Evaluating Incremental/Iterative Software Projects by Valuing Investment/Implementation Risks

Vassilis C. Gerogiannis  
Department of Project Management, Technological Education Institute of Larissa, Greece  
Technological Education Institute of Larissa, 41110, Larissa, Greece  
gerogian@teilar.gr

Androklis Mavridis  
Department of Computer Science, Aristotle University of Thessaloniki, Greece  
Aristotle University of Thessaloniki, 54124, Thessaloniki, Greece  
mavr@altec.gr

Pandelis G. Ipsilandis  
Department of Project Management, Technological Education Institute of Larissa, Greece  
Technological Education Institute of Larissa, 41110, Larissa, Greece  
ipsil@teilar.gr

Ioannis Stamelos  
Aristotle University of Thessaloniki, 54124, Thessaloniki, Greece  
stamelos@csd.auth.gr

Abstract
The optimisation process of scheduling an incremental/iterative software project can be approached as a multi criteria decision problem, formulated by a linear programming model, aimed to propose alternative project schedules and examine cost trade-offs. In this paper, we apply the prominent economic theory of Real Options to analyse project investment risks and discover the economic value associated with each alternative scheduling decision. To justify our approach, we identify two options in an incremental/iterative project plan. The first option is to stall the development at a pre-defined increment/iteration, while the second is to continue increments and deliver the full system functionality. By calculating the expected value of each option, we provide the project manager with the flexibility to compare candidate schedules and decide, under favourable or unfavourable conditions, the most profitable combination of delivered functionalities.

Keywords: Software Project Management, Iterative/Incremental Software Projects, Risk Analysis and Decision Making, Real Options

1. Introduction

In an iterative/incremental software project life cycle, when the final software is iteratively built by developing gradually subsets of the required functionalities, iterations/increments can be performed during pre-determined periods, so called time boxes (Stapleton, 2003). Timeboxing is a suitable process model for executing software projects in which there is a strong requirement to deliver rapidly a working software system as well as for software projects of medium complexity which utilise a stable software architecture (Jalote et al., 2004). Timeboxing adoption in iterative/incremental software projects may enhance the predictability of software delivery times and manage possible risks of violating project
deadlines (i.e., iteration time boxes). Timeboxing divides a project plan into a sequence of stages (e.g., requirements analysis, design, implementation, testing and deployment) that are repeated iteratively by small dedicated development teams. Iterations can be performed in parallel to further reduce the overall project duration. Work parallelism is achieved by following a “pipelined” execution that is similar to instructions execution from hardware architectures. When a team completes the tasks of a stage, it hands over the stage deliverables to another team executing the next stage and then starts executing the same stage in the next timeboxed iteration. However, the high-level of coherence and expertise of development teams are not enough to guarantee that there are no work discontinuities between the same stages in successive timeboxed iterations. Although the same stage is repeated sequentially in different time boxes, violating the continuity of the work between stages in successive time boxes introduces coordination delays that increase the overall project cost and duration (Mookerjee and Chiang, 2002).

Figure 1. Timeboxing process model for iterative/incremental software development

