UNCERTAIN CONTEXT FACTORS IN ERP PROJECT ESTIMATION ARE AN ASSET: INSIGHTS FROM A SEMI-REPLICATION CASE STUDY IN A FINANCIAL SERVICES FIRM

MAYA DANEVA

University of Twente, Computer Science Department
Drienerlolaan 5, 7500 Enschede, The Netherlands
m.daneva@utwente.nl

This paper reports on the findings of a case study in a company in the financial services sector in which we replicated the use of a previously published approach to systematically balance the contextual uncertainties in the estimation of Enterprise Resource Planning (ERP) projects. The approach is based on using three techniques, a parametric model, namely COCOMO II, a portfolio management model, and Monte Carlo simulations. We investigated (i) whether the adjustment of uncertain cost drivers in the COCOMO II model increases the chance of project success in a portfolio of ERP projects, (ii) which cost drivers of the COCOMO II model can be adjusted in a way that maximized the chance of portfolio success under time constraints, and (iii) which cost drivers of the COCOMO II model can be adjusted in a way that maximized the chance of portfolio success under effort constraints. We found that 11 COCOMO II cost drivers can be changed so that the change impacts the project outcomes under both time and effort constraints. This result is different from the result in the first case study in which 13 such factors were found.

Keywords: Project portfolio management; project effort estimation; enterprise systems; empirical evaluation; requirements-based estimation.

1. Introduction and Motivation

In the field of software cost estimation for large projects, it seems today that there is a common agreement that the more time organizations invest in upfront planning the more certain they can be that their project would be within budget. However, organizations who are engaged in adopting the Enterprise Resource Planning (ERP) technology have more often than not short selection processes (called ‘request-for-proposal) that are pressured by bidding consulting services companies that compete with each other with their project implementation proposals and pricing. As a result, a company that considers launching complex ERP projects enters a precarious situation: on one side at the stage of project planning, the understanding of the business requirements is often incomplete and project factors that could impact the project success are largely unknown or only partly known; on the other side, the ERP client
must make project decisions according to the predefined timeline established in the
request-for-proposal process, that is, the ERP client commits to a certain deadline
by which they will make their commitment to a consulting company and a project
identified that companies are interested in increasing their skill levels, capabilities
and levels of tool support in developing realistic business cases that are integral
part of the early requirements process. The survey found that half of the companies
in the Netherlands are keen on estimating the costs and the benefits in the starting
stages of their research to determine whether or not it makes sound business sense
to continue the process of evaluating vendors and/or consulting partners. These
companies wanted to be able to validate the price of the project offered by exter-
nal consultants against internal data they might have collected in company-specific
project databases. In a previously published research [2], we stated that for ERP
adopters to be adequately prepared for this endeavor, they need to acquire knowl-
edge (i) on how each consulting firm arrived at the bidding price and (ii) on how
realistic the effort estimation figures which one sees on the bidding document, are.

This paper draws on this earlier research that included an approach and a case
study to balance uncertainty of context in the very early requirements stages when
an organization undertakes ERP project estimation. The present paper represents a
semi-replication case study in which we demonstrate the application of the approach
to a new and different setting. Semi-replication means that we will consider for
inclusion similar projects and apply similar techniques, however the original research
setup would be adapted to those specific aspects of the organization for which the
original case study and the replicated case study differ.

Our previously published research [2] proposed how uncertainties of context can
be traded-off in an organization, so that the chances for the ERP project success
increase. We designed a multi-technique-based approach that rests on decomposi-
tion of a large ERP project into subprojects and algorithmic estimations on them,
where the role of uncertainty of context characteristics is analyzed by using proba-
bility distributions. Our case study research demonstrated that when incorporating
traditional effort/duration modeling concepts, portfolio management concepts, and
variation concepts, it is possible to find a tradeoff between ERP projects requiring
more effort than expected and those requiring less. We identified project context
characteristics, which, when changed, can add up to a successful ERP implemen-
tation. We felt encouraged from these results, and in this second study, we applied
the approach to understand if the results would be observed in another organization
(that also operates in another business sector).

The paper proceeds as follows. In Sec. 2, we provide background information
on the uncertainty challenges in the ERP effort estimation practice, the solutions
published in literature in information systems management and software economics
and measurement, and the difficulties in using these solutions. In Sec. 3, we present
our replication research process and the techniques that make up our approach to
handling project uncertainties. Section 4 provides insights into the empirical data
for our analysis. Section 5 discusses validity concerns and compares the results of this study and those in our previously published study [2]. Section 6 presents conclusions and implications for practice and future research.

