uncertainty, i.e. the market uncertainty whether the investment will yield a positive return or not.

This paper looks at how the uncertainty associated with software development can be managed. For this purpose, a development model is constructed by analyzing three video games that has become successes for different reasons. The video game industry is used as a benchmark since video games exhibit the same type of uncertainties as software but in a more extreme way while sharing the same product characteristics which makes the comparison possible.

The theoretical point of departure for the analysis is based upon the latest innovation and strategy research. Recent innovation and strategy literature stresses that flexibility in the development not only adds value to the project but also decreases uncertainty in the project because of real options planted into the project. In this paper we focus on how these options were planted into the video games and the result of it. We argue that the possibility to plant these real options is directly related to the use of a platform based development strategy. These video games were chosen so that three different types of platforms could be identified. Each of these platforms resulted in a different type of a real option for future product development.

The analyses focus on the uniqueness in the different platform in the sense that they lead to a competitive advantage for the firm from which different types of real options for future product development can be derived. The general analysis of the platform based development strategy results in three implications for software development. The paper ends with conclusions and suggestions for further research.

KEYWORDS

Platforms, Real options, Software development, Video game Uncertainty

1. INTRODUCTION

The market for software has grown rapidly in the last decades as a result of this and software developing thereby has evolved into a major industry. When developing software, there are several different types of uncertainties facing the project; see e.g. Tüysüz & Kahraman (2006); Pennock & Haimes (2002); Ropponen & Lyytinen (2000) for discussions concerning this. In general terms these uncertainties are associated with the cost of developing the software, the time to market for the software or the return that the software will yield. On the other hand, this is general for more or less all projects but these uncertainties have been argued to more severe for software projects (Fichman, 2004; Hallikainen et al., 2002; Anandarajan & Wen, 1999).

This paper will use successful examples from the video game industry as a benchmark to draw lessons about how the software industry can manage the uncertainties associated with developing new software. From an investment point of view, developing a video game is an uncertain and expensive project with costs exceeding several million US dollars (Jacobs & Ip, 2005) and this would be the same for more or less any software. The uncertainty in recovering the investment cost is due to the shorter and shorter shelf-time a video game gets because consumer preferences are becoming more and more diversified and changing in a faster pace (Jacobs & Ip, 2005; Readman & Grantham, 2006). This aspect is more extreme in the video game industry because of greater competition in comparison to the general software industry. This and the general similarity between the products make the video game industry a suitable benchmark industry.

In analyzing the video game industry, this paper draws upon recent finding from the strategy and innovation literature to analyze how the uncertainty in the videogame development can be managed and ultimately reduced. Innovations, according to Schumpeter (1949) are classified in five different types: new products, new production methods, new sources of supply, exploitation of new markets and new ways of organizing activities. The focus in the literature has been primarily on the first two (Fagerberg, 2003). These two, also known as product innovation and process innovation, is about new or improved products and improved ways to produce these products.

This paper focuses on new and improved ways of developing software, i.e. process innovation by the use of platforms. In this paper we refer to a strategy as, according to e.g. Johnson and Scholes (2006), the direction and the scope of an organization over the long time. A development strategy is then about how a product goes from an idea to a finished product. This paper will be based on the recent literature from the innovation and strategy literature about service platforms and real option as the theoretical framework.

The analysis in this paper is based upon three different video games to highlight different aspects of the developing phase. The focus is on what type of platform that was created and what kind of options for the future this provided and ultimately, how did this affect the uncertainty in the development phase. Because of this focus, the descriptions of the specific video games will be deliberately kept short. In the discussion of the three video games references to other games may occasionally be made to further point out the relevance of platform or the option. The three video games were chosen so that three different uses of platforms can be analyzed.

This paper differs from previous research in the area by not focusing on the platform as an operating system or as parts of the source code from other software as e.g. Taudes et al. (2000) and Fichman (2004). In this paper the platform is viewed more generally as the underlying fundament that provides the opportunity for developing a new product. In similar way this paper differs from other papers using real options not as a capital budgeting technique, see e.g. Benaroch et al. (2006) for a literature review, but by viewing real options as a flexible method for implementing and executing strategies. The remainder of the paper is organized as follows: Section 2 introduces service platforms and real options and Section 3 discusses the three examples from the video game industry. Section 4 provides the analysis of the three examples and section 5 discusses the implications for software development. The paper ends with conclusions and suggestions for further research in section 6.

2. SERVICE PLATFORMS AND REAL OPTIONS

We define service platform as "a combination of established service processes and service systems within an organization that forms the basis for the development of two or more service concepts." This definition is based on the literature on services and service development and the fact that services are often described as consisting of three parts: service concept, service process and service system (Edvardsson et al, 2000). The essence of a service platform is that it creates conditions for a number of different service concepts. Service concepts show how the company meets a customer and what the customer receives. This is the part that the customer pays for and where the great variety is needed and also where alternatives exists.

Service processes and service systems may with this approach be standardized and made common to a number of different service concepts.

Somewhat simplified it can be said that the service process and service system is uninteresting for the customer. Yet much of the focus of service research has been in these areas (Goldstein et al, 2002).

The definition above means that a service platform may consist of a complex combination of e.g. a process and a number of physical or technical resources. This combination is developed in order to be used alone or as part of a larger process and service system, to develop a number of different concepts and services.

The platform can be developed in two fundamentally different ways. The platform can be developed as such from the beginning or the platform can be identified in an existing service or from a part of a service. In this paper the platforms are generated in the latter way.

A real option can, according to Copeland & Antikarov (2003), be defined as the right, but not the obligation, to undertake an action for the future at a predetermined cost for a predetermined period of time. This definition is similar to the definition given by Dixit & Pindyck (1994) and Amran & Kulatilaka (1998). There are several types of real options suitable for different type of settings; see Copeland & Antikarov (2003) for a discussion about the different types of real options.

However, the common denominator for all different types of real options is that they all can be defined by their financial counterparts, i.e. American call and put options. This simplifies the terminology without loss of generalization and is the terminology used in this paper. An American call option gives the right, but not the obligation, to buy the underlying asset for a predetermined price until a predetermined time. An American put option gives the owner the right, but not the obligation, to sell the underlying asset for a predetermined price until a predetermined time. We will refer to these options as call and put options in the rest of the paper. The value of a real option is therefore depending on the same underlying variables as a financial call option or put option. The reader is referred to e.g. Hull (2005) for the technical discussion about the underlying variables and how the affect the value of the different financial options.

In this paper we will focus on how these options can be planted by the use of platforms and what opportunities it has created for the future actions as well as the effect on the uncertainty in the innovation cycle. The main idea behind an option perspective is to incorporate flexibility in the decision making by realizing that decisions concerning tomorrow is best made based on tomorrow's information, although we must make the decision today. To do so we must move away from static decision making models and plant options to allow for flexibility in the decisions we make today and postpone the final decision until we have the relevant information.

There are three circumstances that must be present in the decision setting for the option view to be relevant according to Dixit & Pindyck (1994); firstly, the investment cost must be (or at least partly) irreversible, i.e. involve a sunk cost. Secondly, there must be a possibility to defer the investment if needed or wanted, i.e. it cannot be a now or never setting. Thirdly, there must be some uncertainty about the future regarding. This uncertainty can be of an inner kind or of an outer kind as well as a combination of the two. Inner uncertainty refers to the cost uncertainty of the production of the product. Outer uncertainty is the measure for the uncertainty about the market conditions for the product.

3. THE THREE VIDEO GAMES

This part of the paper will discus the three video games that will serve as examples for the analysis of how a platform based strategy can be used to manage uncertainties during the development of software innovations. Besides a short description of the different games, the discussion will outline the platforms in the different examples and what type of option that was planted for future actions.

3.1 Sims

The first type of platform discussed is the game concept behind the award winning and commercially successful Sims. The first Sims was released in 2000 and was developed by Maxis and published by Electronic Arts. Since then several expansion packs and sequels have and are planned to be released. For more information concerning Sims the reader is referred to http://thesims.ea.com/

What we refer to as the game concept here is the real-life simulation and the game environments in which different characters play a role in Sims, i.e. the different types of societies where the game is played. The success with Sims gave Electronic Arts the call option to reuse the game concept as a platform for developing new versions of Sims. The sequential versions of Sims also became successes as result of the well established platform. The game concept also included toolkits which gave the player the option to customize the gaming experience (Prügl & Schreider, 2006). These toolkits were developed by Electronic Arts as well as by other players.

3.2 Unreal

The second type of platform discussed in this paper is the award winning game engine Unreal Engine developed by Epic for their first-shooter game Unreal that was released in 1998. The game engine has then been further developed into Unreal Engine 2, Unreal Engine 2.5 and Unreal

Engine 3. For more information about the video game Unreal and the Unreal Engine the reader is referred to http://www.unreal.com/

Given the functionality of the game engine, Epic got a put option to license their game engine as a platform for other studios to use when developing their games. Examples of video games developed with the Unreal Engines, apart from later releases of the video game Unreal, include Medal of Honor, Army of Two as well as cartoon shows as Lazytown on the Nickelodeon channel.

3.3 World of Warcraft

The third type of platform discussed in this paper is the use of extension packs for video games. The example explored here is the various paths, i.e. expansion packs that have been released to World of Warcraft developed by Blizzard Entertainment. World of Warcraft was first released in 2004 with two expansion packs released in 2007 (The Burning Crusade) and 2008 (Wrath of the Lich King). For more information about World of Warcraft the reader is referred to http://www.worldofwarcraft.com/

The platform in this case is the original game itself that provided the call option to expand the game by releasing different paths for the game giving new functionalities by introducing new parts to the game environment.

4. ANALYSIS

Based on Section 2 and 3, this section discusses what kind of uncertainty was reduced by the use of a platform to plant options for future actions. The section is concluded with a discussion of the proposed model in this paper.

The common denominator for all three examples above is that the uncertainty in the development cycle was reduced due to the options planted by the investment made in the platforms. These investments created future opportunities for the firm to exercise. In the Sims example outer uncertainty was reduced since sequential products were based on the same concept that made the original game popular. The new versions were developed given a large customer base with a known preference for the game. By innovating within the boundaries of the game concept, Electronic Arts kept renewing the game given the players preferences. This is not the same as saying that given a strong first release a sequential product should be released but merely that the firm has an option to do so. There are numerous examples of where the sequential video game has failed and the option should not have been exercised since the platform was not strong enough.

In the second example concerning the game engine Unreal, Epic reduced the inner uncertainty for further innovations, not only by using Unreal as game engine for later games, but by using its put option to license the game engine to other studios and thereby partly financing their own development costs. From a different view this also reduced the outer uncertainty for Epic since their games was no longer the only source of revenues. By this, Epic will make money even when their competitors are selling video games given that they are using the Unreal platform. In this case the platform resulted in a broadening of the product portfolio and Epic has now a more diversified product portfolio made up of different types of products.

In the third example about World of Warcraft, Blizzard reduced inner and outer uncertainty. By releasing paths instead of entirely new video games, the installed customer base could be recapitalized and made further locked in. This decreases the uncertainty of losing the customers to a competing game which is sold for a higher price than the path. The inner uncertainty is reduced for at least two reasons. Firstly, most of the inputs to the paths are based upon World of Warcraft and thereby already developed. Secondly, the paths represent smaller developing projects and would therefore require less development funds *ceteris paribus*.

The great difference between the three above examples is what constitutes the platform that provides the option. In the Sims example it is only a part of the product, i.e. the game concept that constitutes the platform that defines the option for future innovations. This is similar to the Unreal example where the development of the video game led to the development of the platform, i.e. the game engine that was used in the game. The great difference of what constitutes the platform becomes clear in comparison with the World of Warcraft example where the entire game is the platform that is reused with the paths. Another difference between the examples is type of option created and the implication of that option.

The options created in the Sims example as well as in the World of Warcraft example are options that have provided the firm itself with the opportunity for sequential investment for new games. The opposite can be seen from the Unreal example where by using the option to license the game engine other firms can innovate using the platform. This strategy has made Epic less sensitive to the outcome of their own games since they made money on the licenses from the game engine. So in this example the platform provided a diversification of the risk rather than only a platform for future innovations for Epic.

Although the above, it should not be concluded that strong platforms always results in successful games and that the options always should be exercised. An example of this is the Godfather game series by Electronic Arts that is based upon the successful film series with the same name. In this example we see the story behind the movie series as the platform and the games as product developed based on that platform. Although the strong story to develop games from, the released games received rather poor reviews. The point to be made here is that no matter how strong the platform is, it only serves as an option to develop a game. It is then up to developers to make the best out of the platform for the final game to became well received, i.e. even though the option creates flexibility, management skills is needed to determine which options to exercise and which not to.

4.1 Model of the proposed strategy

Based on the discussions above, Figure 1 below shows the general model depicting the relationship between platforms, options and software development with reduced uncertainty.

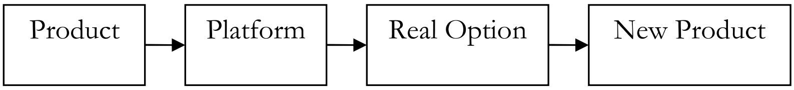

Figure 1. Platform based development strategy

Figure 1 highlights the underlying methodology for software development proposed in this paper whereas Figure 2 below shows the implication of how the platform can evolve during a development cycle and how new options are created that results in different new products. The key issue is that the platform can evolve in different ways depending on the type of products that is developed using it and therefore different options for the future are planted. In Figure 2 this is exemplified by products *A, …, F*. All these products are in one way or another based on that product *A* is developed. Product *B* and *D* are made possible by different parts of product *A*, i.e. there is an option to develop these products based on the platforms from product *A*, as shown by the arrows in Figure 2. These platforms then evolve when products *C, E* and *F* are developed by including more of the former products into the platform.

The point of doing so is to make the development less uncertain and to decrease the development cost and time.

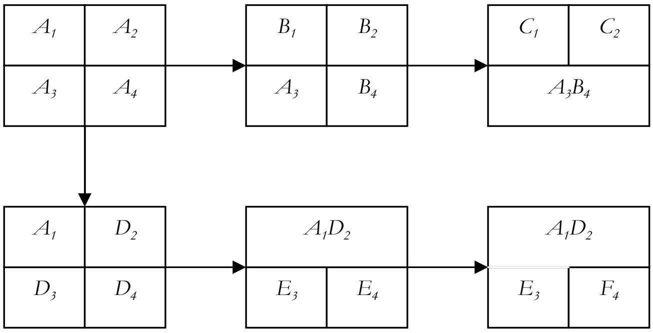

Figure 2. The evolution of the platform in the development cycle

To pinpoint the role of and the relation between the platform and the resulting options, the below distinction is made concerning the type of uncertainty that is reduced by the different types of platforms:

	Type of Platform	
	Concept	Production
Kind of uncertainty managed	Outer	Inner
Options planted	Call	Call/Put

Table 1. The relation between the type of platforms and the of options

From the examples above we distinguish between what constitutes a content platform and a production platform. The former is illustrated in the examples of Sims and World of Warcraft where the platform is made up by the concept of the game, i.e. what the customer (the player) perceives. The second type of platform which is referred to as a production platform is illustrated in the Unreal example. This type of platform does not necessarily influence what is perceived by the customer but how it is produced by the developers. The concept and the production platform play different roles in managing the uncertainty in the development of the product. A concept platform primarily influences the outer uncertainty, i.e. *what* is developed. The concept platform takes advantage of maintaining the consumer preferences from previous products and thereby decreasing the outer uncertainty. A production platform is managing the inner uncertainty, i.e. *how* the product is produced. The production platform reuses production processes and therefore reduces the inner uncertainty.

Finally, we differentiate between the types of options that are created by the platform. A production platform provide the developing firm with a new developing process that they can use themselves in the future but they can also sell the license to others. In this way they can decrease their own development costs but also earn revenues from the platform. The concept platform, on the other hand, primarily creates new opportunities to develop new versions of the product or similar products based on the the original product.

As a final note to the analysis it would be ideal, based upon the proposed strategy in this paper, to develop the platform first since it maximizes the available options for future product development. Although this may not be the typical development strategy we are shortly going to mention one game developed in this way. Skate is a skateboarding game developed by EA Black Box that was released in September 2007. The game was developed around its "flick it" control system that was developed long before the game concept was finalized. The "flick it" control system can be seen as the platform that made big difference between how Skate is played in comparison to other similar games, e.g. Tony Hawk Pro Skate, and what created the intuition behind how the game was played, which made the game into a great success. The game became the best-selling game in its genre and the sequel, Skate 2, has also, much anticipated, recently been released. For more information about Skate the reader is referred to http://www.skate.ea.com/.

5. IMPLICATIONS FOR SOFTWARE DEVELOPMENT

As a result of the analysis we will discuss three implications for software development. Firstly, software developers should develop a more insightful view of what the core product really is that customers, today and in the future, are willing to pay for. As seen in the Unreal case, the core product became the Unreal game engine which made development of new product possible not only for Epic but to other via suing the Unreal engine. The platform thus created the different options for future strategies.