The problem of finding optimum schedules for a software development project that follows timeboxing can be viewed from a multi criteria decision analysis perspective. In previous work, we have proposed multi criteria linear programming (LP) techniques to achieve different scheduling objectives (Gerogiannis and Ipsilandis, 2007; Gerogiannis et al., 2007). These objectives aim to improve the project performance (e.g., minimise project duration, iteration delays and work discontinuities between executions of the same stage in successive iterations) and/or reduce project cost elements (e.g., minimise cost of work discontinuities and iteration delay costs). The overall goal is to provide proper support for the software project manager in his/her decision to select from a set of alternative master project schedules the one that will achieve scope, time and cost requirements. However, this is a very difficult decision since there exist trade-offs among these requirements. In this paper, we concentrate mainly on time and cost trade-offs. Thus, our approach differs from other decision analysis methods applied to scope management and release planning of iterative projects, since they mainly focus on providing support for the selection and prioritisation of software requirements (Greer and Ruhe, 2004; Akker et al., 2008). We utilise a multi-objective linear programming model to produce a set of alternative project schedules (a portfolio of schedules), each one characterized by a corresponding ratio between iteration delay costs and work discontinuity costs. To further support this decision making process and analyse the investment risks associated with each alternative project schedule, we apply a real options approach. Real Options is a prominent financial/decision theory which addresses uncertainties inherent in project investments over time and facilitates adaptation of project management decisions to dynamic environments (Myers, 1977). The possibility to consider Real Options is particularly suitable in case of a software project (Sullivan et al., 1999; Tiwana et al., 2006), when the
project manager has the opportunity (but not the obligation) to make decisions in response to external and/or internal events (e.g., defer the development, expand the system functionalities, abandon the project etc.). By exploiting Real Options, we move forward the optimization decision of an iterative/incremental release plan to perform the cost valuation of different scheduling decisions. We argue for the proposed approach by examining the risk of two options in project case study: the option to stall (abandon) a project plan at a pre-defined iteration and the option to continue iterations (expand development) and deliver the full system functionality.

The paper is structured as follows: In section 2, we define a multi criteria Linear Programming (LP) model for incremental/iterative software project schedules. In Section 3, we briefly present an overview of Real Options applicability in software project management. In Section 4, we define the real options to be analyzed in an iterative/incremental project case study. In section 5, we employ a set of schedules for the case study and we apply a Real Options approach in order to demonstrate how the selection process of the most suitable project schedule can be supported. In the last section, we present the paper conclusions.

2. A Linear Programming Model for Iterative/Incremental Software Projects

In any iterative/incremental software project which follows timeboxing disciplines, we can identify that there is a set of $M$ stages and $P$ project dependency relationships (with or without time-lag). The project is divided into $N$ separate iterations in a “linear” way, where, without loss of generality, the following assumptions (originated from the timeboxing process model) hold: (i) all stages are performed in all iterations, (ii) a stage cannot be performed before the same stage is completed in the previous iteration, (iii) precedence dependencies remain the same in all iterations (i.e., the same planning method is followed). By adopting an AON (Activities on Nodes) network representation for the project life cycle, stages are represented as nodes and dependency relationships among stages are represented as arcs in the project network. Precedence dependencies can be of any type of the known relationships (Start-to-Start/SS, Finish-to-Start/FS, Start-to-Finish/SF, Finish-to-Finish / FF). Let $i = 1, 2, \ldots, M$ denote the project stages and $j = 1, 2, \ldots, N$ denote the project iterations. Scheduling of an iterative/incremental software project can be defined by a linear programming (LP) model as follows.

**Model Variables and Parameters.** Define:
- $d_{ij}$, the duration of stage $i$ in iteration $j$,
- $s_{ij}, f_{ij}$, the start and finish time respectively of stage $i$ in iteration $j$,
- $l_{ij}$, the minimum elapsed time for starting stage $i$ in iteration $j+1$ after finishing stage $i$ in iteration $j$,
- $P_i$, the set of predecessor stages to stage $i$,
- $E$, the set of all stages without successors,
- $WB_i$, the total time of work breaks for stage $i$ because of work discontinuities in successive iterations,
- $UC_j$, the completion time of iteration $j$,
- $D_j$, the promised delivery/release time for the software part produced in iteration $j$,
- $c_j$, the cost (per time unit) of delay in finishing iteration $j$ after the deadline,
- $f_i$, the cost (per time unit) of work breaks / discontinuities in stage $i$.