2. Background and Related Work

2.1. The challenges with uncertainties in ERP project estimation

This section presents the topic of uncertainty in ERP projects from effort estimation perspective and is to help the readers understand the rest of the paper and to avoid misunderstandings. Our belief is that for ERP adopters to successfully deploy an effort estimation technique, they must first understand how ERP projects are different from custom software development or from other types of complex information systems (e.g. workflow management systems). We researched this question from the standpoint of software effort estimation and we found that ERP effort estimation practitioners and researchers [4–19] discuss the challenges with project uncertainties from several perspectives, each of which entails specific issues:

- Who shares and owns the ERP solution and who has the authority to make decisions over (which parts of) the solution? ERP projects do not necessarily have an identifiable owner at overall-solution-level because the solution is shared among business units (or sometimes independent companies). Parts of the system are owned by different units each bearing part of the costs and each absorbing part of the benefits, whereby the cost and benefits are not evenly distributed among the business units.

- In what way is the ERP solution complex? The growing complexity of the inter-organizational business processes means growing complexity in the ERP solution that embeds these processes [11].

- How much architecture is in the scope? ERP systems are not “built” in the sense that a master architect envisions the parts and their relationships; rather they evolve into existence and change through their life cycles as new shared pieces of functionality are built, existing intra-organizational systems connect to become shared, and shared parts of the system are disintegrated as soon as needs of sharing processes and data disappear.

- What are the configuration-specific challenges? Configurations create system capabilities far beyond the sum of the ERP components’ individual capabilities, which, allows the resulting system to qualitatively acquire new properties as result of its configurations. Each one implies the presence of cost drivers unique to each configuration.

- What concept of project duration is assumed? An ERP project delivers a system which is incomplete once the ERP project is over, because an ERP solution must mirror rapidly-changing business requirements, and so be adjusted regularly to accommodate current business needs.
What is estimated?

Solution techniques

Cost estimation challenges

Fig. 1. The “cycle of ERP cost estimation challenges”.

- What concept of size is assumed? There are at least three notions of ERP “size” [6]: “size” can be referred to as an attribute of the implementation tasks (e.g. “size” is defined as the number of ERP transactions to be configured), as an attribute of the ERP user base (e.g. the number of users), or as an attribute of the ERP functionality (e.g. number of function points).

Figure 1 shows the connections between the six perspectives. It indicates that the question of “what is estimated in ERP project settings?” causes specific challenges which, in turn, are countered by specific solutions (addressing specific parts of the cost estimation problem), which themselves potentially give rise to new challenges. The fact that an ERP adopter may choose for their project a specific decision-making authority model and assumes a specific definition of “duration” or of “size” (this is in the box labeled “What is estimated?”), causes specific challenges, which in turn are countered by specific solutions. For example, Function Point Analysis can serve as a solution, suppose the ERP adopter defines “size” in terms of ERP software functionality being used in support of business processes. The latter, however, gives rise to some new challenges: for example, to define what to count as an unit of functionality and how to identify it and count it [3, 16]. Similarly, the real-option analysis [20] technique can serve as a solution to the challenge of valuation of ERP assets, and it also poses new challenges like who of all stakeholders to consider “important” in the valuation process.

2.2. Related work

As this paper reports on research efforts that follow-up on [2], our related work section draws on two literature reviews [2, 4] that we published previously. We make the note that the related publications that we could possibly use for this section are those that we have already reviewed and evaluated in [2]. Therein, we investigated the specific sources of uncertainty in ERP project estimation cited in the current literature and aggregated the solution ideas proposed by the authors that previously published on uncertainty of software project context as well as of ERP project context, specifically. However, to help readers understand how this research stands with respect to existing relevant literature sources, we provide here
a summary of the literature reviews in [2] and [4]. Our reviews were focused on two areas: (1) ERP implementation, and (2) software economics.

In the body of published Management Science research within the area of ERP implementation, we found that researchers [20] are occupied predominantly with the investigation of how financial valuation techniques can help in dealing with project context uncertainties. For example, real option analysis and portfolio management models are two approaches deemed valuable when it comes to evaluating investments in large ERP assets under uncertainty. Management Science researchers, however, warn that although these techniques are promising, their practical application is far from straightforward for at least two reasons: (i) the techniques are very data-intensive and very few organizations among the ERP adopters have collected data to feed into the mathematical models implied, and (ii) they heavily rely on the presence of financial valuation experts specialized in the application of these techniques to IT (or ERP) assets [20].