Secondly, the software developer should have a more portfolio based view on their product line when it comes to the development strategy instead of focusing on the individual software, as suggested by Krueger (2006). Beinhocker (1999) discusses Microsoft in the early days as an example of a firm using a portfolio based development strategy for planting options for the future.

Thirdly, developing firms and individual developers shall adopt more evolutionary strategies, as in the Sims example, instead of revolutionary strategies. This gives a more adaptive strategy and decreased the irreversibility of the development and therefore reduces the uncertainty during the development as well as shortening the time-to-market for the software. Another important aspect of evolutionary strategies is that they keep the options alive longer which allows the firm to exercise the options when it is most profitable to do so and when the uncertainty has been minimized.

6. CONCLUSIONS

This paper has shown the advantages of implementing a platform based strategy when developing software and that platform advantages can be explained with option theory. However, it is wrong to say that a platform based development strategy is always successful. This paper has shown that even the use of a strong platform can lead to products that turn out to be less successful.

This paper identifies two different types of platforms depending on the function they play. The two types of platforms identified in this paper are called production platforms and concept platforms. The definition given in this paper of a service platform maps the combination of established service processes and service systems as the basis for the development of new services. This is in line with the production platforms discussed in this paper. The latter type, the concept platforms, identifies the platform in another part of the service, the service concept. This is not how platforms are discussed in previous service platform literature.

As shown in the analysis, the two types of platforms manage different uncertainties in the development. Production platforms reduce the uncertainty in *how* software is developed which was referred to as inner uncertainty in the discussion concerning real options, while a concept platform reduce uncertainty in *what* is developed and how the product is accepted by the market, i.e. the outer uncertainty. As a result there is a difference between which type of options are planted by the two types of platforms. A concept platform would primarily plant call options for future development. A production platform on the other hand may plant call and/or put options depending on how it is managed.

6.1 SUGGESTIONS FOR FURTHER RESEARCH

Based on the conclusions, the two types of platforms have implications for the definition of a service platform. These implications needs to be further analyzed through more research in the aim of finding a unified definition that can include both types of platforms. Moreover, the two types of platforms need to be confirmed by further studies. An appropriate way to do this is through empirical studies of the software development processes. These studies should also take into consideration whether there are differences between how various types of development companies manage their platforms.

REFERENCES

Amran, M. and Kulatilaka, N. (1989) "Real Options: Managing Strategic Investments in an Uncertain World", *Oxford University Press*,

Anadarajan, A. and Wen, H. J. (1999) "Evaluation of Information Technology Investment" *Management Decision*, Vol. 37, No. 4, pp. 329-337

Beinhocker, E. D. (1999) "Robust Adaptive Strategies", *Sloan Management Review*, Vol. 40, No. 3, pp 95-106

Benaroch, M., Lichtnstein, Y. and Robinson, K. (2006) "Real Options in Information Technology Risk Management: An Empirical Validation of Risk-Option Relationships", *MIS Quarterly*, Vol. 30, No. 4, pp. 827-854

Copeland, T. and Antikarov, V. (2003) "Real Options: A Practitioner's Guide", *Texere*, New York

Dixit, A. K. and Pindyck, R. S. (1994) "Investment Under Uncertainty", *Princeton University Press*, Princeton, New Jersey

Edvardsson, B., Gustafsson, A., Johnsson, M. D. and Sandén, B. (2000) "New Service Development and Innovation in the New Economy" *Studentlitteratur,* Lund.

Fagerberg, J. (2003) "Innovation: A Guide to the Literature" Workshop "The Many Guises of Innovation: What we have learnt and where we are heading", Ottawa

Fichman, R. G. (2004) "Real Options and IT Platform Adoption: Implications for Theory and Practice", *Information Systems Research*, Vol. 15, No. 2, pp. 132-154

Goldstein, S. M., Johnson, R., Duffy, J. and Rao, J. (2002) "The service concept: the missing link in service design research?" *Journal of Operations Management,* Vol. 20, No. 2, pp 121-134

Hallikainen, P, Kivijärvi, H. and Nurmimäki, K. (2002) "Evaluating Strategic IT Investment: An Assessment of Investment Alternatives for a Web Management System", *Proceedings of the 35th Hawaii International Conference on System Science*

Hull, J. C. (2005) "Options, Futures and Other Derivatives", *Prentice Hall*, New York

Jacobs, G., Ip, B. (2005) "Establishing User Requirements: Incorporating Gamer Preferences into Interactive Games Design", *Design Studies*, Vol. 26, No. 3, pp 243-255

Johnson, G. and Scholes, K. (2006) "Exploring Corporate Strategy: Text and Cases", *Financial Times/Prentice Hall*, New York

Kreuger, C. W. (2006) "New Methods in Software Product Line Practice – Examining the Benefits of Next-Generation SPL Methods", *Communication of the ACM*, Vol. 49, No. 12, pp. 37-40

Pennock, M. J. and Haimes, Y. Y. (2002) "Principles and Guidelines for Project Risk Management", *Systems Engineering*, Vol. 5, No. 2, pp. 89-108

Prügl, R., Schreider, M. (2006) "Learning From the Leading-Edge Customizers the Sims: Opening up the Innovation Process Using Toolkits", *R&D Management*, Vol. 36, No.3, pp 237-250

Ropponen, J. and Lyytinen, K. (2000) "Components of Software Development Risk: How to Address them? – A Project Manager Survey", *IEEE Transaction on Software Engineering*, Vol. 26, No. 2, pp. 98-112

Schumpeter, J. (1949) "Economic Theory and Entrepreneurial History, Change and the Entrepreneur", in Schumpeter, J. (1989) "Essays on Entrepreneurs, Innovations, Business Cycles and the Evolution of Capitalism", edited by Richard V. Clemence, *Transaction Publishers*, New Brunswick

Taudes, A. and Feurstein, M., Mild, A. (2000) "Option Analysis of Software Platform Decisions: A Case Study", *MIS Quarterly*, Vol. 24, No. 2, pp. 227-243

CHAPTER 8
PAPER 5

Based on Numminen, E., Wrenne, A. (2010) "The Role of Platforms in Software Development – Planting Real Options to Manage Uncertainties", *Proceedings of the 1st International Conference on Information Management and Evaluation*, pp. 289-295

THE ROLE OF PLATFORMS IN SOFTWARE DEVELOPMENT – PLANTING REAL OPTIONS TO MANAGE UNCERTAINTIES

Emil Numminen, Anders Wrenne

School of Management at Blekinge Institute of Technology, Ronneby, Sweden

emil.numminen@bth.se

anders.wrenne@bth.se

ABSTRACT

The innovation process of developing new software is a challenging job and an uncertain process because of the tasks associated with the development. These uncertainties can be categorized based on what drives the uncertainty. In this paper we separate between inner and outer uncertainty. Inner uncertainty stems from how a software product shall be

developed and the cost associated with it. The outer uncertainty stems from what software product shall be developed and the revenues associated with it.

This paper draws on theories from innovation and development research to develop a model to analyze how the uncertainty during a software development project can be managed. In doing so, we take a supply side view on software development where the firm does not merely respond to given market needs in the development cycle but instead plays a more active role. We operationalize the supply side of the innovation process in the software development by developing a model where we analyze how service platforms create real options for future innovations. An empirical study has been conducted to examine whether and how platforms are used in software development to plant options for future innovations as suggested by the model. The study was conducted at a company that primarily develops IT-phone-service products.

This study shows that the company use platforms for the development of their products. Further, it is a prerequisite for developing software at the pace as well as the cost and the quality demanded by their customers. The platforms play different roles in the development depending on the product being developed. The study describes the development of two different products and how the platforms are used in different ways in the development of these products. The differences in the use of platforms is partly because of the different nature of the products but also due to the market maturity of the software, this result in a difference in the kind of real options that is created for the future. The study shows that both inner and outer uncertainties are reduced by the use of platforms in the development phase.

KEYWORDS

Innovation, Platform, Real Option, Uncertainty Reduction, Software Development

1. INTRODUCTION

The role of developing software is challenging due to the different types of uncertainties in the process, see e.g. Tüysüz and Kahraman (2006); Pennock and Haimes (2002); Ropponen and Lyytinen (2000) for a more thorough discussion about this. In this paper, in accordance with e.g. Dixit and Pindyck (1994), we use a general categorization of the different types of uncertainties in a software development project based on what drives the uncertainty. Inner uncertainty is referred to for the activities that drive the cost uncertainty of the project, e.g. the time it takes to develop the software and the cost attached to it. Outer uncertainty is referred to for the activities that drive the market uncertainty for the project, e.g. consumer preferences for the type of software developed and ultimately the profit gained from selling the software.

This paper deals with how these uncertainties can be reduced in the development of software. Based on the recent literature in innovation and strategy research, a model is developed and empirically tested. The basis for the model is to explore how the uncertainties can be reduced in software development. This paper differentiates from previous research by taking a declared supply side of view of development. What we refer to as a supply side view of development is that the firm will not only react to current and expected future costumer preferences, i.e. market needs. The firm will instead actively try to be proactive to form the market that they operate in by trying to stay one step ahead.

The reminder of the paper is organized as follows. The theoretical discussion underlying this study is presented next. This is followed by a presentation of the developed model. The subsequent section will discuss the empirical validation of the model. The paper is concluded with suggestions for further research based on the conclusion of this paper.

2. BACKGROUND TO A TECHNOLOGICAL INNOVATION SYSTEM ON A FIRM LEVEL

The model that is developed in this paper can be seen as an operationalized version of a technological innovation system (Carlsson and Stankiewicz, 1991; Carlsson, 1995, 1997, 2002, 2004) on a firm level in the sense it tries to explain how software innovations come about. A technological innovation system is a part of the experimentally organized economy as described by Eliasson and Eliasson (1996). A technological system consists of three dimensions: 1) the cognitive dimension that defines the clustering of technologies which results in new technological possibilities, 2) the organizational and institutional dimension that defines the actors and the interaction that contributes to the creation of these technologies, and 3) the economical dimension that defines the actors who transforms these technical possibilities into business opportunities. Technology as referred to in this paper is the set of combinatory design spaces formed by clusters of complementary technical capabilities (Stankiewicz, 2002).

Since the above model is mainly valid on a macro level we have to find a way of making the theory suitable on a firm level. We will do so by looking at what role software platforms plays in planting real options for future innovations and to deal with the uncertainty during the experimentation, i.e. development. To make the use of software platforms more general than e.g. Taudes et al. (2000) and Fichman (2004) we will not only refer to source code or other underlying programs, instead we will base it on the concept of service platforms since software has more resemblance to a service than a product, see e.g. Shapiro and Varian (1999) for a greater discussion on the topic.

3. SERVICE PLATFORMS AND REAL OPTIONS

In this paper, we define service platform as "a combination of established service processes and service systems within an organization that forms the basis for the development of two or more service concepts." The definition is rooted in the literature on service management and service development. In that literature a service is often described as the combination of: 1) a service concept, 2) a service process and 3) a service system (Edvardsson et al, 2000). The essence of a service platform is that it creates conditions for a number of different service concepts. Service concepts show how the customer is approached by the firm and also what the customer receives. This is the part that the customer is interested in and where competing firms are offering alternatives.

The service processes and service systems may, given the above, be standardized with the aim of making them common to several different service concepts. The service process and service system are less interesting for the customer than the service concept since this is what the costumer sees and pays for. Yet, the main body of research has focused on these areas (Goldstein et al, 2002). The combination of these is what we refer to as the production knowledge of a service.

The definition above means that a service platform may consist of a complex combination of e.g. a process and a number of physical or technical resources. This combination is developed in order to be used alone or as part of a larger process and service system, to develop a number of different concepts and services.

The platform can be developed in two fundamentally different ways. The platform can be developed as such from the beginning or the platform can be identified in an existing service or from a part of a service. In this paper the platforms are generated in the latter way.

A real option can accordingly to Copeland & Antikarov (2003) be defined in the following way: A real option gives the owner the right, but not the obligation, to undertake an action for the future at a predetermined cost for a predetermined period of time. This definition is also given by Dixit & Pindyck (1994) and Amran & Kulatilaka (1998). There are a number of different types of real options which are suitable for different types of decision settings; see Copeland & Antikarov (2003) for a discussion about this. But, the common denominator for all real options is that they all can be derived from their financial counterparts, i.e. American call and put options.

This simplifies the terminology without loss of generalization in this paper. An American call option gives the right, but not the obligation, to buy the underlying asset for a predetermined price until a predetermined time. An American put option gives the owner the right, but not the obligation, to sell the underlying asset for a predetermined price until a predetermined time. We will refer to these options as call and put options in the rest of the paper. The value of a real option is therefore derived out of the same underlying variables as a financial call option or put option. Since this paper is not about the pricing of real options but how to use real options as a strategic tool in the development of software, the reader is referred to Hull (2005) for a thorough discussion about option pricing.

In this paper we will focus on how these options can be planted by the use of platforms and what opportunities it has created for the future actions as well as the effect on the uncertainty in the innovation of new software. The main idea behind an option perspective is to incorporate flexibility in the decision making by realizing that decisions concerning tomorrow is best made based on tomorrow's information. Real options are relevant given three circumstances according to Dixit & Pindyck (1994): 1) the investment cost must be (or at least partly) irreversible, i.e. a sunk cost, 2) there must be a possibility to postpone the decision if suited, i.e. it cannot be a now or never setting, and 3) there future is assumed to be uncertain. This uncertainty can be of an inner kind or of an outer kind as well as a combination of the two.

4. THE DEVELOPED MODEL

In the firm-level model for the technology innovations system we see the platform as the driver of the cognitive dimension that creates the real options for future innovations for the firm to develop. It can be argued that the platform decreases the time-to-market for the innovation but at the same time it decreases the design space. The main difference from traditional technology innovation systems and the model posed in this paper lies in the two remaining dimensions which are assumed to be within the boundaries of the single firm instead of being a cluster of several actors. The general model as proposed by Numminen & Wrenne (2009) can be illustrated as:

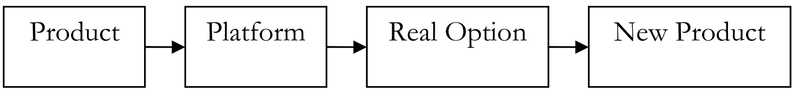

Figure 1. A firm-level technology innovation system

Figure 1 above shows the relation between the platform and the real option and the role it plays in the innovation process concerning new software. In Figure 2 below each matrix illustrates one software product, consisting of a couple of modules. The figure shows how the platform could evolve over time as noted by the horizontal axis and into new software product lines (see e.g. Pohl et al., 2005 for more on software product lines) as noted by vertical axis. It could be argued that the former strategy can be seen as an evolutionary innovation strategy whereas the latter would resemblance a revolutionary innovation strategy better, se e.g. Beinhocker (1999) or Shapiro & Varian (1999) for more on this.

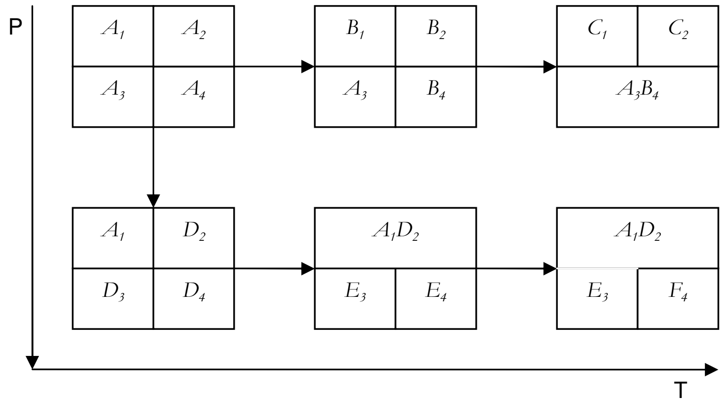

Figure 2. The evolution of the platform in technological innovation system

In Figure 2 above the example concerns six software products in two different product lines that are all in one way or another related to one or more of the four modules developed for software *A*, in the left upper corner. The point of developing in this way is to decrease the uncertainty during development and to decrease the time-to-market for the software.

To clarify the relation between the platform and the resulting real option and what type of uncertainty the below difference is made:

	Type of platform	
	Concept	Production
Uncertainty managed	Outer	Inner
Real option planted	Call	Call/Put

Table 1. The relation between the platform type and the option type

A concept platform is related to what the customer sees and pays for. In software this could be the graphical interface of different software. This is assumed to reduce the outer uncertainty since customers have a lower adoption cost in newer versions. The firm may capitalize on this by developing new software using the same concept, i.e. exercising their call option. A production platform is a combination of processes and systems that are reused in the production of additional software. Given this reuse the inner uncertainty is reduced since the new software is developed given the knowledge and structures from previous development. It could be argued that the firm thereby has a call option to reuse this knowledge when developing new software given that an idea for the new software exists in the same software product family or in a new one. The firm may have a put option if the production knowledge can be packaged and sold or licensed to other firms to use, see Numminen & Wrenne (2009) for a discussion and empirical observations of this.