**Constraint definitions.** Define:
- Stage duration constraints:
  \[ f_{ij} = s_{ij} + d_{ij} \quad \forall \ i = 1, 2, \ldots, M, j = 1, 2, \ldots, N \]
- Project linearity constraints:
\[ s_{ij+1} \geq f_{ij} + l_{ij} \forall \quad i = 1, 2, \ldots, M, j = 1, 2, \ldots, N-1 \]

- **Technological dependencies:**

\[ s_{ij} \geq f_{kj} \forall \quad i = 1, 2, \ldots, M, j = 1, 2, \ldots, N, k \in P_i \]

- **Iteration completion time:**

\[ UC_j \geq f_{kj} \quad \forall \quad j = 1, 2, \ldots, N, k \in E \]

*UC*\(_j\) is the completion time for iteration \(j\) and \(UC_N\) is the project duration.

- **Resource delays (work breaks / discontinuities):**

\[ WB = \sum_{j=1}^{N} (s_{ij+1} - f_{ij}) \quad \forall \quad i = 1, \ldots, M \quad WB = \sum_{i=1}^{M} WB_i \]

**Global Objective Function.**

\[
\text{Minimize } \sum_{j=1}^{N} c_j (UC_j - D_j) + \sum_{i=1}^{M} f_i WB_i
\]

Depending on the values of the cost parameters \(c_j\) and \(f_i\), the above general objective function can be used to achieve different objectives or analyze trade-offs between the cost parameters. The function generates a set of alternative project schedules. Examples include: i) a schedule that minimises the project duration (set \(c_N\) equal to 1, rest of \(c_j\) and \(f_i\) equal to 0), ii) a schedule that minimises the total work break / discontinuity time (set all \(f_i\) equal to 1 and all \(c_j\) equal to 0), iii) a schedule that minimises the completion time of iterations (set all \(f_i\) equal to 0 and all \(c_j\) equal to 1), iv) a schedule that minimises the total cost of work breaks / discontinuities (set all \(c_j\) equal to 0), and v) a schedule that minimises delay costs (set all \(f_i\) equal to 0). In addition, sensitivity analysis on the parameters of this objective function can be used to examine optimum schedules at different levels of cost relations by considering the ratio of iteration delay costs to the costs of work discontinuities (Gerogiannis and Ipsilandis, 2007).

3. **Real Options in Software Project Management**

Research that has been conducted since the mid of 90s, oriented towards the employment of financial theories in software engineering application areas, has addressed the issues of separating the value of a software product from its cost, maximising the value added by a given software project investment as well as valuing the hidden intangibles behind software development. An example of such research initiatives are the Economic Driven Software Engineering annual workshops (EDSER, 2006). Within this research context, there are efforts towards the exploitation of the economic theory of Real Options (Myers, 1977), in order to analyze, in a monetary fashion, the economic value that different software investments could generate (Amram and Kulatilaka, 1999; Benaroch, 2002). The problem can be stated as follows: what is the most appropriate option (from a portfolio of options) that can result in the best value of a software product, process or project? Hence, the problem of software valuation can be viewed as a decision making process that takes place under uncertainty and incomplete knowledge. These uncertainties include the cost and schedule required to develop a software product, the software requirements which are likely to change in the future, the presence of software faults and failures, the impact of process/technology changes on cost and scheduling elements, etc. (Sullivan et al., 1999). The core idea is to cope with these exogenous and endogenous uncertainties and mitigate the corresponding risks in the project investment. Such prediction is necessary for valuing the long-term investment of adopting a particular software development life cycle.

Classical financial techniques (Discounted Cash Flow - DCF and Net Present Value-NPV) fall short in dealing with flexibility and uncertainty in decision making (Schwartz and Trigeorgis, 2000). Real options theory overcomes inabilities of conventional budgeting...
techniques by addressing the strategic value of a project investment. A real option gives the “option holder” the right, but not the obligation, to evolve the project opportunities by making follow-up investments (e.g., consider cases of reuse, expand the range of provided functionalities, follow-up or terminate the project, explore new markets etc.). If conditions favourable to investing arise, the project manager can exercise the option by investing the strike price defined by the option. Therefore, real options have been found a suitable approach to introduce flexibility, facilitate active software project management, and, consequently, handle the dynamic nature and uncertainty of a given software project investment (Wu et al., 2007).