Furthermore, our review on research published by the software economics yielded a variety of solution proposals geared more often than not towards coping with “epistemic uncertainty” [21]; this is the understanding that uncertainty results from inaccurate or incomplete information and can be reduced or even eliminated given better models or additional observations. In [4], we found that the software economics authors made substantial progress converging efforts towards incorporating uncertainty-handling techniques into traditional effort estimation models (e.g. COCOMO II [22]) by using, for example, concepts of fuzzy sets theory or probability theory [24]. For example, instead of using “data points” as inputs into algorithmic models of effort estimation, one could and should consider representing uncertain inputs by using probability distributions. These uncertain inputs are, then, processed by means of some simulation techniques, for example a Monte Carlo simulation [23, 24] or a Latin Hypercube simulation [24]. For estimation analysts, a recognized [40] advantage of using such an approach is the ability to calculate the implications of uncertainties (by means of the simulations themselves). We make the note that one important conclusion of our earlier review that has implications for this research was the following: we found that, very few of the literature sources in our review have studied the extent to which the problem of ERP cost estimation matches the assumptions of the underlying techniques being used to account for uncertainty in ERP project estimation. This leaves researchers and practitioners with incomplete understanding on when and in what context what technique to use.

3. The Replication Research Process

The research method we used in this work is a replicated case study. It was inspired by Yin [25]. In the replicated case, we use the same guidelines for case study research as in [2]. However, as already indicated in the Introduction, we adapted
the techniques to the case study site. This adaptation was justified, as research methodologists suggest that no two empirical studies are identical. This is because empirical research takes place in real life settings and no two organizations are the exactly same, even if they are operating in the same business sector.

We will provide a thorough discussion on the specific settings in our case organization in Sec. 4. However in the rest of this section, we first present our approach that we used both in the case study [2] and in the present study.

The overall approach is depicted in Fig. 2. It is based on the complementary application of multiple techniques: (i) the COCOMO II reference model [22] that lets us account for ERP adopter’s specific cost drivers, (ii) the Monte Carlo simulation [24] which lets us approach the cost drivers’ degrees of uncertainty, (iii) the effort-and-deadline-probability-based portfolio management concept [27] which lets us quantify the chance for success with proposed interdependent deadlines for a set of related ERP projects. The choices that motivated this combination are discussed in [2] and are outside the scope of this paper. Below we provide a description of each technique.

3.1. COCOMO II

COCOMO II [22] is one of the best-known algorithmic model for setting budgets and schedules as a basis for planning and control. It comprises:

- Five scale factors, which reflect economies and diseconomies of scale observable in projects of various sizes; that is, these parameters have large influence on big projects and small influence on small projects;
- 17 cost drivers, which serve to adjust initial effort estimations; these are called effort multipliers (EM) and mean parameters which effect effort the same amount regardless of project size.

In COCOMO II, the degrees of both the scale factors and the cost drivers vary from extra low, very low, low and nominal to high, very high and extra high. Suppose
ERP project stakeholders assign a degree to each scale factor and cost driver, the estimation of project effort and duration will result from the two equations below:

\[
\text{Effort} = A \times (\text{Size})^E \times \prod_{i=1}^{17} EM_i \quad (1)
\]

and

\[
\text{Time} = C \times (\text{Effort})^F, \quad (2)
\]

where \(E\) and \(F\) are calculated via the following two expressions, respectively:

\[
E = B + 0.01 \times \sum_{j=1}^{5} SF_j \quad \text{and} \quad F = D + 0.2 \times (E - B).
\]

In (1) and (2), \(SF\) stands for the scale factors, and \(EM\) means cost drivers.

3.2. The Monte Carlo simulations

The term “Monte Carlo” refers to the field of applied and computational mathematics and denotes a broad family of techniques used to approximate such quantities as integrals and sums of random variables, for which analytic, closed-form formulas are not available because of the form or complexity of the situation [21]. In the area of software effort estimation, a Monte Carlo simulation technique is used to identify variation in the results as a function of the uncertainty of the inputs. It is a problem-solving technique helping to approximate the probability of certain outcomes by running multiple trial runs, called simulations, using random variables. A Monte Carlo simulation approaches the inherent uncertainty of cost drivers by letting inputs vary according to specific statistic distributions, for example, in Fig. 3,
we present normal, triangular, and uniform distributions. When used in combination with COCOMO II, repeatedly running the model many times and collecting samples of the output variables for each run helps the estimation analysts produce an overall picture of the combined effect of different input variables distribution on the output of the model.

Our motivation for choosing a Monte Carlo simulation technique as a component of our approach to ERP project uncertainty included the following:

- It was already used by project estimation analysts at large government agencies as the THAAD Project Office (USA). Monte Carlo simulation was also used at Jet Propulsion Lab [24, 26] and RAND Corporation [23].
- It has the reputation of a well-studied, and well-understood numerical technique with an accumulated body of literature of its own [21].
- It can provide a final cost probability distribution directly, without the necessity of first doing a deterministic cost estimate (a cost point estimate can be derived from any desired function of the probability distribution, such as the mean, median, or mode) [21].