5. THE EMPIRICAL STUDY

The empirical study was conducted at a company that develops and sells solutions for interaction between IT and telecommunications. The product range consists of speech driven products, contact center solutions and reference systems for companies and organizations. The company is relatively young, with their first delivery in 2002 and has since had a steady growth. The industry is highly competitive and has low margins, which creates high demand on efficient development processes, i.e. high developing phase at low prices. Many of the customers are large companies or organizations, such as telecom operators, call centers, municipalities, hospitals, etc. The customers are often knowledgeable and make great demands on the products. The company was formed by a merger of three different companies. A large part of the ongoing development work is therefore to integrate the products from the three previous companies. The study was conducted through interviews at site with developers, product owners and development manager at the company. In addition, written material was studied to enhance the understanding of the company's development processes and products.

One of the company's competitive advantages is that they are flexible in their development. This means that they have the ability to relatively easily adapt to customers' individual demands. This has always been a deliberate strategy and a way for the company to be competitive in the market against large competitors. The merger has reduced this natural flexibility to some extent, which requires that they are actively working to build in flexibility in the products and processes. The creation of platforms is an important part in accomplishing this.

5.1 The two platforms

The study is focused on the development of two different products based on two different platforms. Because of the complexity of the products and the focus of the paper, we have chosen not to describe the products and its features in detail. In brief, the products can be described as a speech-driven operator and system, including reference and customer service. What is interesting about the products in this study is how they are developed and how they are structured with a special focus on the use of platforms in development. The products are of different nature. One product, which we might call product A has been on the market a rather long time and is considered mature. It has evolved subsequently and is now functioning and adaptation is now confined to small details. The second product, product B, is facing major changes. Development work is broader and includes the essential elements of the product.

Platform is a term used frequently in the company. However, it is a concept that is found not easy to define and describe. It is used in several ways. Platform is used as a selling term meaning the products the company sells. In the industry they operate in, a platform is used as a way to illustrate that they do not sell a single product but a more complex system, to be used by itself or in combination with other products. Platforms are also used synonymously with software product lines to describe the different categories of products of the product line. The company also uses the platform concept technologies, i.e. a third way of looking at platforms.

It is thus clear that the platform concept is relevant to the company and that it is used. The question is how the platforms are used in the development and what the role they play? Do they create opportunities for innovation and reduce the uncertainty of development? What types of platforms have we have identified at the company? Based on our theoretical framework, we have identified platforms of different nature in the company. Both of the studied products are based on a platform, but in different ways.

One product, product A, is based on a platform that does not change very often. The platform is stable and the adjustments are made for different customers at the concept level. As new product versions are released the impact on the platform is normally low. The second product, product B, relates to the platform in a slightly different way. New versions of the product also results in changes in the platform. Large parts of the platform are the same between the products but the platform is still in a developmental stage and therefore not as stable as for product A. It also implies that the boundary between what is product and what is platform is harder to distinguish.

It may seem to be an obvious way to develop software, to base them on a platform. This is true to some extent, but what separates the studied company from its competitors is that they build their own platforms. One of the interviewed product owners claim that *"there is, to my knowledge, no one else than us, that build a custom platform. Most others use standard platforms, which can be bought".* It is no doubt that most of the industry use platforms in the development. It reinforces the idea that the platform has significant advantages for software development and that the issue is not *if* platforms are used but *how* they are used.

The company's choice to build their own platforms instead of using standard platforms is a strategic choice primarily to become more cost-effective, when price pressures in the market do not allow that they are paying for a standard platform. Large volumes will pay the costs of developing their own platforms in the long term. The interviewed product owner continues, *"we also would be able to offer other things and have control in a better way",* on the choice of developing their own platforms. *"We want to be flexible"* is the third way that the product owner justifies the choice of their own platform above standard platforms.

5.2 What does the company achieve with the use of platforms?

"It is extremely important to reduce uncertainty. We spend years on an application. Then to rebuild everything when an external condition changes, it does not work. It takes months or even years to rebuild and we have not won anything" (Product owner/developer of the company)

We have shown that the company uses platforms in the development of their products. It is used in the development of both the studied products, albeit in slightly different level. What is achieved by the use of platforms? The firm deliberately uses platforms when developing products. What we see is that the company through this approach achieves several advantages.

1) The cost of developing new products is reduced. Much of the content of a new product is reused from the software developed earlier. One of the interviewees puts it as follows: *"It´s the same platform with small changes in applications that we deliver to a large, multinational company and a small company with 100 employees."* All development costs, the cost for the hours used to develop the new version but also planning and managing/controlling the development could be reduced if the proportion of reuse is increased. This could be achieved by further development of the platforms so it can be used in additional software product lines.

2) The time for development of software decreases. This allows a new product can reach customers more quickly, which in turn can have positive impact on the revenue side.

3) The quality of the products increases. Large parts of the product, it is included in the platform, has been tested and gradually improved. This means that the risk of bugs in software is reduced which reduces the development cost additionally since you would only include stable modules in the platform.

4) The learning curve for the customers concerning new software is reduced since the customer recognizes the graphical interface from previous versions of the software.

5) The platforms increase the possibilities for flexibility in the development of the products. If you don't have to devote time and resources for

the production of the product, you can add this to find solutions that are unique to the customer. The company claims that one of their major competitive advantages is that they are flexible. It is a relatively small company and as such needs to be easily movable in order to compete with larger competitors. Flexibility is a deliberate strategy and the use of platforms to plant option for future development is one way to achieve this.

Previously in the paper, we argued that the development of software is a process associated with different types of uncertainty and these uncertainties can be categorized into inner and outer uncertainties. What the studied company achieves when using platforms is to reduce both of these categories of uncertainty. They reduce inner uncertainty, i.e. *how* to develop software, by reducing costs. It also reduces the outer uncertainty, i.e. *what* one develops, for example by offering great flexibility to customers.

Another way to explain how uncertainty is reduced is the use of real options, as discussed in the introduction of this paper. With the assumption that a real option is a right but not the obligation to undertake an action in the future we may see that the company through its conscious creation of platforms creates options to develop and sell products in the future. By building platforms they have the ability to develop new products that are personalized to lower cost and in less time than would have been possible without the platforms. They have not the obligation but the right to develop the products. Since there was no way to manage the costs of developing a platform from the ground and then start thinking about developing software that can be sold, the company were forced to develop products to customer at the beginning. That product has since been the basis for a platform, which in turn created the real option to develop the next product. This is illustrated by the general model in Figure 1.

Although the firm acknowledges the flexibility created by an option view on software development they are not as proactive now in planting options as they ones were. Nowadays the time the developers may use to implement anticipated updates is reduced. The update is instead developed for the next scheduled release at a higher cost. Further, they are not as active in developing prototypes for the salesmen to pitch for customers on sales visits. On the other hand they have created an application for their customers to use in creating their own small applications that are not part of the original system delivered. The system can thus be seen as the platform that gives the customer the option to fine-tune it for their specific needs.

We can see that the company's development of the platforms over time is a complex process. The two platforms in this study are in different stages. However they have both been developed in almost the same way. The platforms that the company uses today were developed over a long time and has gradually built up to what they are today. They are loaded with new elements that previously remained at the application level, but which proved to be so general and well-functioning that they are suitable to integrate into the platform. This way to develop platforms is similar to Figure 2 that shows the evolution of the platform in technological innovation system.

6. CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

This paper has shown how a technological innovation system can be applied on a firm-level. To do so the theory has to be changed in the respect to what constitutes the different dimensions. In this paper we have shown that the properties from a service platform combined with the flexibility of real options can be used to do this. The model, and its underlying approach, developed in this paper has been shown to decrease the uncertainty in developing software and is in fact utilized the studied company.

The studied company uses platforms in their product development. The paper has shown that this is a prerequisite for developing software at the pace as well as the cost and the quality demanded by their customers. The company sees the platforms implicitly as options in the development of new products. However, it is difficult to be as proactive nowadays as before because of the severe competition on the market and the rather homogeneous demands from the buyers.

Although the results presented in this paper highlights the role platforms play in the development of software we should be aware that there is more to be done to confirm, clarify and strengthen the results. More research is therefore needed in this area to clarify the role of platforms in software development and what really constitutes a platform for development. One way to do this is to conduct studies in different types of companies. This will give a more diverse picture of the role platforms and options play in software development. Greater knowledge is needed about what types of platforms are being used and how these platforms can contribute to uncertainty reduction. This would create conditions for development of models as in e.g. Numminen & Wrenne (2009) which are further developed in this paper. Better founded models could in turn help industry to manage uncertainties and decrease costs when developing software.

REFERENCES

Amran, M., Kulatilaka, N. (1989) "Real Options: Managing Strategic Investments in an Uncertain World", *Oxford University Press*, Oxford.

Beinhocker, E. D. (1999) "Robust Adaptive Strategies", *Sloan Management Review*, Vol. 40, No. 3, pp 95-106.

Carlsson, B. (Ed.) (1995) "Technological Systems and Economic Performance: The Case of Factory Automation", *Kluwer Academic Publishers*, Boston

Carlsson B. (Ed.) (1997) "Technological Systems and Industrial Dynamics", *Kluwer Academic Publishers*, Boston

Carlsson, B. (Ed.) (2002) "New Technological Systems in the Bio Industries – An International Study", *Kluwer Academic Publishers*, Boston,

Carlsson, B. (2004) "The Digital Economy – What is New and What is Not?", *Structural Change and Economic Dynamics*, Vol. 15, No. 3, pp. 245-264

Carlsson, B., Stankiewicz, R. (1991) "On the Nature, Function, and Composition of Technological Systems", *Journal of Evolutionary Economics*, Vol. 1, No. 2, pp. 93-118

Copeland, T., Antikarov, V. (2003) "Real Options: A Practitioner's Guide", *Texere*, New York

Dixit, A. K., Pindyck, R. S. (1994) "Investment under Uncertainty", *Princeton University Press*, Princeton

Edvardsson, B., Gustafsson, A., Johnsson, M. D. and Sandén, B. (2000) "New Service Development and Innovation in the New Economy", *Studentlitteratur*, Lund

Eliasson, G., Eliasson, Å. (1996) "The Biotechnical Competence Bloc", *Revue d'Economie Industrielle*, Vol. 78, No. 4, pp. 7-26

Fichman, R. G. (2004) "Real Options and IT Platform Adoption: Implications for Theory and Practice", *Information Systems Research*, Vol. 15, No. 2, pp. 132-154

Goldstein, S. M., Johnson, R., Duffy, J. and Rao, J. (2002) "The service concept: the missing link in service design research?", *Journal of Operations Management*, Vol. 20, No. 2, pp 121-134

Hull, J. C. (2005) "Options, Futures and Other Derivatives", *Prentice Hall*, New York

Numminen, E., Wrenne, A. (2009) "Uncertainty Reduction in Software Development by the use of a Platform Based Development Strategy", *Proceedings of the 3rd European Conference on Information Management and Evaluation*, pp. X-XX

Pennock, M. J., Haimes, Y. Y. (2002) "Principles and Guidelines for Project Risk Management", *Systems Engineering*, Vol. 5, No. 2, pp. 89-108

Pohl, K., Böcke, G., van der Linden, F. (2005) "Software Product Line Engineering – Foundations, Principles, and Techniques", *Springer Verlag*, Berlin

Ropponen, J., Lyytinen, K. (2000) "Components of Software Development Risk: How to Address Them? – A Project Manager Survey", *IEEE Transaction on Software Engineering*, Vol. 26, No. 2, pp. 98-112

Shapiro, C., Varian, H. L. (1999) "Information Rules – A Strategic Guide to the Network Economy", *Harvard Business Press*, Boston

Stankiewicz, R. (2002) "The Cognitive Dynamics of Biotechnology and the Evolution of its Technological System", In: Carlsson, B. (Ed.) (2002), "Understanding the Digital Economy: Data, Tools, and Research", *MIT Press*, Cambridge, pp. 35–52

Taudes, A., Feurstein, M., Mild, A. (2000) "Option Analysis of Software Platform Decisions: A Case Study", *MIS Quarterly*, Vol. 24, No. 2, pp. 227-243

Tüysüz, F., Kahraman, C. (2006) "Project Risk Evaluation Using a Fuzzy Analytical Hierarchy Process: An Application to Information Technology Projects*", International Journal of Intelligent Systems*, Vol. 21, No. 6, pp. 559-584

CHAPTER 9
PAPER 6

Based on Numminen, E. (2010) "Why Software Platforms Make Sense in Risk Reduction in Software Development – A Portfolio Theory Approach", *Proceedings of the 4th European Conference on Information Management and Evaluation*, pp. 282-290

WHY SOFTWARE PLATFORMS MAKE SENSE IN RISK REDUCTION IN SOFTWARE DEVELOPMENT – A PORTFOLIO THEORY APPROACH

Emil Numminen

School of Management at Blekinge Institute of Technology, Ronneby, Sweden

Emil.Numminen@bth.se

ABSTRACT

Software development is generally characterized by risk and uncertainty. The uncertainty can be argued to be of two general types, either a technical uncertainty and/or a market uncertainty. From an economical stand point we view technical uncertainty as the cost uncertainty for develop-

ing the software whereas the market uncertainty is the uncertainty regarding the revenues later gained from the sales of the software.

A great deal of research has focused on how to overcome these types of uncertainties by developing models for exploring them with the aim of increasing the ability to manage them. This research has shown that reuse of previous software functionalities in the form of software platforms or as components in software product line engineering can reduce the uncertainty of a software investment. Employment of these strategies can also be seen empirically by studying firms in the software industry.

This paper adds to that research by developing an underlying model that explains why and how platform based developing strategies for software development reduce market uncertainty. The model also explains why certain types of software functionalities are better combined with other types of software functionalities into software platforms in order to reduce market uncertainty. The developed model is based upon portfolio theory which is a branch of financial economics that deals with uncertainty reduction in investments.

The main part of the paper consists of numerical analysis to show why and how the platform strategy reduces the uncertainty in the software development without an equal reduction in the expected return of the investment in the software development. The analysis begins with a decision setting consisting of two types of software functionalities that can be developed. It is shown that different proportions of the two types of functionalities result in different risk and return profiles. It is also shown that the two functionalities should be developed as a platform instead of developing each functionality separately if the market uncertainty is to be reduced.

The analysis is then extended and generalized into a decision setting consisting of several types of functionalities that can be developed and added to the platform. The analysis is concluded with a real option based discussion to explain why the market potential is increased with a platform based developing strategy. The paper ends with a discussion about managerial implications and suggestions for further research based on the presented results in this paper.

KEYWORDS

Software platform, uncertainty reduction, portfolio theory, software development, real options

1. INTRODUCTION

Software development is characterized by risk and uncertainty (Boehm & DeMarco, 1997; Baccarani et al., 2004) where risk is an operationlized measure of the uncertainty. Software development uncertainty can be divided into two general issues; what shall be developed and how shall it be developed. The former type of uncertainty is referred to as market uncertainty, i.e. uncertainty concerning the expected revenues from the software. The latter type is referred to as technical uncertainty, i.e. uncertainty concerning the development cost. The literature on software development uncertainty is vast and the research to find methods to manage these uncertainties has been an ongoing quest for several decades now, see e.g. Lyytinen et al. (1998) for an extended discussion and literature review.

One way of reducing uncertainty in software development discussed to a great extent in the software economics literature is reuse of software components, see e.g. Kreuger (1992) for the history of software reuse. A component based developing strategy will not only reduce the technical uncertainty given that the reused components are tested and their behavior is well known, but may also reduce the market uncertainty since time to market will be decreased (Brereton et al., 2002).

Another development strategy that is based on reusing software components is software product line engineering (Clements & Northrop, 2001) with the aim of leading to the same benefits; see e.g. Sugumaran et al., (2006) for discussion. The difference between the two is that software product line engineering is applied to portfolios of software systems rather than individual software. The common denominator for both of the approaches, i.e. the reuse of software components, can be traced back to a paper presented 1969 by McIlroy (McIlroy, 1969) at the NATO software engineering conference which initiated the research in software reuse.

A third strategy used by firms to reduce the uncertainty for developing software is to do it for the mass-market rather than for specific customers (Brooks Jr., 1987). A version of this strategy is to customize the mass-market developed software. The economic reasoning for doing so that it leads to a higher revenue than selling the mass-market version to the customer in comparison to the cost of customizing the software (Shapiro & Varian, 1999).

To capitalize any of the above discussed strategies there must be an ability to reuse parts of the software. In order to do so the firm must have a deliberate development strategy that supports this. One way of doing so is to develop the software based on a platform consisting of the key functionalities that is demanded by the different software and of the expected future software versions. A key question then becomes which functionalities this software platform should include. The problem is that different software will have different needs of functionalities and different customers will have a different willingness to pay for different functionalities.

From the developing firm's point of view, the question becomes what functionalities should be included in the software platform and what functionalities should be developed for the individual software. In order to reduce the market uncertainty of the line of software that the firm is developing, or is expecting to develop, the firm wants to develop a platform that is most usable for this strategy. From an investment point of view it means that the firms should invest in software platform that minimizes the market uncertainty to the greatest extent. The combinations of software functionalities that accomplish this will be denoted as efficient software platforms. From an economic point of view the firm wants to maximize the return of the investment while minimizing the risk. To do so the firm will invest different amount of money in different functionalities.

This paper will show what type of software platforms decreases the market uncertainty the most and why. The analysis will also show why such software platform is therefore the most suitable software platforms in a reuse development strategy.