The approach of planning an iterative/incremental software project by taking into account multiple factors (e.g., iteration completion times, project duration and work discontinuities for work teams) and trade-offs between them, offers the software project manager with the flexibility to consider a set of optimum schedules. However, when selecting an appropriate schedule for a project, based only on the cost trade-offs and the project static expected value (NPV), the software project manager fails to consider future uncertainties and, hence, to tally for project risks. These uncertainties may range from internal ones, such as an unexpected delay of a specific stage in a certain iteration, to external ones, such as the introduction to the market of a new development tool/technique that can ease the workload or minimize cost. Each possible scheduling option has a different inherent value (an option value) which is the value of the produced software, if this schedule is to be adopted. The manager has the right to exercise this option and he/she can do so, if business conditions become favourable for the project success.

4. Valuing Options in an Iterative/Incremental Software Project

We will discuss our approach through a hypothetical scenario of a software company that follows the timeboxing process model for developing iterative/incremental software projects (Jalote et al., 2004). Before a project commences, the company’s R&D management utilizes the LP model presented in section 2, identifies a set of alternative schedules and justifies them since, under certain conditions, a different schedule can be the most suitable (in terms of project duration, iteration / work discontinuity delays, cost elements, delivered functionalities etc.). After the schedules identification, the company’s management board will estimate the best schedule profitability. One possible solution is to calculate statically the NPV for all candidate schedules, based on the estimated project development costs (including delay costs) and the expected free cash flows. However, a combined Real Options–NPV approach can provide a better way to deal with project uncertainties and “discover” the “hidden value” within all possible schedules. The management board builds a step wise scenario, for each candidate schedule, that involves a review of the incremental delivery plan, to examine the delivered functionality (i.e., the number of requirements developed) at a certain point of the development life-cycle (i.e., at a certain iteration) when a working pilot application is planned to be available.

To simplify discussion, we make the assumption that in the presented example all necessary preparatory tasks (i.e., before identifying and analyzing the real options to be investigated) have been already performed, prior to the presented analysis. These steps typically include, but are not limited, to (Sullivan et al., 1999; Tiwana et al., 2006): (i) identification of the real assets of a software project to be analysed by real options (e.g., development costs and future cash flows), (ii) monitoring the important project uncertainties and approximate the probability distribution of these uncertainties.

The example software project life cycle follows a set of 6 stages which are executed in an iterative/incremental approach. These stages are Domain Modeling (stage A), Use-Cases
Analysis (stage B), Requirements Review (stage C), Preliminary Design & Review (stage D), Detailed Design & Review (stage E) and Coding & Testing (stage F). The work of each stage is done by a small stage-specific team (2-3 experts) and all iterations should be timeboxed. The final software application is originally scheduled, according to the incremental delivery plan, to be delivered after 6 iterations of these 6 discrete stages. The AON diagram in Figure 2 depicts all stage relationships (they are FS relationships) along with the most likely estimate (in weeks) of the duration of each stage, at each of the 6 iterations. The critical path of the entire project consists of stages A and C in iteration 1, the sequence of stage E in all iterations, and stage F in iteration 6.

![AON diagram for the example project](image)

*Duration of each Stage (in weeks) at each of the 6 iterations. Stage C is not included in the 6th iteration.
Due to technological constraint there is a lag of 2 weeks between the finish of Stage A and the start of Stage C (Lag=2).
There is a 5 weeks delay in stage B between iterations 3 and 4.