The specific step-by-step procedure in which we applied the Monte Carlo simulation technique was NOSTROMO, originally devised by the THAAD Project Office [24].

3.3. The portfolio management concept

We couple the above techniques with a portfolio management concept [27] based on an effort-and-deadline-probability model that allows us to quantify the uncertainty associated with a project estimate. The motivation for choosing this model is described in more detail in [2]. In summary, we chose it because (i) it is applicable at the stage of requirements or project bidding, (ii) its only input requirement is a record of previous projects; (although it does require an effort estimate for every project, it need be nothing more sophisticated than a subjective opinion [27]); and (iii) it fits with the ERP adopters’ project realities suggesting that an ERP project is implemented as a portfolio of interdependent subprojects [2]. We make the note that breaking down a complex projects into subprojects and managing it as a portfolio has also been recommended by Rand Corporation [23].

In our approach, each subproject is a piece of functionality (or an ERP module) linked to other pieces (or modules). For example, the Sales and Distribution module in a package is tightly linked with the Accounts Receivable and Profits Center Reporting functionality of the Financial Accounting and Controlling modules. Suppose we have a set of interdependent subprojects, the effort-and-deadline-probability model [27] will yield (i) the probability of portfolio’s success with the proposed deadlines for each subproject in this portfolio, and (ii) a set of new deadlines which will result in a required probability of success. The portfolio success is judged by two conditions applied to any two subprojects $a$ and $b$ for which deadline $a$ is earlier...
than deadline. The conditions are that: (i) subproject \( a \) is to be over by deadline \( a \) and (ii) subproject \( a \) and subproject \( b \) are to be over by deadline \( b \). In other words, the conditions require all subprojects planned with a deadline before deadline, to be completed by deadline, rather than just project \( b \). This is the key to the portfolio approach, because uncertainty about completion of project \( b \) incorporated uncertainty from all previous projects.

Suppose the ERP adopter engages in total \( E \) people in the project and let \( d \) be the number of work days it takes from start date to deadline, then the total available resources is \( E \times d \). So, suppose an ERP portfolio \( Y \) is made up by \( n \) subprojects, the success conditions are represented as follows:

\[
\begin{pmatrix}
Y_1 \\
Y_1 + Y_2 \\
\vdots \\
Y_1 + Y_2 + \cdots + Y_n
\end{pmatrix}
\leq
E
\begin{pmatrix}
d_1 \\
d_2 \\
\vdots \\
d_n
\end{pmatrix},
\]

where \( Y_i \) is the estimated effort for subproject \( i \) to succeed. We check if, for any \( j, (j = 1, \ldots, n) \), the sum of \( Y_1, \ldots, Y_j \) is greater of \( E \times d_j \). If this is true, then deadline \( d_j \) has failed. Success probabilities result from simulations in which \( Y_1, \ldots, Y_n \) are generated from a predetermined probability distribution. If we deem \( Y_1, \ldots, Y_n \) is satisfying all conditions, then we say that the portfolio \( Y \) succeeds. The portfolio’s probability of success is equal to the ratio of the number of successes in the set \( Y \) to the number of trials in the simulation.

3.4. Executing the replication of the approach

As in [2], in this study, the application of our approach starts with the position that at the point in time when we produce the estimate, we can posit a trade-off of probability of success for schedule and other resources [27]. Suppose the ERP adopter’s organization (i) has reasonably characterized the context, (ii) has established some operational variance for the size of the ERP portfolio and the complexity of the subprojects included in it, (iii) has some understanding of the ranges of difficulty in obtaining the right configurations for the system, and (iv) has some calibrated way of processing this information, then it is possible for the organization to simulate what might happen when the ERP adopter runs the project.

Our solution approach consists of eight steps which are presented in Fig. 2. Because we designed our approach with the RE stage in mind, we suggest Unadjusted Function Points (FP) [28] be used as a size estimate. This is consistent with the position of the COCOMO II authors [22, page 17]. We chose this measure of functional size because (i) it is a sizing method for ‘paper’ deliverables [39] (and in the bidding stage the requirements are such a deliverable); (ii) is applicable to any ERP package and not to a specific package’s context [3]; (iii) it allows direct measurement of the rate at which requirements change and, thus, may make projects grow larger during implementation; this is possible because both the
original requirements and changed requirements will have function point counts; and (iv) it has been used in industrial studies for more than 20 years.