The rest of the paper is organized as follows; Section 2 will introduce the theoretical frame of reference used in this paper to develop the model for efficient software platforms. Section 3 will provide a numerical analysis to show that the market uncertainty is reduced by implementing the developed model. Section 4 will discuss the implications of the model and the approach. The paper is concluded with suggestions for further research.

2. THEORETICAL FRAME OF REFERENCE

The theoretical frame of reference in this paper is portfolio theory. Portfolio theory is a branch of financial economics which aims at minimizing the risk of an investment strategy for a given level of expected return by choosing different combination of assets (Constantinides & Malliaris, 1995; Sastri et al., 1997; Elton & Gruber, 1998). The risk is measured by the volatility of the asset, i.e. the standard deviation of the expected or historical return of the asset. The two main variables for achieving the minimization of the risk is which assets one invest in and also the size, i.e. weight, of the different investments in relationship to the total investment. Depending on the return characteristics of the different assets there exist some combinations and weightings of assets that result in lower risk than other alternatives. These are the portfolios that will be held by the investors and are called efficient portfolios.

Given the above, the underlying problem in portfolio theory is to find the weights of the different assets that minimize the risk in the portfolio given the different assets return characteristics.

The development of modern portfolio theory has its roots in the seminal works by Markowitz (1952; 1959), Roy (1952) and Tobin (1958). These results were later generalized and extended e.g. by Sharpe (1964), Lintner (1965), Mossin (1966), Samuelson (1969) Merton (1973) and Ross (1976). Since it is not within the aim of this paper to give an extensive coverage of portfolio theory as such, so the interested reader is referred to e.g. Bernstein (2005) for an intuitive presentation or e.g. Elton et al. (2003) for a more technical coverage.

3. SOFTWARE PLATFORMS AS PORTFOLIOS OF FUNCTIONALITIES

Given the discussion above, a software platform can be seen as a portfolio of different software functionalities. The problem of selecting what functionalities to develop is thus similar to the classic portfolio optimization problem in financial economics in a simplified setting in the way it will be analyzed in this paper. We can therefore use portfolio theory to analyze what constitutes an efficient software platform in order to guide the investments made by the firm. An efficient software platform is, as denoted above, a combination of functionalities that yields a lower risk then other combinations of functionalities for a given level of return. If the firm wants to minimize the market risk for its software development this is the platform they wish to develop and use as an underlying platform for future development of software.

The reminder of this section will show why a platform based development strategy decreases the market risk and how functionalities shall be picked to be included in the platform for the market risk to be reduced. In order to do so, we will use a numerical case where the numbers and functionalities are arbitrarily chosen.

Assume that a firm has two functionalities, X and Y, which it is considering to develop. Theses functionalities may be used in any of three software, i.e. S_1, S_2 and S_3, which are planned to be developed in the future. The functionalities will provide different incremental returns in the software since the different users of the software have different willingness to pay extra for the functionalities. Further, assume that there is a probability of 1/3 that the costumers actually are willing to pay extra for the software given the added functionality, i.e. the firm faces market uncertainty. This probability will be denoted as p and is assumed to be the same for all functionalities. Given the below assumed returns for the three functionalities in the three different software, how shall the firm proceed if it wants to minimize the expected risk?

	p	X	Y
S_1	1/3	10%	5%
S_2	1/3	30%	15%
S_3	1/3	20%	25%

Table 1. The setting for the development of functionality *X* and *Y*

To calculate the volatility of the two functionalities we first need to calculate the expected return for the functionalities which is done using Equation 1 below,

$$E[R_f] = \sum_{i=1}^{N} p_i \times R_i \tag{1}$$

were $R = \dfrac{CF - I}{I}$. The return is thus a measure of difference between the cash flow (*CF*) from the functionality and the investments cost (*I*) for developing it divided by the investment cost to get a relative measure.

The expected return is thus the sum of the different returns times the different markets shares of the software. The expected return for *X* is *20* and *Y* is 20% and 15% respectively. So if the firm can only finance one of the functionalities and wants to maximize the expected return it should develop *X*. The volatility of the expected return from two functionalities is calculated by Equation 2 below given the variance formula in Equation 3.

$$\sigma_{R_f} = \sqrt{\sigma_R^2}\,, \tag{2}$$

where

$$\sigma_{R_f}^2 = \sum_{i=1}^{N} p_i \times \left(R_i - E[R]\right)^2 . \tag{3}$$

The volatility for both *X* and *Y* is 8.16%. Given the expected return and the volatilities of the functionalities the rational choice would be to develop functionality *X* since it provides a higher expected return for the same market risk. What if the firm would instead combine the development of the two functionalities into a software platform by investing in the development of both functionalities? What effect would this have on the expected return and the volatility of the investment?

Assume that the firm invests $a = 70\%$ in the development of functionality *X* and $b = (1 - a) = 30\%$ in the development of functionality *Y* since *X* has a higher expected return. The expected return would then become 18.5% for the software platform given Equation 4 below where *w* denotes the weights in the different functionalities.

$$E[R_p] = \sum_{i=1}^{N} w_i \times R_i \tag{4}$$

The expected return of the platform is thus lower than the expected for functionality *X* but main question is what happens to the volatility of the project. To calculate the volatility of the platform we use Equation 2 once again but have to modify the way we calculate the variance, i.e. Equation 3, of the expected return to include co-variation between the returns of the individual functionalities, see e.g. Falmagne (2003) for the properties of variance calculations. The variance for the expected return of the software platform consisting of functionality X and Y is by Equation 5 below instead,

$$\sigma_P^2 = a^2\sigma_X^2 + (1-a)^2\sigma_Y^2 + 2a(1-a)r_{XY}\sigma_X\sigma_Y, \tag{5}$$

Where $r_{XY}\sigma_X\sigma_Y$ is the covariance between the expected returns of the two functionalities calculated using the correlation (*r*) between the expected returns from functionality *X* and *Y* as defined by Equation 6,

$$r_{XY} = \frac{E[(R_X - E[R_X])(R_Y - E[R_Y])]}{\sigma_X\sigma_Y}, \tag{6}$$

where the numerator of Equation 6 defines the covariance between the returns of functionality *X* and *Y*.

The volatility of the software platform is then 7.26% which is lower than the volatility for either of individual functionalities. This result supports the idea that software platforms do reduce the market risk in software development in comparison to investing in software functionalities separately. Given that the volatility is reduced in the above setting, what is then the optimal investments in *X* and *Y* if the firm wants to minimize the market risk in the investment of the software development? The question is answered by calculating the first-order condition of the variance function, i.e. Equation 5, and set it to zero and solve for the proportion of functionality *X* that minimizes the volatility. By doing so we get

$$\frac{\partial\sigma_P^2}{\partial a} = 2a\sigma_X^2 - 2\sigma_Y^2 + 2a\sigma_Y^2 + 2r_{XY}\sigma_X\sigma_Y - 4r_{XY}\sigma_X\sigma_Y = 0, \tag{7}$$

$$a^* = \frac{\sigma_Y^2 - r_{XY}\sigma_X\sigma_Y}{\sigma_X^2 + \sigma_Y^2 - 2r_{XY}\sigma_X\sigma_Y}. \tag{8}$$

Given the above setting we get $a^* = 50\%$ and $b^* = (1 - a^*) = 50\%$. The expected return of investing accordingly is 17.5% by Equation 4 and a volatility of 7.07% given Equation 2 which is derived using Equation 5 and Equation 6. This is lower than the volatility for the platform when 70% where invested in functionality *X* and 30% invested in functionality *Y*. We therefore conclude that an equal investment in the two functionalities yields the lowest market risk. We will return to discuss the validity of this result and show that 50%/50%-investment is the optimal investments scheme.

3.1 Generalizing the setting

Assume now that the setting for the decision changes and a third functionality is introduced as shown in Table 2 below, does this change the optimal investment scheme derived above?

	p	*X*	*Y*	*Z*
S_1	1/3	10%	5%	40%
S_2	1/3	30%	15%	20%
S_3	1/3	20%	25%	1%

Table 1. The setting for the development of functionality *X*, *Y* and *Z*

The expected return and volatility of functionality *Z* is 20.33% and 15.92% calculated by Equation 1 and Equation 2, given Equation 3. Functionality *Z* has thus, more or less, as high expected return as functionality *X* but almost twice as high volatility. So based upon a risk-to-reward ratio analysis functionality *Z* is inferior to both functionality *X* and *Y* and would not be prioritized in the development. Is there a reason for the firm to consider functionality *Z* since it is inferior to the other two functionalities?

To find out assume that the firm invest 35% in the development of both functionality X and Y, and 35% in the development of functionality Z given the high expected return of functionality Z. Then the expected return of the software platform is 18.35% using Equation 4 and the volatility is 2.54% using Equation 2 and substituting Equation 5 into Equation 9 below to take into account the co-variation between the expected return of all three functionalities. In that way Equation 9 can be seen as an extension of Equation 5, see Copeland et al. (2005) for a derivation.

$$\sigma_P^2 = a^2\sigma_X^2 + b^2\sigma_Y^2 + c^2\sigma_Z^2 + 2abr_{XY}\sigma_X\sigma_Y + 2acr_{XZ}\sigma_X\sigma_Z + 2bcr_{YZ}\sigma_Y\sigma_Z \tag{9}$$

Although the functionality Z on its own is not a preferred investment alternative it is does reduce the volatility of the platform and since individually it has the highest expected return, it also increases the expected return of the portfolio. As in the case of two functionalities, we can solve for the optimal proportions of X, Y and Z. However, since we cannot rewrite the variance function as dependent of only one of the weights, we solve for the optimal weights using the Lagrange multiplier for the variance-covariance matrix for the expected returns of the functionalities. We therefore get after imposing the condition that $a + b + c = 1$, i.e. the proportions of the investments must equal one,

$$g(a, b, c, \lambda') = \sigma_P^2 + \lambda'(1 - a - b - c), \tag{10}$$

or in matrix form

$$X^T AX + \lambda'(1 - a - b - c) = g(a, b, c, \lambda') \tag{11}$$

where X denotes the weight vector of the investments made in the three functionalities and ^{T} denotes the transpose of the vector. A denotes the variance-covariance matrix and λ denotes the Lagrange multiplier. Given the above setting we get can solve for the weight of X, Y and Z by equaling the below first-order conditions to zero

$$\frac{\partial g}{\partial a} = 0, \tag{12}$$

$$\frac{\partial g}{\partial b} = 0, \tag{13}$$

$$\frac{\partial g}{\partial c} = 0, \tag{14}$$

$$\frac{\partial g}{\partial \lambda'} = 0. \tag{15}$$

Equation 12 to 14 stated in a matrix form gives

$$AX = \lambda e \text{ where } e = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \text{ and } \lambda = \lambda'/2. \tag{16}$$

We can now solve the system of linear equations for the weights using Cramer's rule, i.e.

$$X_i = \frac{\left|\hat{A}_i\right|}{|A|}, \tag{17}$$

where $\left|\hat{A}_i\right|$ is the matrix obtained from A by replacing the i th column with the constant vector and $|A|$ is the determinant of the matrix, see e.g. Sydseater & Hammond (2002) and Sydseater et al (2005) for the method. The proportions of functionality *X*, *Y* and *Z* that minimizes the market risk for the software platform is then $a = 1.24\%$, $b = 65.17\%$ and $c = 33.70\%$ given two decimals. The expected return and volatility for this portfolio is 16.88% and the 0.009%, respectively. The volatility for the three-functionality platform is thus lower than the volatility for the two-functionality software whereas the expected return is slightly less due to the higher proportion of functionality *Y*. We therefore conclude that adding additional functionalities to the software platform will decrease the volatility its volatility further. We will return to the circumstances for this below.

3.2 Optimality

The above analysis has shown that a software platform will reduce the market risk. This section will discuss how general these results are. The answer to this lies in Equation 6, i.e. the correlation coefficient for which $-1 \leq r \leq 1$ holds. As long as the added functionality has a correlation below one with the expected return of the platform the risk will be reduced for every functionality that is added. The greatest risk reduction is gained when the functionality has a negative correlation with the expected return from the platform.

In the above analysis we used the Lagrange multiplier explicitly or implicitly and treated the problem as a linear programming problem by imposing the condition that proportions of the functionalities sums up to one. From an empirical point of view it is more relevant to impose the condition that $a, b, c \geq 0$ which states that the investment in a functionality is non-negative, i.e. we do not allow short positions in any of the functionalities. The problem then becomes a nonlinear programming problem and for the solution to be optimal it must satisfy the Kuhn-Tucker conditions

$$\frac{d\sigma_P^2}{dX_i} + U_i = 0, \qquad (18)$$

$$X_i U_i = 0, \qquad (19)$$

$$X_i \geq 0, \qquad (20)$$

$$U_i \geq 0 \ . \qquad (21)$$

In the above equations, *a*, *b* and *c* are denoted by *X* to simplify the notation and *U* is the slackness that is imposed to get an equality in the problem, see e.g. Sydseater & Hammond (2002) and Sydseater et al (2005) for more on non-linear programming. Our solutions satisfies Equation 18 through 21 since *a*, *b*, and *c* are non-negative and *U* is zero.

We know that we have a strictly convex function and therefore and therefore solve for a local minima since

$$\frac{\partial^2 \sigma_P^2}{\partial a^2} \geq 0,\ \frac{\partial^2 \sigma_P^2}{\partial b^2} \geq 0,\ \frac{\partial^2 \sigma_P^2}{\partial a^2}\frac{\partial^2 \sigma_P^2}{\partial b^2} - \left(\frac{\partial^2 \sigma_P^2}{\partial ab}\right)^2 > 0, \qquad (22)$$

$$\frac{\partial \sigma_P^2}{\partial a}\frac{\partial \sigma_P^2}{\partial b} - \left(\frac{\partial^2 \sigma_P^2}{\partial a \partial b}\right)^2,\ \frac{\partial \sigma_P^2}{\partial a}\frac{\partial \sigma_P^2}{\partial c} - \left(\frac{\partial^2 \sigma_P^2}{\partial a \partial c}\right)^2,\ \frac{\partial \sigma_P^2}{\partial b}\frac{\partial \sigma_P^2}{\partial c} - \left(\frac{\partial^2 \sigma_P^2}{\partial b \partial c}\right)^2 > 0.$$

(23)

The case where the software platform consists of two functionalities satisfies Equation 22 and the three functionality case satisfies Equation 23. Since the variance function is quadratic and strictly convex it can only have one minima so the local minima we solve for must also be the global minima. The above is verified by the positive eigenvalues for the variance matrix. Therefore, there will always exist two or more different weights of the investments in the functionalities that result in the same volatility depending on the number of functionalities in software platform. The difference between them is that one will yield a higher expected return and it is therefore these weights that the firm shall invest accordingly to.

This is depicted in Figure 1 below where $a \in (0,1)$ and $b = 1 - a$ for the two-functionality software platform and $a, b, c \in (0,1)$ and $a + b + c = 1$ for the three-functionality software platform. It shall be noted that the graph for the three-functionality software platform is concentrated around the optimal investment from which the weight of one functionality at the time is increased in order to have the graph two-dimensional and this is why not all return/volatility options are depicted The firm will therefore invest in the optimal software platforms as long as the firm chooses proportions of the functionalities north-east of the minimum variance platform. If the firms want the lowest market risk for the software platform it will stick with investing according to the minimum variance proportions, as shown in Figure 1 below for the two cases analyzed in this paper.

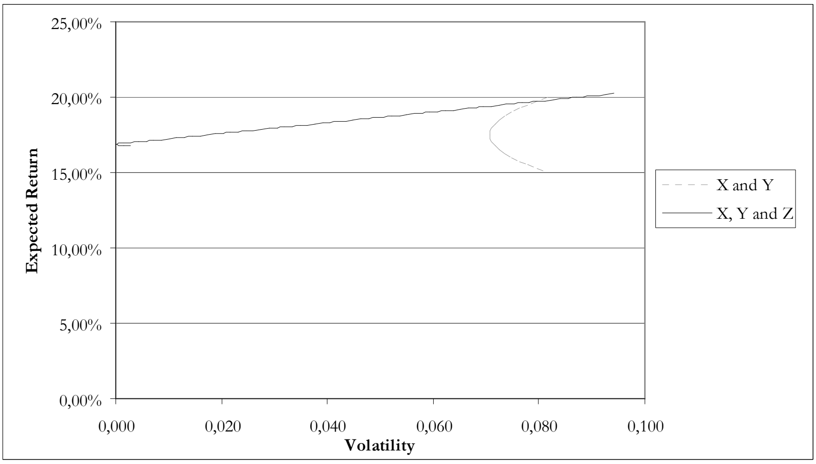

Figure 1. Expected return and volatility for the two platforms

Figure 1 above also shows the main result of this paper which is that combining functionalities into platforms reduces the market uncertainty, the more functionalities that are combined the more is the uncertainty reduced. This can be seen by comparing the curves for the two-functionality platform and the three-functionality platform. For this result to hold the return characteristics of the functionalities cannot be perfectly positively correlated and the more negatively correlated the functionalities are the greater is the diversification effect.