**Figure 2. AON diagram for the example project**

In this case project an intermediate review will take place before the beginning of the third iteration. If the delivered functionality at that point is the promised and the conditions (external: market competency, economic situation etc. / internal: company status, company policy, product uncertainty, resources uncertainty etc.) are favourable, then the management board decides to continue with the development of the rest of the iterations. Otherwise, the board decides to stall development and seek for salvage portion of the costs. The first option (to stall development) is the Option to Abandon, while the second (to proceed with the full product development) is the Option to Expand (Wu et al., 2007). The application of such approach within the context of an incremental/iterative life cycle might support the successful development of a pilot software application in a tight schedule. The management board conceives the first batch of iterations as a pilot application for the whole project. If the pilot application meets the company’s standards/customer’s expectations and the conditions for its full development are favourable, the company continues funding and proceeds to the full scale/full functionality product.

### 5. Project Schedules Evaluation

Having defined the project network structure (Figure 2), the project duration (48 weeks) as well as the number of iterations/increments (6), the development work is constrained by a strict time plan of 1 year (48 working weeks) and a fixed budget (initial outlay) of 30,000€.
The R&D management suggests evaluation of three candidate schedules, in terms of their profitability: (i) scheduling stages according to their Earliest Start (Finish) time, (ii) scheduling stages according to their Latest Start time, and (iii) scheduling stages to minimize work discontinuities without extending the overall project duration (48 weeks).

5.1 Alternative Schedules
The R&D management, by setting in the global objective function presented in section 2 set all $f_i$ equal to 0 and all $c_j$ equal to 1, obtains the schedule that minimizes both the project duration and the completion time (i.e., the release time for software parts) of all iterations. This is actually the schedule produced by the Critical Path Earliest Start (Finish) Method (CPM EFT). Figure 3 presents a linear diagram that describes CPM EFT. The progress of each stage through the project iterations is represented by a piecewise straight line. The slope of this line corresponds to the “production rate” in a specific stage at each of the 6 iterations. Horizontal segments on the progress line correspond to work discontinuities between executions of the same stage in successive iterations. Vertical segments represent cases where a stage is planned not to be performed in the corresponding iteration (e.g., the Requirements Review stage (C) in the fifth iteration).

Figure 3. CPM Earliest Start/Finish Time – CPM EFT

CPM EFT can be considered as an “under-estimate” schedule that provides a minimum bound for the time box duration of each iteration (i.e., the earliest time that iteration 1 should be completed is before 18 weeks, iteration 2 should be before 24 weeks etc.). The danger with an under-estimate schedule is the effect on software quality, since obtaining partial software deliveries, as early as possible, could affect negatively the software quality. CPM EFT is actually a baseline for the rest of the analysis. It is a “too optimistic” plan that results in minimum values of completion times for all iterations. Next, the R&D management considers scheduling tasks according to their Latest Start (LS) times, as it is demonstrated in Figure 4. CPM LS schedule is obtained by pushing stages to their LS times and transferring work discontinuities from the last project stages to those in the beginning. The LP model can be also solved towards the objective of total work discontinuity minimization. This is achieved by introducing the 48 weeks of CPM duration in the global objective function and setting all $f_i$ equal to 1 and all $c_j$ equal to 0. The resulting schedule is shown in Figure 5 (CPM WD). The minimum duration of 48 weeks is achieved with a minimum of 26 weeks of work discontinuities at stages B and C. A further reduction of work discontinuities is not possible without extending the project duration beyond 48 weeks. Iteration delays, in both CPM LS and CPM WD, have been calculated from the corresponding minimum iteration completion times (i.e., the minimum time boxes) derived from the baseline schedule (CPM EFT).
Figure 4. CPM Latest Start Time – CPM LS

Figure 5. Minimizing work discontinuities – CPM WD

5.2 Calculating Net Present Values

The R&D management delivers to the company’s management board the baseline schedule (CPM EFT) and the two alternative schedules. The board initially calculates the NPV for the project by considering an one-off implementation for each schedule. As CPM EFT minimizes the project duration, by obtaining the software delivery as early as possible, this selection could affect positively the financial performance of the project, especially in the particular project case, where the management board reviews the first batch of iterations, to decide continuing funding the full scale/full functionality product development. Hence, the NPV of CPM EFT will be an indication of the desired (ideal) expected profit. We also assume that time series analysis or multivariate regression of historical or comparable project data (past undertaken projects employing CPM EFT, CPM LS and CPM WD schedules with similar time constraints and overall budget) has been applied to forecast the free cash flows at each iteration as well as the terminal expected values of the project for each schedule.