4. DISCUSSION

The above analysis has shown that combining individual functionalities into software platforms do reduce the market risk faced by the developer. This result holds for any functionality as long as the correlation between the incremental expected return from the added functionality and the platform is below one. If this condition is met, the added functionality will keep on decreasing the market risk of the software portfolio. If the return characteristics of the functionality are perfectly positively correlated with the platform return characteristics it is, from a risk point of view, better to develop the functionality for an unique software and not including it in the software platform.

In order for the correlation to be below one it must increase the market for the software in such a way that new costumer preferences are capitalized. It is therefore beneficial from a risk management point of view to include functionalities that may not be used directly but may be needed by future software. The firm will thus profit from planting options for the future today in the form of a reduced market risk for the functionalities that are needed today.

For the firm to capitalize the most from the software platform it has to view functionalities not only as source code. Functionalities have to be viewed as the pieces of the puzzle that creates value for the buyer. By having this broader view it also becomes easier to find functionalities can be combined in order to reduce the market risk since the entire experience will be taken into account.

As a final note to the above analysis something has to be mentioned about the underlying assumptions of the approach in this paper and its limitation. The model is based on the expected return of the different functionalities where the cost of development is implicitly included. The firm can therefore only get risk reduction shown above if the development cost can be covered given the optimal proportion for the functionality, i.e. if the revenues in the return are large enough. If not, the firm cannot achieve the cost coverage directly and should consider the incremental cost as the cost for planting an option for future development or deviate form the optimal risk reduction schedule.

The second underlying assumption for the modeling in this paper is the market type that the developing firm is operating under. The modeling in this paper is only relevant for competitive firms. If the firm has a monopoly power in its market there exists no market uncertainty to reduce. The monopoly firm can always control the demand for its software by using revolutionary developments strategies (Beinhocker, 1999; Shapiro and Varian, 1999). This could be done by not including backward compatibility in the new version or to terminate support for previous versions when new versions are released to persuade customers to adopt the new version.

4.1 Suggestions for further research

The models and analysis in this paper are derivations for risk reduction in fund management. So it is relevant to test them empirically to see how they work in the setting of software development. The key issues to study are whether the approach is applicable and useful for managing the market risk in software development. It may not be relevant for all firms to implement the full modeling of this paper. So it is relevant also to take a more general view of *if*, *how* and *why* firms use software platforms.

If empirical studies show that the approach is not applicable in the setting it ought to provide relevant input of how the approach can be further developed for it to be applicable and useful for risk reduction in software development.

REFERENCES

Baccarani, D., Salm, G., Love, P. E. (2004) "Management of Risk in Information Technology Projects", *Industrial Management & Data Systems*, Vol. 104, No. 4, pp. 286-295

Beinhocker, E. D. (1999) "Robust Adaptive Strategies", *Sloan Management Review*, Vol. 40, Vol. 3, pp. 95-106

Bernstein, P. L. (2005) "Capital Ideas – The Improbable Origins of Modern Wall Street", John *Wiley and Sons*, Hoboken

Boehm, B. W., DeMarco, T. (1997) "Software Risk Management", *IEEE Software*, Vol. , No. , pp. 17-19

Brereton, P., Linkman, S., Thomas, N., Bøegh, J., De Panfilis, S. (2002) "Software Components – Enabling a Mass Market", *Proceedings of the 10th International Workshop on Software Technology and Engineering Practice (STEP'02)*, pp. 169-176

Brools Jr., F. B. (1987) "No Silver Bullet Essence and Accidents in Software Engineering", *Computer*, Vol. 20, No. 4, 1-16

Clements, P., Northrop, L. (2001) "Software Product Lines: Practices and Patterns", *Addison-Wesley Professional*, New York

Copeland, T. E., Weston, F. J., Shastri, K. (2005) "Financial Theory and Corporate Policy", *Addison-Wesley*, Boston

Constantinides, G. M., Mallaris. A. G. (1995) "Portfolio Theory" in Jarrow, R. A., Maksimovic, V., Ziemba, W. T. (Eds.) (1995) "Handbooks in

Operations Research and Management Science Volume 9: Finance", *Elsevier Science*, New York

Elton, E. J., Gruber, M. J. (1998) "Modern Portfolio Theory, 1950 to Date", [online], NYU Working Paper No. FIN-98-026, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1297051 2009-12-12

Elton, E. J., Gruber, M. J., Brown, S. J., Goetzmann, W. N. (2003) "Modern Portfolio Theroy and Investment Analysis", *John Wiley and Sons*, Hoboken

Falmagne, J-C. (2003) "Lectures in Elementary Probability Theory and Stochastic Processes", *McGraw-Hill*, New York

Kreuger, C. W. (1992) "Software Reuse", *ACM Computing Surveys*, Vol. 24, No. 2, pp. 131-183

Lintner, J. (1965) "Security Prices, Risk, and Maximal Gains from Diversification", *Journal of Finance*, Vol. 20, No. 4, pp. 587-615

Lyytinen, K., Mathiassen, L., Ropponen, J. (1998) "Attention Shaping and Software Risk – A Categorical Analysis of Four Classic Risk Management Approaches", *Information Systems Research*, Vol. 9, No. 3, pp. 233-255

Markowitz, H. (1952) "Portfolio Selection", *Journal of Finance*, Vol. 7, No. 1, pp. 77-91

Markowitz, H. (1952) "Portfolio Selection: Efficient Diversification of Investments", *John Wiley and Sons*, New York

Merton, R. C. (1973) "An Intertemporal Capital Asset Pricing Model", *Econometrica*, Vol. 41, No. 5, pp. 867-887

Mossin, J. (1966) "Equilibrium in a Capital Asset Market", *Econometrica*, Vol. 34, No. 4, pp. 768-783

Ross, S. A. (1976) "The Arbitrage Theory of Capital Asset Pricing", Journal of Economic Theory, Vol. 13, No. 3, pp. 341-360

Roy, A. D. (1952) "Safety First and the Holding of Assets", *Econometrica*, Vol. 20, No. 3, pp. 431-449

Samuelson, P. (1969) "Lifetime Portfolio Selection by Dynamic Stochastic Programming", *Review of Economics Statistics*, Vol. 57, No. , pp. 239-246

Sastri, S. R. S., Mohanty, S., Rao, K. K. Elton, E. J., Gruber, M. J. (1997) "Modern Portfolio Theory, 1950 to Date", *Journal of Banking and Finance*, Vol. 21, No. 11, pp. 1743-1759

Shapiro, C., Varian, H. (1999) "Information Rules – A Strategic Guide to the Network Economy", *Harvard Business School Press*, Boston

Sharpe, W. F. (1964) "Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk", *Journal of Finance*, Vol. 19, No. 3. pp. 425-442

Sugumaran, V., Park, S., Kang, K. C. (2006) "Software Product Line Engineering", *Communications of the ACM*, Vol. 49, No. 12, pp. 29-32

Sydseater, K., Hammond, P. (2002) "Essential Mathematics for Economic Analysis", *Prentice Hall*, London

Sydseater, K., Hammond, P., Seierstad, A., Strom, A. (2005) "Further Mathematics for Economic Analysis", *Prentice Hall*, London

Tobin, J. (1958) "Liquidity Preference as Behavior Towards Risk", Review of Economic Studies, Vol. 25, No. 2, pp. 65-86

CHAPTER 10 CONCLUDING DISCUSSION

In this chapter we will extend and generalize the discussions of the contributions of the six papers. Suggestions for further research will be discussed based upon the findings in this study. We will begin the chapter by summarizing the main aspects of the six papers and the main conclusions from this dissertation.

POSITIONING THE STUDY

The purpose of this study, as introduced in Chapter 1, is to "explore how the economic return of a software investment can be assessed and managed". More or less the same problem has been studied for more than 55 years from different theoretical perspectives using various approaches as reviewed by Banker Kauffman (2004). Different research has focused on various aspects of this based on different aims of the research. Some research has focused on the macro-economic effects of software investments (e.g. Aral et al., 2006; Brynjolfsson, 2003; Dedrick et al., 2003) whereas other research has focused on the organizational impact of and changes due to software investments (e.g. Orlikowski, 2009; 2007; Orlikowski and Scott, 2008). On the organizational level research has also focused on user adoption issues (e.g. Davidson and Heslinga, 2007 Venkatesh et al., 2003; Venkatesh and Davis, 2000; Fichman and Kemerer, 1999). The level of analysis for this study is the micro-economic level, i.e. the firm level. Previous research on the micro-economic level can be divided into two streams. One stream has focused on the *a posteriori* evaluation of the effect of software investments (e.g. Standing et al., 2006; Gottshalk and Karlsen, 2005; Dehning et al., 2007). The second stream of research, which this study belongs to, have addressed the problem of *a priori* assessing the economic return of a software investment (e.g. Chaou et al., 2006; Irani; 2002; Irani and Love, 2002).

This study differs from previous research using the value-maximization proposition as the basis for decisions of which investments the firm shall accept or reject. In order to know which software investments to undertake we need to incorporate the intangible consequences into the cash flow function, a problem that previous research has not solved. If this is not solved, the basis for the decision will not reflect the true return of the software. Hence, we do not know whether we have made a rational decision or not. Finally, since the decision is made *a priori* it includes uncertainty which we need to incorporate ways to assess and manage the uncertainty in the decision.

SUMMARY OF THE STUDY AND CONTRIBUTIONS

The problem

In this study we have analyzed how the economic return of a software investment can be assessed and managed. This has been argued to be difficult, impossible or even pointless in previous research due to the characteristics of a software investment (e.g. Rehesaar and Mead, 2005; Irani, 2002; Banker et al, 1993; Ashford et al, 1988). The foundation for this view stems from the problem of defining a cash flow function for a software investment and the inherent uncertainty in a software investment. The problem of defining a cash flow function is due to the consequences that a software investment results in (e.g. Fichman, 2004; Halikainen et al., 2002; Gunasekaran et al., 2001). These consequences are claimed to be mainly intangible and are not easily to measure in time-dependent economic terms, i.e. cash flows. This results in uncertainty concerning the cost and benefit from a software investment (e.g. Lee and Kim, 2006; Love et al., 2000). The uncertainty also extends to the development of software and especially if the developing firm is a non-monopoly firm. In this setting the firm needs to take into account not only the technical uncertainty, i.e. cost uncertainty, of development but also the market uncertainty (e.g. Day et al., 2007; Liu and Mintram, 2005).

In the six papers which Chapter 4 to Chapter 9 is based on we analyze the various parts of the above problems from an economic perspective in the attempt to develop a more comprehensive solution. To do so the main constructs of economic value from a software investment are analyzed. It is also studied how various constructs create value and how these can be managed to maximize the economic return of the software investment. The main features of the studies that make up the main part of this dissertation will be summarized shortly here to show how they are related to previous research and the solution of the overall problem.

A static solution to the problem

The main focus in Chapter 4 is the analysis is how the economic return of a software investment can be assessed in the case of no managerial flexibility. Underlying the modeling in the chapter is the analysis of why intangible consequences pose a problem in assessing the economic return of a software investment. As a solution to the problem, given the stated goal function for the firm, a basic categorization is developed based on how consequences affect the cash flow from the investment. Based on this analysis it is argued for a stochastic cash flow function to incorporate the relevant economic characteristics from the consequences of a software investment when modeling the economic return.

The above adds to previous research (e.g. Love et al., 2005; Milis and Mercken, 2004; Fichman, 2004; Irani, 2002) in two ways. The first contribution is the basic categorization of consequences from a software investment. Having a basic categorization that captures the relevant economic characteristics of a software investment makes it easier to define a relevant cash flow function. The second contribution is the analytical model for assessing the economic return from a software investment. The model is based on a stochastic cash flow function to incorporate the argued economic characteristics of the consequences.

Uncertainty reduction and software platforms

In Chapter 7 and Chapter 8 we study the role played by software platforms in managing uncertainty in software development. Depending on which type of software platform is created the main type of uncertainty that is managed differs. In the analysis we differentiate between a "concept platform" which is mainly used to manage market uncertainty and a "production platform" which is mainly used to manage cost uncertainty in development. A second aspect of software platforms is also studied the chapters. We analyze what types of options are created by the two types of software platforms. It is concluded that concept platforms mainly creates call-style options for future development whereas production platforms can create call-style options as well as put-style options for the firm. The models underlying the analysis in the chapters are mainly developed in Chapter 7 and the discussion of the empirical study is presented in Chapter 8.

The role of software platforms in uncertainty reduction has been studied previously in different ways in e.g. Pohl et al. (2005), Fichman (2004), Birk et al. (2003), Clements and Northrop (2002), Taudes et al. (2000). The result from Chapter 7 and Chapter 8 contributes to this research in two ways. Previous research has not separated different types of software platform and the type of uncertainty which is managed by each type as done in Chapter 7 and Chapter 8. The second contribution from the chapters is the analysis of the relation between the different types of software platforms and the types of options they create.

The efficient software platform

Chapter 9 further develops the analysis of the role played by software platforms in uncertainty reduction in software development. This is done by applying portfolio theory to model how software platforms should be constructed and what constitutes an efficient software platform for reducing the uncertainty in software development the most. The discussion of the relation between a software platform and options created is extended to conclude that the efficient platform is also the platform that creates the most valuable options for the future.

Chapter 9 adds to previous research by developing a model that shows why platforms reduce uncertainty in software development. It is also analyzed what constitute an efficient software platform and the role it plays to create options.

Real options and managerial flexibility

The focus in Chapter 5 and Chapter 6 is to analyze how real options contribute to reduce the uncertainty in the cash flow from a software investment. It is shown that the managerial flexibility created by real options to handle different types of uncertainty is depending on the type of option being created. The chapters differ in the type of setting analyzed; Chapter 6 analyzes the value created by options in different decision settings when the software investment is acquired. The example used in the chapter is open source software but it is also discussed how to generalize the results to the case of proprietary software. The analysis in Chapter 5 concentrates on the value created by different options when developing software and the empirical circumstances when these options can be created. A common feature of both papers is the analysis of the value added by the real options.

Applying option theory into software investment decisions is being done since the seminal papers Clemmons (1991); Dos Santos (1991) and Kambil et al (1993). Later studies include e.g. Benaroch et al. (2007); Benaroch et al. (2006); Fichman (2004); Schwartz and Zozaya-Gorostiza (2003); Erdongmus (2002); Taudes et al. (2000). The common denominator for much of the research that the above exemplifies in software economics is that it is either based on analytical continuous-time models or analyzes cases given the existence of an option. Analytical continuous-time models make it easy assess the value of the option but are based on restrictive assumptions that may not be empirically validated. The results in Chapter 5 and Chapter 6 add to this research using discrete-time models for the modeling underlying the analysis in the chapters. The discrete-time models used have the advantage of being based on fewer assumptions and hence more applicable in real-life. Further, the chapters add to the research in discussing how various options can be created in different settings.

We shall return to the use of real options when extending the discussion of intangible consequences below.

CONCLUSIONS

The main conclusion of this dissertation is that the return of a software investment can be assessed and managed by applying the models and methods proposed in this study. It is shown that applying a real option and portfolio theory approach provides a more comprehensive approach compared to previous research. This different approach yields the method to manage the uncertainty of a software investment and create opportunities for future development and hence maximizes the return of the investment. It is also shown that it is possible to define a cash flow function for a software investment by analyzing the economic characteristics of the consequences from a software investment. By knowing these characteristics we can use appropriate methods for modeling their economic impact on the cash flow. So, by using more sophisticated economic models the return of a software investment can be assessed and strategically managed.

Much of the discussions throughout the study have assumed that it is a one-firm setting that is analyzed. However, Equation 2 in Chapter 1 is not only valid for a one-firm setting but also for the case when the developing firm and the acquiring firm is not the same. Hence it can be seen as a general economic return function for a software investment.

On a general level the main contribution of this study is that is shown how the economic return of a software investment can be assessed and managed. To do this a more comprehensive approach must be applied in comparison to previous research.

DISCUSSION

This section will extend the analysis of the findings on how the various parts affect the return of a software investment. It will also be discussed how the various results affect the economic return of a software investment as assessed by Equation 2 in Chapter 1.

Value added by software platforms

Chapter 7, Chapter 8 and Chapter 9 have shown that a firm can reduce the uncertainty when developing software. By reducing the uncertainty, the opportunity cost of capital, i.e. k. in the numerator of Equation 2 in Chapter 1, is reduced since k is determined by the relation between the systematic risk of the investment and the market portfolio, i.e. a perfect portfolio; see e.g. Elton et al. (2006); Copeland et al. (2005) or De Matos (2001) for a thorough discussion and a derivation of this result. This means that the expected value of future cash flow increase in value and as a result the return of the software investment increases in value. This result is regardless of if it is the market uncertainty or the technical uncertainty that is reduced. By using a software platform as a base for developing software, we can reduce the investment cost for the development, i.e. I in Equation 2 in Chapter 1, which increases the return of the investment.

Another advantage of a software platform is that it creates option for future development given the reuse of already developed components. Depending on the type of platform developed, different types of options are created and different types of uncertainty can be managed. This will also have a positive effect on the return of the software investment since these should be viewed as sequential investments. So, software platforms do not only reduce uncertainties but may also create valuable investment opportunities as seen in Chapter 7. This further increases the value of the software investment since it has a positive effect on O in Equation 2 in Chapter 1.

Software platforms and general purpose technologies

In order to reduce uncertainty to increase the value of the investment, the firm must have a deliberate strategy when developing the software platform. This issue is the focus of the analysis in Chapter 9. To decrease the uncertainty the firm shall develop a software platform that suits to the most consumer preferences. The intuition behind this is that the platform is then the most usable for developing software from. If a functionality does not reduce the market risk of the platform, it is, from a uncertainty reduction point of view, better to develop that functionality for a unique software instead of including it in the software platform. This result can be extended to the analysis of the technical uncertainty as well. The platform that would appeal to most consumer preferences also minimizes the variance of the return in a general setting. This platform is the platform that we refer to as the efficient software platform.

The result is in line with the development of a general purpose technology. Given general purpose technologies, it may not be the number of functionalities that has the greatest impact on the risk reduction but the type of functionality included. As an analogy of this we may think of electricity being one product with many ways of use depending on our preferences.

Another way of viewing an efficient platform is to consider it as the optimal bundling scheme where bundling refers to packaging *n* products in fixed proportions (Shapiro and Varian, 1999). We conclude that the minimum-variance software platform is the efficient bundling scheme since this platform is made up of different proportions of *n* functionalities which minimizes the volatility of the software platform by appealing to the most customer preferences

Software platforms and managerial flexibility

By developing the software platform the firm also acquires put and/or call options on that platform for future actions. In most cases this would be a call option. However, there are also cases where the firm acquires a put option from the platform, see Chapter 6 and Chapter 7 for discussions of empirical results on this. The call option would in most cases refer to the option to reuse the software platform in future software development. For the option to be as valuable as possible the software platform must meet as many needs as possible. Since the minimum-variance software platform is the platform that appeals to the most consumer preferences it is the software platform that creates the most options for the future.

It is shown in Chapter 6 and Chapter 8 how real options can be incorporated on a general level into software development project and the added return from this, i.e. the effect on *O* in Equation 2 in Chapter 1. The key is that the flexibility can be included in most software development projects. The way to do so is in most cases to change how we view the decision setting under which the software is developed. Given that the cash flow function of software investment is stochastic (Numminen 2008a) there is more need for incorporating options to manage negative outcomes. Doing so we can make the cash flow easier to predict and less uncertain and which gives us a lower discount rate since the required risk-premium is reduced.

Although options for future action can be incorporated in the decision setting it does not mean that all options shall be exercised. Before exercising the option it must be assessed that the value added by the option exceeds the exercise cost. In the end, the software investment must still have a positive economic return when the added values by the options are taken into account.

Real options do not only add value or reduce uncertainty during the development of software. An acquiring firm can also benefit from creating options to maximize the uncertain economic return from the software investment. One way of doing so is to invest in module-based software when possible. Doing so creates valuable flexibility in the adoption of the software compared to acquiring the full system at once. Acquiring additional modules can be seen as options to expand the software. These options shall only be exercised if the expected added economic return exceeds the exercise price, i.e. the investment cost. A module-based investment strategy also reduces the uncertainty of the economic return in comparison to acquiring the full system at once due to the managerial flexibility. This managerial flexibility also derives value from the possibility to defer investments in additional modules until needed.

An option view on intangible consequences

As discussed in Chapter 1 previous research has argued that the true value of a software investment is not possible to asses using traditional investment models. A main reason for this is the presence of the intangible consequences that make up an important part of the value of a software investment. To overcome this problem a different view on intangible consequences is needed. Instead of viewing intangible consequences as a problem that makes valuation of software investments pointless, they should instead be viewed as additional sources of economic value that has to be realized, i.e. an option for future cash flows. As with any option they require an effort to be exercised, i.e. there is an exercise price for killing the option and receiving the underlying asset.

The line of reasoning above is in line with the conclusions in e.g. Brynjolfsson (2003) and Brynjolfsson et al. (2003) where the full economic return from a software investment is reached when it is combined with an organizational change, i.e. the organizational change is the independent variable that separates the economic return that different organizations have received from a software investments. This is also validated empirically by e.g. Stratopoulus and Dehining (2000) where evidence is shown that it is not the amount of money that is invested in software that make the difference but how the software is managed by the organization where it is implemented. Similar results but from another point of view can also be found in e.g. Orlikowski and Scott (2008) and Orlikowski (2007; 1992) where it is argued from an organization theoretical point of view that it is meaningless to separate the software and the organization where it is used since it is value of the co-effect that can be observed. On a general level this is equivalent of saying that you cannot separate the usage of the software from the software when it comes to the creation of economic return.

The role of managerial skills

The prerequisite for an organization to fully capitalize on a software investment is by the above discussion the managerial skills of how to utilize the software in the organization. A part of the managerial skill is to understand of which complementary investments must be made to gain the maximum economic return from a software investment. The same software can be used by different organizations with different economic returns as a result depending on the firm-specific settings (Peteraf, 1993). The more general the software is, the more important the managerial skills become to fully capitalize the software investment in order to get the desired outcomes from the software investment. General software can be used for more tasks than specific software which has designated tasks for which it is developed for. An investment in general software, i.e. a general purpose technology type of software, would therefore also result in more intangible consequences than the specific software. It would therefore require more managerial skills for the economic return from the software to be maximized. The level of intangible consequences from a software investment is therefore positively correlated with the level of efficiency of the software platform from which it is derived.

The managerial skills needed to maximize the cash flows from the software can then be argued to be the knowledge of what type of effort is needed to exercise the options for future cash flows. This is depending on the type of option that the organization has. In general terms these can be of two types; growth call-type options or protective put-type options (Cheung and Bagranoff, 1991). In the case of growth-call type options the firm expects to increase the cash flow by capitalizing on the intangible consequences, i.e. the intangible benefits, when exercising the option. The exercise price may then be reorganization of staff as discussed above or customizing the software as discussed in Chapter 6. Another type of investments that can be seen as an exercise price is the cost for increased training of the staff in using the adopted software. If the firm wants to receive less of the intangible consequences, i.e. the intangible costs, the firm may use a protective put-type option and scale down the usage of the software as discussed in Chapter 5.

SUGGESTIONS FOR FURTHER RESEARCH

This dissertation has partly aimed to show that the use of economic models to the field of software development and adoption may provide better basis for decision making. Although research in the field of software economics has been growing there is still a need for more research in the area. More research in exploring the use of more sophisticated models and methods from economic theory may further improve the decision-making in the field of software economics. Based on the conclusions from this study and the state of previous research the below suggestions for further research are motivated.

The concept of platforms has been used in a wider sense than in previous research in software economics in this study. There is still a need for more empirical and theoretical research on software platforms in a wider sense in software economics. From a managerial point of view, we would like to know how platforms can be used to manage uncertainty in different settings. It is also relevant to study what types of opportunities, i.e. real options, for future software product development are created by different types of software platforms. To decrease uncertainty and to create the most options for the future it we need to know how the relation between platforms and options look in different settings.

Real options modeling have been applied for some time now in the software economics field. Most of this research has been devoted to applying option valuation methods given an option. This is a well-explored branch in financial economics. Less research has been devoted to the question of how options are created in software investments. There is need for more research on a managerial level but also on a more operational level on this matter. More research is also needed on how managerial skills can be used and what type of managerial skills is needed to create the flexibility, to manage the outcomes of software investments and to fully capitalize the possible economic return from the investment.

Finally, there is a need for more research on the realization of the economic characteristics of a software investment during the development stage as well as after the adoption of the software. By knowing this, the right type of underlying process for the cash flow function can be derived and thereby make the economic modeling more accurate and useful. The realization may not look the same for all types of software and therefore it needs to be studied for different types of software.

REFERENCES

Aija, S. A., W, D. (2007) "Empirical Study of the Effects of Open Source Adoption on Software Development Economics", *The Journal of Systems and Software*, Vol. 80, No. 9, pp. 1517-1529

Aggoun, L., Elliot, R. (2004) "Measure Theory and Filtering - Introduction with Applications", *Cambridge University Press*, New York

Alchian, A. A., Demetz, H. (1972) "Production, Information Costs and Economic Organization", *American Economic Review,* Vol. 62, No. 5, pp. 777-795

Anadarajan, A., Wen, H. J. (1999) "Evaluation of Information Technology Investment" *Management Decision*, Vol. 37, No. 4, pp. 329-337

Aral, S., Brynjolfsson, E., Wu, D. J. (2006) "Which Came First, IT or Productivity: Virtuous Cycle of Investment and Use in Enterprise Systems", *Twenty Seventh International Conference on Information Systems, Milwaukee, 2006*, available at:
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=942291
2009-07-28

Ashford, R W., Dyson, R. G., Hodges, S D. (1988) "The Capital-Investment Appraisal of New Technology: Problems, Misconceptions and Research Directions", The Journal of Operational Research Society, Vol. 39, No. 7, pp. 637-642

Asundi, J., Kazman, R. (2001) "A Foundation of the Economic Analysis of Software Architectures", *Proceedings of the 3rd Workshop on Economic-Driven Software Engineering*, pp. X-XX

Baccarini, D., Salm, G., Love, P. E. D. (2004) "Management of Risks in Information Technology Projects", *Industrial Management and Data Systems*, Vol. 104, No. 4, pp. 286-295

Baddeley, M. C. (2003) "Investment – Theories and Analysis", *Palgrave Macmillan*, New York

Banker, R. D., Kauffman, R. J. (2004) "The Evolution of Research on Information Systems: A Fiftieth-Year Survey of the Literature", *Management Science*, Vol. 50, No. 3, pp. 281-298

Banker, R.D., Kauffman, R.J. and Mahmood, M.A. (1993), "Information Technology, Strategy and Firm Performance Evaluation: New Perspectives for Senior Management From Information systems research", in Banker, R. D., Kauffman, R. J. and Mahmood, M. A. (Eds), Strategic Information Technology Management: Perspectives on Organizational Growth and Competitive Advantage, *Idea Group Publishing*, Harrisburg

Barki, H., Rivard, S., Talbot, J. (1993) "Towards an Assessment of Software Risk", *Journal of Management Information Systems*, Vol. 10, No. 2, pp. 203-225

Beinhocker, E. D. (1999) "Robust Adaptive Strategies", *Sloan Management Review*, Vol. 40, No. 3, pp 95-106

Bellatine, J., Stray, S. (1998) "Financial Appraisal and the IS/IT Investment Decision Making Process", *The Journal of Information Technology*, Vol. 13, No. 1, pp. 3-14

Bellatine, J., Stray, S. (1999) "Information System and Other Capital Investments: Evaluation Practice Compared", *Logistics Information Management*, Vol. 12 No. 1-2, pp. 78-93

Benaroch, M., Jeffery, M., Kauffman, R. J., Shah, S. (2007) "Option-Based Risk Management: A Field Study of Sequential Information Technology Investment Decisions", *Journal of Management Information Systems*, Vol. 24, No. 2, pp. 103-140

Benaroch, M., Lichtenstein, Y., Robinson, K. (2006) "Real Options in Information Technology Risk Management: An Empirical Validation of Risk-Option Relationship", *MIS Quarterly*, Vol. 30, No. 4, pp. 827-854

Berinato, S. (2001) "Using Portfolio Management to Demonstrate IT Value", *CIO Magazine*, Oktober 01, pp. 1-4

Berk, J., DeMarzo, P. (2008) "Corporate Finance", *Prentice Hall*, New York

Birk, A., Heller, G., John, I., Schmid, K., von der Massen, T., Muller, K. (2003) "Product Line Engineering: The State of the Practice", *IEEE Software*, Vol. 20, No. 6, pp. 52-60

Boehm, B. "Value-Based Software Engineering", *Software Engineering Notes*, Vol. 28, No. 2, pp. 1-12

Bolton, P., Dewatripont, M. (2005) "Contract Theory", *The MIT Press*, New York

Borenstein, D., Betencourt, P. R. B. (2005) "A Multi-Criteria Model for IT Investments", *INFOR*, Vol. 43, No. 1, pp. 1-21

Borison, A. (2005) "Real Option Analysis: Where are the Emperor's new Cloths?", *Journal of Applied Corporate Finance*, Vol. 17, No. 2, pp. 17-31

Brealey, R., Myers, S. (2003) "Corporate Finance", *McGraw-Hill*, New York

Brynjolfsson, E. (2003) "The IT Productivity Gap", *Optimize*, July

Brynjolfsson, E., Hitt, L. M. (2003) "Computing Productivity: Firm-Level Evidence", *Review of Economics and Statistics*, Vol. 85, No. 4, pp. 793-808

Butler, S., Chalasani, P., Jha, S., Raz, O., Shaw, M. (1999) " The Potential of Portfolio Theory in Guiding Software Decisions", *Proceedings of the 1st Workshop on Economic-Driven Software Development,*

Carlsson, B. (Ed.) (1995) Technological Systems and Economic Performance: the Case of Factory Automation, *Kluwer Academic Publishers*, Boston.

Carlsson B. (Ed.) (1997) Technological Systems and Industrial Dynamics, *Kluwer Academic Publishers*, Boston.

Carlsson, B. (Ed.) (2002) New Technological Systems in the Bio Industries – an International Study, *Kluwer Academic Publishers*, Boston.

Carlsson, B. (2004) The Digital Economy – What is New and What is Not?, *Structural Change and Economic Dynamics*, Vol. 15, No. 3, pp. 245-264.

Carlsson, B. and Stankiewicz, R. (1991) On the Nature, Function, and Composition of Technological Systems, *Journal of Evolutionary Economics*, Vol. 1, No. 2, pp. 93-118.

Chan, Y. E. (2000) "IT Value: The Great Divide Between Qualitative and Quantitative and Individual and Organizational Measures", *Journal of Management Information Systems*, Vol. 16, No. 4, pp. 225-261

Cheung, J. K., Bagranoff, N. A. (1993) "Assessing Intangible Costs and Benefits in Systems Decisions via Option Pricing Theory", *Journal of Information Systems*, Vol. 5, No. 1, pp. 36-47

Chou, T-Y., Chou, S-C. T., Tzeng, G-H. (2006) "Evaluating IT/IS Investments: A Fuzzy Multi-Criteria Decision Model Approach", *European Journal of Operational Research,* Vol. 173, No. 3, pp. 1026-1046

Clements, P., Northrop, L. (2002) "Software Product Lines: Practice and Patterns", *Addison-Wesley,* New York

Clemmons, E. (1991) "Evaluation of Strategic Investments in Information Technology", *Communication of the ACM*, Vol. 34, No. 1, pp. 22-36

Cools, K., van Praag, M. (2003) "The Value Relevance of Disclosing a Single Corporate Target: An Explorative Empirical Analysis", *Tindberg Institute Discussion Paper*, available at:
http://papers.ssrn.com/paper=244788
2007-04-09

Copeland, T., Antikarov, V. (2003) "Real Options – A Practitioner's Guide", *Texere*, New York

Copeland, T. E., Weston, F. J., Shastri, K. (2005) "Financial Theory and Corporate Policy", *Addison-Wesley*, Boston

Davidson, E., Heslinga, D. (2007) "Bridging the IT Adoption Gap for Small Physician Practices: An Action Research Study on Electronic Health Records", *Information Systems Management*, Vol. 24, No.1, pp. 15-28

Davis, F. D. (1989) "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology", *MIS Quarterly*, Vol. 13, No. 3, pp. 319-340

Dedrick, J., Gurbaxani, V., Kraemer, K. L. (2003) "Information Technology and Economic Performance: A Critical Review of the Empirical Evidence", *ACM Computing Surveys*, Vol. 35, No. 1, pp 1-28

Dehning, B., Richardson, V. J., Zmud, R. W. (2007) "The Financial Performance Effect of IT-Based Supply Chain Management Systems in Manufacturing Firms", *Journal of Operations Management,* Vol. 25, No. 4, pp. 806-824

De Matos, J. A. (2001) "Theoretical Foundations of Corporate Finance", *Princeton University Press*, Princeton, New Jersey

Devaraj, S., Kohli, R. (2003) "Performance Impacts of Information Technology: Is Actual Usage the Missing Link?", *Management Science*, Vol. 49, No. 3, pp. 273-289

Dey, P. K., Kinch, J., Ogunlana, S. O. (2007) "Managing Risk in Software Projects: A Case Study", *Industrial Management and Data Systems*, Vol. 107, No. 2, pp. 284-303

Dixit, A., Pindyck, R. S. (1994) ”Investment under Uncertainty”, *Princeton University Press*, Princeton

Dixit, A., Pindyck, R. S. (1995a) ”The Option Approach to Capital Investment”, Harvard Business Review, May/June, pp. 105-115

Dixit, A., Pindyck, R. S. (1995b) ”The Option Approach to Capital Investment” *Long Range Planning*, Vol. 28, No.4, pp. 129-129

Dos Santos, B. L. (1991) "Justifying Investments in new Technologies", *Journal of Management Information Systems*, Vol. 7, No.4, pp. 71-90

Edvardsson, B., Gustafsson, A., Johnsson, M. D. and Sandén, B. (2000) "New Service Development and Innovation in the New Economy" *Studentlitteratur*, Lund.