A terminal expected value for the whole project refers to the value of the project at the end of the growth period (48 weeks). For the 6 project periods (iterations/ increments),
assuming that the annualized discount rate is equal to 12%, the compound discount rate can be calculated equal to 12.61%, by using the following expression:

\[
\left(1 + \frac{\text{discount \ rate}}{\text{periods}}\right)^{\text{periods}} - 1
\]

We finally assume that the free cash flows at each time box are the revenues coming from diffusion of project results to other development company streams. The management board estimates a 50% probability for the software product marketing success, due to market uncertainty and the very strict estimate of the development duration (48 weeks). The NPV calculation for CPM EFT (Table 1(a)) results in an expected revenue discounted by 50% (the success probability), that is equal to 41.025€ (82.050€ x 0.5), a 50% discount of the initial outlay that is equal to 15.000€ (30.000€ x 0.5), and a final expected value for CPM EFT that is equal to 26.025€ (41.025€ - 15.000€). Accordingly, the NPV calculation for CPM LS (Table 1(b)) results in an expected revenue that is equal to 35.620,1€ (71.240,3€ x 0.5) and a final expected outlay that is equal to 15.000€ (30.000€ x 0.5). Thus, the expected value for CPM LS is equal to 20.620,1€ (35.620,1€ - 15.000€). Finally, calculating the NPV for CPM WD (Table 1(c)) results in an expected revenue equal to 37.784,2€ (75.568,4€ x 0.5), a final expected outlay equal to 15.000€ (30.000€ x 0.5) and an expected value for CPM WD that is equal to 22.784,2€ (37.784,2€ - 15.000€).

<table>
<thead>
<tr>
<th>(a) NPV for CPM EFT Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration Number</td>
</tr>
<tr>
<td>Initial Outlay</td>
</tr>
<tr>
<td>Cash Flow</td>
</tr>
<tr>
<td>Terminal Value</td>
</tr>
<tr>
<td>Net Cash Flow</td>
</tr>
<tr>
<td>Discounted Rate</td>
</tr>
<tr>
<td>Present Value</td>
</tr>
<tr>
<td>Net Present Value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(b) NPV for CPM LS Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration Number</td>
</tr>
<tr>
<td>Initial Outlay</td>
</tr>
<tr>
<td>Cash Flow</td>
</tr>
<tr>
<td>Terminal Value</td>
</tr>
<tr>
<td>Net Cash Flow</td>
</tr>
<tr>
<td>Discounted Rate</td>
</tr>
<tr>
<td>Present Value</td>
</tr>
<tr>
<td>Net Present Value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(c) NPV for CPM WD Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration Number</td>
</tr>
<tr>
<td>Initial Outlay</td>
</tr>
<tr>
<td>Cash Flow</td>
</tr>
<tr>
<td>Terminal Value</td>
</tr>
<tr>
<td>Net Cash Flow</td>
</tr>
<tr>
<td>Discounted Rate</td>
</tr>
<tr>
<td>Present Value</td>
</tr>
<tr>
<td>Net Present Value</td>
</tr>
</tbody>
</table>