Eisenhardt, K. M. (1989) "Agency Theory: An Assessment and Review", *The Academy of Management Review*, Vol. 14, No. 1, pp. 57-74

Eliasson, G. and Eliasson, Å. (1996) The Biotechnical Competence Bloc, *Revue d'Economie Industrielle*, Vol. 78, No. 4, pp. 7-26.

Elton, E. J., Gruber, M. J., Brown, S. J (2006) "Modern Portfolio Theory and Investment Analysis", *John Wiley & Sons*, Hoboken

Erdogmus, H. (2002) “Valuation of Learning Options in Software Development Under Private and Market Risk”, *Engineering Economist*, Vol. 47, No. 3. pp. 308-353

Erdomus, H. ,Favoro, J., Halling, M. (2006) ”Valuation of Software Incentives Under Uncertainty: Concepts, Issues and Techniques”, in Biffl, S., Aurum, Ü., Boehm, B., Erdogmus, H., Grnbacher, P (2006) “Value-Based Software Engineering”, *Springer Verlag*, Berlin Heidelberg

Estrin, S., Laidler, D., Dietrich, M. (2008) “Microeconomics”, *Prentice Hall,* Harlow

Falk, T., Olve, N-G. (1996) “IT som Strategisk Resurs” (IT as a Strategic Resource, authors own translation), *Liber-Hermods*, Malmö

Falmagne, J-C. (2003) “Lectures in Elementary Probability Theory and Stochastic Processes”, *McGraw-Hill*, New York

Fichman, R. G. (2004) “Real Options and IT Platform Adoption: Implications for Theory and Practice”, *Information Systems Research*, Vol. 15, No. 2, pp. 132-154

Fichman, R. G., Kemerer, C.F. (1999) ”The Illusory Diffusion of Innovation: An Examination of Assimilation Gaps”, *Information Systems Research*, Vol. 10, No. 3, pp. 255-275

Fisher, I. (1930) “The Theory of Interest: As Determined by Impatience to Spend Income and Opportunity to Invest it”, *Macmillan*, New York

Frederick, S., Loewenstein, G., O'Donoghue, T. (2002) "Time Discounting and Time Preferences: A Critical Review", *Journal of Economic Literature*, Vol. 40, No. 2, pp. 351-401

Freeman, R. E. (1984) "Strategic Management: A Stakeholder Approach", *Pitman*, Boston

Glass, R. L. (2004) "A Look at the Economics of Open Source – Is Open Software the Future of the Software Field of a Passing Fad?", *Communications of the ATM*, Vol. 47, No. 2, pp. 25-27

Goldstein, S. M., Johnson, R., Duffy, J. and Rao, J. (2002) "The service concept: the missing link in service design research?" *Journal of Operations Management*, Vol. 20, No. 2, pp 121-134

Gottschalk, P., Karlsen, J. (2005) "A Comparison of Leadership Roles in Internal IT Projects Versus Outsourcing Projects", *Industrial Management and Data Sourcing*, Vol. 105, No. 9, pp. 1137-50

Graham, J. R., Harvey, C. R. (2001) "The Theory and Practice of Corporate Finance: Evidence From the Field", *Journal of Financial Economics*, Vol. 60, No. 2-3, pp. 187-244

Gunasekaran, A., Love, P. E. D., Rahimi, F., Miele, R. (2001) "A Model for Investment Justification in Information Technology Projects", *International Journal of Information Management*, Vol. 21, No. 5, pp. 349-364

Gunasekaran, A, Ngai, E. W. T., McGaughey, R. E. (2006) "Information Technology and Systems Justification: A Review for Research and Applications", *European Journal of Operational Research*, Vol. 173, No. 3, pp. 957-983

Hallikainen, P, Kivijärvi, H, Nurmimäki, K. (2002) "Evaluating Strategic IT Investment: An Assessment of Investment Alternatives for a Web Management System", *Proceedings of the 35th Hawaii International Conference on System Science*, pp. X-XX

Hirshleifer, J. (1958) "On the Theory of Optimal Investment Decision", *Journal of Political Economy*, Vol. 66, No. 4, pp. 329-352

Hochstrasser, B., Griffiths, C. (1991) "Controlling IT Investment: Strategy and Management", *Chapman & Hall*, New York

Holmstrom, B. (1979) "Moral Hazard and Observability", *Bell Journal of Economics*, Vol. 10, No. 1, pp. 74-91

Im, K. S., Dow, K. E., Grover, V. (2001) "Research Report: A Reexamination of IT Investment and the Market Value of Firm – an Event Study Methodology", *Information Systems Research*, Vol. 12, No. 1, pp. 103-117

Irani, Z. (2008) "Information Systems Evaluation: What Does it Mean?", *Construction Innovation*, Vol. 8, No. 2, pp. 88-91

Irani, Z. (2002) "Information Systems Evaluation: Navigating Through the Problem Domain", *Information & Management*, Vol. 40, No. 1, pp. 11-24

Irani, Z., Sharif, A. M., Love, P. E. D. (2007) "Knowledge Mapping and Information Systems Evaluation in Manufacturing", *International Journal of Production Research*, Vol. 45, No. 11, pp. 2435-2457

Irani, Z., Love, P. E. D. (2002) “Developing a Frame of Reference for *ex-ante* IT/IS Investment Evaluation”, *European Journal of Information Systems*, Vol. 11, No. 1, pp. 75-82

Irani, Z., Love, P. E. D. (2000) “The Propagation of Technology Management Taxonomies for Evaluating Investments in Information Systems”, *Journal of Management Information Systems*, Vol. 17, No. 3, pp. 161-177

Jeffery, M., Lelivald, I. (2004) ”Best Practices in IT Portfolio Management”, *MIT Sloan Management Review*, Vol. 45, No. 3, pp. 41-49

Jensen, M. C. (2001a) “Value Maximization, Stakeholder Theory, and the Corporate Objective Function”, *European Financial Management*, Vol. 7, No. 3, pp. 297-317

Jensen, M. C. (2001b) “Value Maximization, Stakeholder Theory, and the Corporate Objective Function” in Chew, Jr., D. H., Gillian, S. L. (2005) “Corporate Governance at the Crossroads: A book of Readings”, *McGraw-Hill*, New York

Jensen, M. C., Meckling, W. H. (1976) ”Theory of the Firm: Managerial Behavior, Agency Costs and Ownership Structure”, *Journal of Financial Economics*, Vol. 3, No.4, pp. 305-360

Johnson, J. P. (2001) “Economics of Open Source Software”
Available at:
http://opensource.mit.edu/papers/johnsonopensource.pdf
2009-02-18

Kambil, A., Henderson, J. C., Mohsensadeh, H. (1993) ”Strategic Management of Information Technology: an Options Perspective”, *CISR WP No. 222 – Sloan WP No. 3319*, available at:
http://dspace.mit.edu/bitstream/handle/1721.1/49166/strategicmanagem00kamb.pdf?sequence=1
2009-11-03

Katsamakas, E., Xin, M. (2005) “An Economic Analysis of Enterprise Adoption of Open Source Software”, *NET Institute Working Paper Series 05-29*
Available at:
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=851306
2009-02-18

Kauffman, R. J., Wu, P. (2006) ”Optimal Timing for Software Functionality Additions by Internet Portals”, Proceedings of the 39th Hawaii International Conference on Systems Science, pp. X-XX

Kazman, R., Asundi, J., Klein, M. (2001) “Quantifying the Costs and Benefits of Architectural Decisions”, *Proceedings of the 23rd International Conference on Software Engineering,* pp. 297-306

Kester, W. C. (1984) “Today’s Options for Tomorrow’s Growth”, *Harvard Business Review*, Vol. 62, No. 2, pp. 153-160

Kim, G. T., Kim, Y. B. (2005) “An Expository Note on an Economic Justification of Investments in Advanced Manufacturing Systems”, *Journal of Advanced Manufacturing Systems*, Vol. 26, No. 7-8, pp. 934-941

Knight, F. (1921) “Risk, Uncertainty and Economic Profit”, *Houghton Mifflin Co.*, Boston

Kreuger, C. W. (2000) "New Methods in Software Product Line Practice – Examining the Benefits of Next-Generation SPL Methods", *Communication of ACM*, Vol. 49, No. 2, pp. 37-40

Krishnan, M. S., Mukhopadhyay, T., Kriebel, C. H. (2004) "A Decision Model for Software Maintenance", *Information Systems Research*, Vol. 15, No. 4, pp. 396-412

Kumar, R. L. (1995) "An Option View of Investments in Expansion-Flexible Manufacturing Systems", *International Journal of Production Economics*, Vol. 38, No.2-3 , pp. 281-291

Laffont, J-J., Marimort, D. (2002) "The Theory of Incentives – The Principal-Agent Model", *Princeton University Press*, Princeton, New Jersey

Lancaster, K. (1963) "An Axiomatic Theory of Consumer Time Preferences", *International Economic Review*, Vol. 4, No. 2, pp. 221-231

Lee, S., Kim, S. H. (2006) "A Lag Effect of IT Investment on Firm Performance", *Information Resource Management Journal*, Vol. 19, No. 1, pp. 43-69

Lefley, F. (2004) "An Assessment of Various Approaches for Evaluating Project Strategic Benefits- Recommending the Strategic Index", *Management Decision,* Vol. 42, No. 7 pp. 850-862

Lefley, F. (1996) "Strategic Methodologies of Investment Appraisal of AMT Projects: A review and Synthesis", *The Engineering Economist*, Vol. 41, No. 4, pp. 345-363

Lefley, F. (1994) "Capital Investment Appraisal of Manufacturing Technology", *International Journal of Production Research*, Vol. 32, No. 12, pp. 2751-2776

Lerner & Tirole (2000) "The Simple Economics of Open Source", *HBS Finance Working Paper*, No. 00-059
Available at:
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=224008
2009-02-18

Lerner & Tirole (2001) "The open source movement: Key research questions", *European Economic Review*, Vol. 45, No. 4, pp. 819-826

Lesjak, D. and Vehovar, V. (2005), "Factors Affecting Evaluation of E-Business Projects", *Industrial Management & Data Systems*, Vol. 105, No. 4, pp. 409-28

Li, Y., James, B. E., Madhavan, R., Mahoney, J. T. (2007) "Real Options: Taking Stock and Looking Ahead", *Advances in Strategic Management*, Vol. 24, pp. 31-66

Lin, B.-W. (2007) "Information Technology Capability and Value Creation: Evidence From the US Banking Industry", *Technology in Society*, Vol. 29, No. 1, pp. 93-106

Lintner, J. (1965) "Security Prices, Risk, and Maximal Gains from Diversification", *Journal of Finance*, Vol. 20, No. 4, pp. 587-615

Liu, Q., Mintram, R. C. (2005) "Preliminary Data Analysis Methods in software Estimation", *Software Quality Journal*, Vol. 13, No. 1, pp. 91-115

Love, P. E., Irani, Z., Standing, C., Lin, C., Burn, J. M. (2005) "The Enigma of Evaluation: Benefits, Costs and Risks of IT in Australian Small-Medium-Sized Enterprises", *Information and Management*, Vol. 42, No. 7, pp. 947-964

Lubbe, S., Remenyi, D. (1999) "Management of Information Technology Evaluation – The Development of a Managerial Thesis", *Logistics Information Management*, Vol. 12, No. 1/2, pp. 145-156

Luce, R. D., Raiffa, H. (1957) "Games and Decisions: An Introduction and Critical Survey", John Wiley and Sons, New York

McDonald, R. L. (2006) "The Role of Real Options in Capital Budgeting: Theory and Practice", *Journal of Applied Corporate Finance,* Vol. 18, No. 2, pp. 28-39

Mas-Colell, A., Whinston, M. D., Green, J. R. (1995) "Microeconomic Theory", *Oxford University Press*, New York

Massé, P. (1962) "Optimal Investment Decisions", *Englewood Cliffs*, New York

Milis, K., Mercken, R. (2004) "The use of Balanced Scorecard for the Evaluation of Information and Communication Technology Projects", *International Journal of Project Management*, Vol. 22, No. 2, pp. 87-97

Mitra, S. (2005) "Information Technology as an Enabler of Growth in Firms: An Empirical Assessment", *Journal of Management Information Systems*, Vol. 22, No. 2, pp. 279-300

Modigliani, F., Miller, M. H. (1958) "The Cost of Capital, Corporate Finance and Theory Investment", *American Economic Review*, Vol. 48, No. 3. pp. 261-297

Mossin, J. (1966) "Equilibrium in a Capital Asset Market", *Econometrica*, Vol. 34, No. 4, pp. 768-783

Mukherji, N., Rajagopalan, B., Tanniru, M. (2006) "A Decision Support Model for Optimal Timing in Investments in Information Technology Upgrades", *Decision Support Systems*, Vol. 42, No. 3, pp. 1684-1696

Murphy, K. E., Simon, S. J. (2002) "Intangible Benefit Valuation in ERP Projects", *Information Systems Journal*, Vol. 12, No. 4, pp. 301-320

Mustonen (2001) "Copyleft - The Economics of Linux and Other Open Source Software", *Information Economics and Policy*, Vol. 15, No. 1, pp. 99-121

Myers, S. C. (1977) "Determinants of Corporate Borrowing", *Journal of Financial Economics,* Vol. 5, No. 2, pp. 147-176

Myers, S. C. (1984) "Finance Theory and Financial Strategy", *Interfaces*, Vol. 14, No. 1, pp. 126-137

Neftci, S. N. (2000) "An Introduction to the Mathematics of Financial Derivatives", *Academic Press*, New York

Numminen, E. (2008a) "Software Investment Under Uncertainty: Modeling Intangible Consequences as a Stochastic Process", *Blekinge Institute of Technology Licentiate Dissertation Series 2008:7*, available at:

http://www.bth.se/fou/forskinfo.nsf/3efcc0263496fe93c125681f0031bfd0/c4fc1a96b53c2937c125746500360fec!OpenDocument
2009-07-28

Numminen, E. (2008b) "Valuation of a Software Investment When you Have the Option to Customize it – The Case of Open Source Software", presented at the 18th Triennial Conference of the International Federation of Operational Research Societies in Johannesburg, South Africa, unpublished

Numminen, E (2008c) "Modeling the Return of Information System Investments", *Proceedings of the 2nd European Conference on Information Management and Evaluation*, pp. 325-334

Numminen, E. (2010a) "Hedging Uncertainty in Software Development – A Real Option Modeling Approach", Presented at the 24th European Conference for Operations Research in Lisbon, Portugal, unpublished

Numminen, E. (2010b) "Why Software Platforms Make Sense in Risk Reduction in Software Development – A Portfolio Theory Approach", *Proceedings of the 4th European Conference on Information Management and Evaluation*, pp. 282-290

Numminen, E., Wrenne, A (2009) "Uncertainty Reduction in Software Development by the use of a Platform Based Development Strategy", *Proceedings of the 3rd European Conference on Information Management and Evaluation,* pp. 356-362

Numminen, E., Wrenne, A. (2010) "The Role of Platforms in Software Development – Planting Real Options to Manage Uncertainties", *Proceed-*

ings of the 1st International Conference on Information Management and Evaluation, pp. 289-295

Orlikowski, W. J. (2008) "The Sociomateriality of Organizational Life: Considering Technology in Management Research", Cambridge Journal of Economics, Vol. 34, No. 1, pp. 124-141

Orlikowski, W. J. (2007) "Sociomaterial Practices: Exploring Technology at Work", *Organization Studies*, Vol. 28, No. 9, pp. 1435–1448

Orlikowski, W. J. (1992) "The Duality of Technology: Rethinking the Concept of Technology in Organizations", *Organization Science*, Vol. 3, No. 3, pp. 398–427

Orlikowski, W. & Scott, S. (2008) "Sociomateriality: Challenging the Separation of Technology", Work and Organization, *The Academy of Management Annals*, Vol. 2, No. 1, pp. 433–474

Papoulis, A., Pillai, S. U. (2002) "Probability, Random Variables and Stochastic Processes", *McGraw-Hill*, New York

Peteraf. M. A. (1993) "The Cornerstones of Competitive Advantage: A Resource-Based View, *Strategic Management Journal*, Vol. 14, No. 3, pp. 179-192

Pohl, K., Böcke, G., van der Linden, F. (2005) "Software Product Line Engineering – Foundations, Principles, and Techniques", *Springer Verlag*, Berlin

Powell, P. (1992) "Information Technology Evaluation: is it Different?", *The Journal of Operational Research Society*, Vol. 43, No. 1, pp- 29-42

Rapp, B. (1974) "Models for Optimal Investment and Maintenance Decisions", *Almqvist & Wiksell International*, Stockholm

Rapp, B., Thorstenson, A. (1994) "Vem tar Risken" (Who's Taking the Risk, author's own translation), *Studentlitteratur*, Lund

Raymond, E. S. (2000) "The Cathedral and the Bazaar", *Thyrsus Enterprises*
Available at:
http://www.catb.org/~esr/writings/cathedral-bazaar/
2009-02-18

Renkama, T. J. W., Berghout, E. W. (1996) "Methodologies for Information Systems Investments at a Proposal Stage: A Comparative Review", *Information and Software Technology*, Vol. 39, No. 1, pp. 1-13

Rehesaar, H., Mead, A. (2005) "An Extension of Benefit Cost Analysis to IS/IT Investments", *The Business Review*, Vol. 4, No. 1, pp. 89-93

Renkema, Th. J. W and Berghout, E.W. (1996) "Methodologies for Information Systems Evaluation at the Proposal Stage: A Comparative Study", *Information and Software Technology*, Vol. 39, No.1, pp. 1-13

Reuer, J. J., Tong, T. W. (2007) "How do Real Options Matter? Empirical Research on Strategic Investments and Firm Performance", *Advances in Strategic Management,* Vol. 24, No. 145-173

Roberts, J., Hann, I-H., Slaughter, S. (2006) "Understanding the Motivation, Participation and Performance of Open Source Software Develop-

ers: A Longitudinal Study of the Apache Projects", *Marshall School of Business Working Paper Series No. IOM 01-01*
Available at:
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=918518
2009-02-18

Rose, D. (1999) "Teams, Firms, and Evolution of Profit Seeking Behavior", *Working Paper Department of Economics at the University of Missouri-St. Luis*
Available at:
http://papers.ssrn.com/paper=224438
2007-04-09

Ross, S. A. (1976) "The Arbitrage Theory of Capital Asset Pricing", *Journal of Economic Theory*, Vol. 13, No. 3, pp. 341-360

Ross, S. A., Westerfield, R., Jaffey, J. (2007) "Corporate Finance", *McGraw-Hill*, New York

Rubinstein, M. (2003) "Great Moments in Financial Economics I. Present Value"
Available at:
http://leeds.colorado.edu/uploadedFiles/_Documents/Centers_of_Excellence/Burridge_Center/RubinsteinHistoryofPresentValue.pdf
2010-11-10

Rubinstein, M. (2003) "Great Moments in Financial Economics: II. Modigliani-Miller Theorem", *Journal of Investment Management,* Vol. 1, No. 2, pp. 7-13

Ryan, P. A., Ryan, G. P. (2002) "Capital Budgeting Practices of the Fortune 1000: How Have Things Changed?", *Journal of Business and Management*, Vol. 8, No. 4, pp. 355-364

Schwalbe, K. (2002) "Information Technology Management", *Thomson Learning*, Scarborough

Schwartz, E. S., Trigeorgis, L. (Eds.) (2001) "Real Options and Investment Under Uncertainty: Classical Reading and Recent Contributions", *The MIT Press*, Cambridge

Schwartz, E. S., Zozaya-Gorostiza, C. (2003) "Investment Under Uncertainty in Information Technology: Acquisition and Development Projects", *Management Science*, Vol. 49, No. 1, pp. 57-70

Serafemidis, V., Smithson, S. (2000) "Information System Evaluation in Practice: A Case Study of Organizational Change", *Journal of Information Technology*, Vol. 15, No. 2, pp. 93-106

Shapiro, C., Varian, H. L. (1999) "Information Rules – A Strategic Guide to the Network Economy", *Harvard Business Press*, Boston

Sharpe, W. F. (1964) "Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk", *Journal of Finance*, Vol. 19, No. 3. pp. 425-442

Shirky, C. (2008) "Here Comes Everybody: The Power of Organizing Without Organizations", *Penguin Press HC*, New York

Shreve, S. E. (2004a) "Stochastic Calculus for Finance I – The Binomial Asset Pricing Model", *Springer Verlag*, New York

Shreve, S. E. (2004b) "Stochastic Calculus for Finance II – Continuous-Time Models", *Springer Verlag*, New York

Shuurman, P. M., Berghout, E. W., Powell P. (2009) "Benefits are From Venus, Costs are From Mars", *Sprouts: Working Papers on Information Systems*, Vol. 9, No.3, pp. 1-16,
Available at:
http://sprouts.aisnet.org/9-3
2009-07-28

Sivzattian, S., Nuseibeh, B. (2001) "Linking the Selection of Requirements to Market Value: A Portfolio-Based Approach", *Proceedings of the 7th International Requirements Engineering Conference*, pp. 1-8

Small, M. H., Chen, J. (1995) "Investment Justification of Advanced Manufacturing Technology: An Empirical Analysis", *Journal of Engineering and Technology Management*, Vol. 12, No. 1-2, pp. 27-55

Smithson, S., Hirschheim, R. (1998) "Analysing Information System Evaluation: Another Look at an old Problem", *European Journal of Information Systems*, Vol. 7, No. 3, pp. 158-174

Standing, C., Guilfoyle, A., Lin, C., Love, P. E. D. (2006) "The Attribution of Success and Failure of IT Projects", *Industrial Management and Systems Data*, Vol. 106, No. 8, pp. 1148-1165

Steffens, P. R., Douglas, E. J. (2007) "Valuing Technology Investments: Use Real Option Thinking but Forget Real Option Valuation", *International Journal of Technoentrepreneurship*, Vol. 1, No.1 pp. 58-77

Stratopoulos, T., Dehning, B. (2000) "Does Successful Investment in Information Technology Solve the Productivity Paradox?", *Information and Management*, Vol. 38, No. 2, pp. 103-117

Szajna, B. (1996) "Empirical Evaluation of the Revised Technology Acceptance Model", *Management Science*, Vol. 42, No. 1, pp. 85-92

Taudes, A., Feurstein, M., Mild, A. (2000) "Option Analysis of Software Platform Decisions: A Case Study", *MIS Quarterly*, Vol. 24, No. 2, pp. 227-243

Taylor, S., Todd, P. A. (1995) "Understanding Information Technology Usage: A Test of Competing Models", *Information System Research*, Vol. 6, No. 2, pp. 144-176

Thorp, J. (2003) "The Information Paradox: Realizing the Business Benefits of Information Technology", *McGraw-Hill*, New York

Trigeorgis, L. (1996) "Real Options – Managerial Flexibility and Strategy in Resource Allocation", *The MIT Press*, Cambridge

Tüysüz, F. and Kahraman, C. (2006) "Project Risk Evaluation Using a Fuzzy Analytical Hierarchy Process: An Application to Information Technology Projects", *International Journal of Intelligent Systems*, Vol. 21, No. 6, pp. 559-584

van der Hejden, H. (2004) "User Acceptance of Hedonic Information Systems", *MIS Quarterly*, Vol. 28, No. 4, pp. 695-704

Venkatesh, V, Davis, F. D. (2000) "A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies", *Management Science*, Vol. 46, No. 2, pp. 186-204

Venkatesh, V., Morris, M. G., Davis, G. B., Davis, F.D. (2003) "User Acceptance of Information Technology: Toward a Unified View", *MIS Quarterly*, Vol. 27, No. 3, pp. 425-478

Verhoef, C. (2002) "Quantitative IT Portfolio Management", *Science of Computer Programming*, Vol. 45, No. , pp. 1-96

von Eisenhardt, S. (2008) "The Economic Properties of Software", *Jena Research Papers 2008 – 045*, available at:
http://zs.thulb.uni-jena.de/receive/jportal_jparticle_00101440
2010-04-23

Wilcocks, L., Lester, S. (1994) "Evaluating the Feasibility of Information Technology Investments: Recent UK Evidence and New Approaches" in: Wilcocks, L. (Ed.) (1994) "Information Management: The Evaluation of Information Technology Investments", *Chapman & Hall*, New York

Zhang, X., Windsor, J. (2003) "An Empirical Analysis of Software Volatility and Related Factors", *Industrial Management and Data Systems*, Vol. 103, No. 3-4, pp. 275-281

APPENDIX 1
MIT INFORMATION

The Swedish Research School of Management and Information Technology

The *Swedish Research School of Management and Information Technology* (MIT) is one of 16 national research schools supported by the Swedish Government. MIT is jointly operated by the following institutions: Blekinge Institute of Technology, Gotland University College, Jönköping International Business School, Karlstad University, Linköping University, Linnaeus University Växjö, Lund University, Mälardalen University College, Stockholm University, Örebro University, IT University of Göteborg, and Uppsala University, host to the research school. At the Swedish Research School of Management and Information Technology (MIT), research is conducted, and doctoral education provided, in three fields: management information systems, business administration, and informatics.

DISSERTATIONS FROM THE SWEDISH RESEARCH SCHOOL OF MANAGEMENT AND INFORMATION TECHNOLOGY

Doctoral theses (2003-)

1. **Baraldi, Enrico (2003),** *When Information Technology Faces Resource Interaction: Using IT Tools to Handle Products at IKEA and Edsbyn.* Department of Business Studies, Uppsala University, Doctoral Thesis No. 105.
2. **Wang, Zhiping (2004),** *Capacity-Constrained Production-Inventory Systems: Modelling and Analysis in both a Traditional and an E-Business Context*. IDA-EIS, Linköpings universitet och Tekniska Högskolan i Linköping, Dissertation No. 889
3. **Ekman, Peter (2006),** *Enterprise Systems & Business Relationships: The Utilization of IT in the Business with Customers and Suppliers*. School of Business, Mälardalen University, Doctoral Dissertation No 29.
4. **Lindh, Cecilia (2006),** *Business Relationships and Integration of Information Technology*. School of Business, Mälardalen University, Doctoral Dissertation No 28.
5. **Frimanson, Lars (2006),** *Management Accounting and Business Relationships from a Supplier Perspective.* Department of Business Studies, Uppsala University, Doctoral Thesis No. 119.
6. **Johansson, Niklas (2007),** *Self-Service Recovery*. Information Systems, Faculty of Economic Sciences, Communication and IT, Karlstad University, Dissertation KUS 2006:68.
7. **Sonesson, Olle (2007),** *Tjänsteutveckling med personal medverkan: En studie av banktjänster.* Företagsekonomi, Fakulteten för ekonomi, kommunikation och IT, Karlstads universitet, Doktorsavhandling, Karlstad University Studies, 2007:9.
8. **Maaninen-Olsson, Eva (2007),** *Projekt i tid och rum: Kunskapsintegrering mellan projektet och dess historiska och organisatoriska kontext.* Företagsekonomiska institutionen, Uppsala universitet, Doctoral Thesis No. 126.
9. **Keller, Christina (2007),** *Virtual learning environments in higher education: A study of user acceptance.* Linköping Studies in Science and Technology, Dissertation No. 1114.

10. **Abelli, Björn (2007),** *On Stage! Playwriting, Directing and Enacting the Informing Processes*. School of Business, Mälardalen University, Doctoral Dissertation No. 46.

11. **Cöster, Mathias (2007),** *The Digital Transformation of the Swedish Graphic Industry*. Linköping Studies in Science and Technology, Linköping University, Dissertation No. 1126.

12. **Dahlin, Peter (2007),** *Turbulence in Business Networks: A Longitudinal Study of Mergers, Acquisitions and Bankruptcies Involving Swedish IT-companies*. School of Business, Mälardalen University, Doctoral Thesis No. 53.

13. **Myreteg, Gunilla (2007),** *Förändringens vindar: En studie om aktörsgrupper och konsten att välja och införa ett affärssystem*. Företagsekonomiska institutionen, Uppsala universitet, Doctoral Thesis No. 131.

14. **Hrastinski, Stefan (2007),** *Participating in Synchronous Online Education*. School of Economics and Management, Lund University, Lund Studies in Informatics No. 6.

15. **Granebring, Annika (2007),** *Service-Oriented Architecture: An Innovation Process Perspective.* School of Business, Mälardalen University, Doctoral Thesis No. 51.

16. **Lövstål, Eva (2008),** *Management Control Systems in Entrepreneurial Organizations: A Balancing Challenge*. Jönköping International Business School, Jönköping University, JIBS Dissertation Series No. 045.

17. **Hansson, Magnus (2008),** *On Closedowns: Towards a Pattern of Explanation to the Closedown Effect*. Swedish Business School, Örebro University, Doctoral Thesis No. 1.

18. **Fridriksson, Helgi-Valur (2008),** *Learning processes in an inter-organizational context: A study of krAft project*. Jönköping International Business School, Jönköping University, JIBS Dissertation Series No. 046.

19. **Selander, Lisen (2008),** Call Me *Call Me for some Overtime: On Organizational Consequences of System Changes.* Institute of Economic Research, Lund Studies in Economics and Management No. 99.

20. **Henningsson, Stefan (2008),** *Managing Information Systems Integration in Corporate Mergers & Acquisitions*. Institute of Economic Research, Lund Studies in Economics and Management No. 101.

21. **Ahlström, Petter (2008),** *Strategier och styrsystem för seniorboendemarknaden*. IEI-EIS, Linköping universitetet och Tekniska Högskolan i Linköping, Doktorsavhandling, Nr. 1188.

22. **Sörhammar, David (2008),** *Consumer-firm business relationship and network: The case of "Store" versus Internet.* Department of Business Studies, Uppsala University, Doctoral Thesis No. 137.

23. **Caesarius, Leon Michael (2008),** *In Search of Known Unknowns: An Empirical Investigation of the Peripety of a Knowledge Management*

System. Department of Business Studies, Uppsala University, Doctoral Thesis No. 139.

24. **Cederström, Carl (2009),** *The Other Side of Technology: Lacan and the Desire for the Purity of Non-Being.* Institute of Economic Research, Lund University, Doctoral Thesis, ISBN: 91-85113-37-9.
25. **Fryk, Pontus, (2009),** *Modern Perspectives on the Digital Economy: With Insights from the Health Care Sector*. Department of Business Studies, Uppsala University, Doctoral Thesis No. 145.
26. **Wingkvist, Anna (2009)**, *Understanding Scalability and Sustainability in Mobile Learning: A Systems Development Framework.* School of Mathematics and Systems Engineering, Växjö University, Acta Wexionesia, No. 192, ISBN: 978-91-7636-687-5.
27. **Sällberg, Henrik (2010),** *Customer Rewards Programs: Designing Incentives for Repeated Purchase.* Blekinge Institute of Technology, School of Management, Doctoral Dissertation Series No. 2010:01.
28. **Verma, Sanjay (2010),** *New Product Newness and Benefits: A Study of Software Products from the Firms' Perspective*, Mälardalen University Press, Doctoral Thesis.
29. **Iveroth, Einar (2010),** *Leading IT-Enabled Change Inside Ericsson: A Transformation Into a Global Network of Shared Service Centres.* Department of Business Studies, Uppsala University, Doctoral Thesis No. 146.
30. **Nilsson, Erik (2010),** *Strategi, styrning och konkurrenskraft - en longitudinell studie av Saab AB*, IEI-EIS, Linköpings universitet och Tekniska Högskolan i Linköping, Doktorsavhandling, Nr. 1318.
31. **Sjöström, Jonas (2010),** *Designing Information Systems: A pragmatic account*, Department of Informatics and Media, Uppsala University.
32. **Numminen, Emil (2010),** *On the Economic Return of a Software Investment - Managing Cost, Benefit and Uncertainty*, Blekinge Institute of Technology, School of Management.

MIT

swedish research school of management and information technology

Contact person: Professor Pär Ågerfalk, director of MIT, Uppsala University
par.agerfalk@im.uu.se
Address: The Swedish Research School of Management and Information Technology, Department of Business Studies, Uppsala University, Box 513, 751 20 Uppsala
http://www.forskarskolan-mit.nu/mit/

ABSTRACT

The purpose of this dissertation is to explore how the economic return of a software investment can be assessed and managed. This topic has been studied in research and has been a concern for firms making software investments. In order to study this we need a model of the underlying factors affecting the economic return. Assessing and managing the return of a software investment is been argued to be difficult due to specific economic characteristics of a software investment, i.e. high degree of intangible consequences and uncertainty about the total investment cost. Given these characteristics it is has been concluded that it is difficult to derive a return function.

In this dissertation we question this conclusion and propose a comprehensive model to assess and manage the intangibles and the underlying uncertainty. The model is deduced from general assumptions of the economic behavior of the firm. To develop the model we analyze the relevance of intangibles in relation to the economic purpose of making a software investment. Based on this a new way of deriving a cash flow function for a software investments is defined.

Further it is analyzed how the underlying uncertainty of a software investment can be managed. The analysis uses a quantitative approach and methods from financial economics. It includes how the application of a real option and portfolio approach can reduce the uncertainty in a software investment and the role of efficient software platforms. The relation between software platforms and the opportunity to create different types of real options for future development is inferred from empirical studies. The studies in this dissertation show how a managerial view on a software investment corresponds with the overall economic goal of the firm. They also show how a strategic value of a software investment can be created, assessed and managed.

This study is part of the Swedish Research School of Management and Information Technology (MIT) which is one of 16 national Research Schools backed by the Swedish Government. MIT is jointly operated by the following institutions: Blekinge Institute of Technology, Gotland University College, Jönköping International Business School, IT University of Gothenburg, Karlstad University, Linköping University, Lund University, Mälardalen University College, Stockholm University, Växjö University, and Uppsala University, host of the research school. At MIT, research is conducted, and doctoral education provided, in three fields: management information systems, business administration and informatics.

BLEKINGE TEKNISKA HÖGSKOLA · BTH ·

2010:13

ISSN 1653-2090

ISBN: 978-91-7295-194-5