| Table 1. NPV Calculations |

5.3 Applying Real Options

In this step, the management board considers that the project can be deferrable and additional development can be undertaken only if favourable conditions are valid in the future. In terms of Real Options, the value of two options will be evaluated for both CPM
LS and CPM WD schedules: stall development at a specific iteration (Option to Abandon) or continue with the full product development (Option to Expand). In particular, the management board examines what would be the profits of the schedules in case when, instead of having the project implemented continuously from iteration 1 to iteration 6, development executes the first two iterations and then, if there are “favourable” conditions, the company has the option to continue funding the project and further implement the next four iterations. If not, then the management board has the option to abandon the project and loose only the initial investment for the first two iterations. From the total amount of the initial investment (30,000€), the management board will consider the decision to further invest an amount of 20,000€ after the second iteration. The initial cash outlay for the first two iterations is considered equal to 10,000€ for both CPM LS and CPM WD schedules.

(a) Cash Flows under CPM LS Schedule

<table>
<thead>
<tr>
<th>Iteration Number</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Outlay</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cash Flow</td>
<td>10,000</td>
<td>12,000</td>
<td>15,000</td>
<td>12,000</td>
<td>11,000</td>
<td>10,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminal Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>110,000</td>
</tr>
<tr>
<td>Net Cash Flow</td>
<td>-10,000</td>
<td>10,000</td>
<td>12,000</td>
<td>-20,000</td>
<td>15,000</td>
<td>12,000</td>
<td>11,000</td>
<td>120,000</td>
</tr>
<tr>
<td>Discounted Rate</td>
<td>0%</td>
<td>12.61%</td>
<td>12.61%</td>
<td>0%</td>
<td>12.61%</td>
<td>12.61%</td>
<td>12.61%</td>
<td>12.61%</td>
</tr>
<tr>
<td>Present Value</td>
<td>-10,000</td>
<td>8,880.2</td>
<td>9,463.7</td>
<td>-20,000</td>
<td>10,504.2</td>
<td>7,462.6</td>
<td>6,077.3</td>
<td>58,852.3</td>
</tr>
</tbody>
</table>

(b) Cash Flows under CPM WD Schedule

<table>
<thead>
<tr>
<th>Iteration Number</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Outlay</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cash Flow</td>
<td>10,000</td>
<td>13,000</td>
<td>16,000</td>
<td>14,000</td>
<td>13,000</td>
<td>11,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminal Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>110,000</td>
</tr>
<tr>
<td>Net Cash Flow</td>
<td>-10,000</td>
<td>10,000</td>
<td>13,000</td>
<td>-20,000</td>
<td>16,000</td>
<td>14,000</td>
<td>13,000</td>
<td>121,000</td>
</tr>
<tr>
<td>Discounted Rate</td>
<td>0%</td>
<td>12.61%</td>
<td>12.61%</td>
<td>0%</td>
<td>12.61%</td>
<td>12.61%</td>
<td>12.61%</td>
<td>12.61%</td>
</tr>
<tr>
<td>Present Value</td>
<td>-10,000</td>
<td>8,880.2</td>
<td>10,252.3</td>
<td>-20,000</td>
<td>11,204.4</td>
<td>8,706.4</td>
<td>7,182.3</td>
<td>59,342.8</td>
</tr>
</tbody>
</table>

Table 2. Estimated Cash Flows for the Project

Estimation of failure/success probabilities, in terms of the economic value of the various project decision elements, can be performed by empirical analysis on historical data from previous company projects. This process can be also supported by automated instrumentation tools (Costa et al., 2007). In our example, the management board has estimated, when considering the CPM LS schedule, a 33% probability of failure for the initial project phase (iterations 1-2) and a 67% probability of success. If the management board gives the approval for the additional 20,000€ fund, then it is estimated a 25% probability of failure and a 75% probability of success.

The future cash flows for the project under the CPM LS schedule are presented in Table 2(a). The expected outlay of the option to abandon is equal to 3.300€ (10,000€ x 0.33), while the expected outlay of the option to expand is equal to 5.100€ (30,000€ x 0.17). The expected revenue for CPM LS has been previously calculated by NPV analysis that is equal to 35,620,1€. Thus, the expected final value for CPM LS is equal to 27,220,1€ (35,620,1€ - 8,400€). Similarly, when considering the CPM WD schedule, the management board acknowledges an 80% probability of success for the initial project phase (iterations 1-2), and hence a 20% probability of failure. The reason for this optimistic estimate is that CPM WD presents less “slack times” and thus may result in achieving a high level of work continuity and a smooth flow of development work over the initial two iterations. If the board takes the option to expand and invest the additional 20,000€ fund, then it is estimated a 25% probability of failure and a 75% probability of
success. Table 2(b) presents the estimated future cash flows for the project under the CPM WD schedule. The expected outlay of the option to abandon is equal to 2,000€ (10,000€ x 0.20), while the expected outlay of the option to expand is equal to 9,000€ (30,000€ x 0.30). The expected revenue for CPM WD has been calculated previously by the NPV analysis that is equal to 37,784,2€. Therefore, the expected final value for CPM WD is equal to 26,784,2€ (37,784,2€ - 11,000€).

5.4 Retrospect the Analysis
We notice that applying real options and giving the management board a different view of the potential risks - and hence the flexibility to adjust the incremental delivery plan - we derive a different suggestion, compared to the corresponding indication produced by calculating NPVs. With NPV, the management board is advised to select the CPM WD schedule, as it is expected to result in an amount of profit equal to 22,784,2€, instead of 20,620,1€ expected from CPM LS. With real options though, the decision should be different. Estimating the risks accordingly, the expected value of CPM WD (26,784,2€) is less than this of CPM LS (27,220,1€). This can be explained by closely considering the two schedules’ characteristics (Figures 4 & 5). Both CPM WD and CPM LS present the same iteration delays but CPM LS, due to its latest pushing time characteristic, may provide a better managerial flexibility. Pushing stages to their LS times transfers coordination delays (work discontinuities) from the last stages to those in the beginning of the project. In both schedules the work discontinuities have been transferred from the final stages (stages D, E and F) to those in the beginning (stages A, B and C). However, the total work discontinuity time for stages A, B and C in CPM LS is equal to 48 weeks, while the same stages in CPM WD present a total work discontinuity equal to 26 weeks. Thus, the R&D management may review unexpected changes and coordination delays earlier in the project. Furthermore, minimisation of work discontinuities may negatively affect the coordination time between project stages. On one hand, this may improve the smooth flow of development work; on the other, the risk of having a low defect removal efficiency in early iteration stages is increasing (and consequently in later stages). Coordination is affected by dynamic factors that cannot be easily predicted, due to the differences in the intensity of coordination needed at different project stages. In general, since the development teams’ knowledge improves with time, a lot of coordination may be needed early in the project (Mookerjee and Chiang, 2002).

6. Conclusions
In this study, we approached the optimisation process of scheduling an incremental/iterative software project as a multi criteria decision problem. By applying real options, we have performed cost valuation of different scheduling decisions. We have examined the risk of two options: i) stall (abandon) an incremental delivery plan at a pre-defined iteration or ii) continue iterations (expand development) and deliver the full system functionality. To demonstrate the usefulness of the approach, we calculated the static discounted Net Present Values of selected schedules in a project case study, and then we compared these values with those resulting from real options. The analysis highlighted how real options can provide increased managerial flexibility as they force management to consider investment risks associated with the alternative project scheduling decisions, as unexpected events in one stage or iteration may affect not only the iteration completion/delivery times but also the work continuity in project resources. Furthermore, we discussed that applying real options analysis can be useful to discover knowledge concerning the value of candidate schedules to be adopted in
iterative/incremental projects in a retrospective manner and, thus to enable decision makers/ project managers to better manage the scheduling alternatives. For future work, we are interested in considering real options as a possible tool for the selection and prioritization of features to be delivered in each software iteration/ increment.

References