The FP sizing process is about measuring the functionality by the numbers of ways the system interacts with its users. It starts by counting the number of external inputs, external interface files, external outputs, external queries, and logical internal files. External inputs can be thought of as ERP data-entry screens. External interface files are file-based inputs or outputs. The ERP external outputs are reports and static outputs. External queries are referred to as to external communication into or out of the ERP system. Logical internal files can be thought of the number of tables in the database. We make the note that these definitions are simplified and we add them for the purpose to illustrate the key ideas of FP counting.

Furthermore, to account for uncertainty of the ERP project context, we suggest the COCOMO II model take as inputs the probability distributions of the five COCOMO scale factors and 17 cost drivers, instead of using as inputs single values (as in [2]). This design choice has been recommended by the THAAD Project Office [26] and by the JLP NASA practitioners [26] as well. Deploying the Monte Carlo simulation manes to ascribe a particular distribution type to an input variable in a model, get randomly-selected values, feed them into the COCOMO II model and, then, see how likely each resulting outcome is. In other words, for each uncertain factor, our approach yields possible effort and duration estimation values. In contrast to COCOMO II, our output is the probability distributions of effort and duration and not the most likely effort and duration (which COCOMO II creates).

The probability distributions are, then, fed into the portfolio management method [27]. To run it, we first formulate a condition for success, as in (3), then we bunch projects into portfolios and we obtain the probability of successfully delivering the projects under time constraints as well under effort constraints.

4. The Semi-Replication Case Study Execution

4.1. Application of the method

The solution approach was applied in a setting of a large organization-wide ERP roll-out that included six functional modules of one ERP package (namely SAP) and covered 12 locations of a North American financial services firm. Our data came from 11 SAP projects implemented in the replicated study company. The projects were carried out between May 2003 and September 2007. In this period, the author met representatives of the company’s project teams at a SAP user event in Toronto. (Prior to being a researcher, the author was a practicing SAP process analyst). The ERP implementation process model adopted in the context of the projects was the AcceleratedSAP (ASAP) RE process [29]. It is a project-specific process, engineered and standardized by SAP, and provided to clients by ASAP-certified consulting partners. The ASAP process has been extensively elaborated
in [29]. The practical settings for our 11 projects are presented as follows:

- To manage implementation complexity, each of our projects was broken down into a number of subprojects reflecting the number of components to be configured. For example, the first project had to implement three components and was broken down into six subprojects. This is justified because despite the matter that requirements are collectively presented in a single document, specific requirements pertain to a specific component [2]. Requirements are divided among business process scenarios within the overall project and, thus, facilitate subproject estimates. The total number of our subprojects in which the standard ASAP process was instantiated was 38.

- For each subproject, there was a dedicated RE team. This is a group of individuals who are assigned to a specific subproject, contribute to a solution team, and run the RE cycle for this subproject, and deliver the business requirements document for a specific SAP component.

- Each RE team consisted of one or two SAP consultants who provided in-depth knowledge in both the ASAP implementation process and the SAP components, and a number of business representatives, the so-called process owners. They were department managers and subject matter experts who contributed the necessary line know-how, designed new processes and operational procedures to be supported by the SAP modules, and provided the project with the appropriate authority and resources. All process owners had at least seven years of experience in information systems projects in their departments. Next, we considered our consultants as an even mix of experts and novices. Each expert had at least nine years of configuration and integration experience with a specific SAP functional module. All experts had ASAP RE experience. All consultants had experience in the banking or in the insurance sectors.

- All the teams were supported by three “solution architects” responsible for architecting the solution and consulting on ongoing basis with the teams on SAP reuse, process methods, and tools. The architects served as shared resources to the subproject. The 38 teams worked separately and with relatively little communication among them. This allowed us to initially consider and include 38 subprojects in our case study.

For each of the 38 sub-projects, the following data pieces were collected:

(i) project size data,
(ii) reuse levels,
(iii) start and end dates, and
(iv) scale factor and cost driver ratings.

In the rest of this section we discuss the choices associated with each of these factors:

(1) Functional size was measured in terms of unadjusted IFPUG FP [28]. The organization favored this choice because they had experience in counting FP
according to the IFPUG standard. Reuse levels were formed by using a reuse indicator that included reused requirements as a percentage of total requirements delivered [3]. The exact calculation rules for counting FP and reuse indicators (based on FP) are presented in [3]. As in [2], we make the note that what is unique about the application of the IFPUG standard to ERP is that a set of rules must be established for mapping the data and process counting components of the IFPUG standard onto the components of the data and process requirements models in the ERP business requirements document [3]. These rules say (i) how to identify the boundary of the ERP project to be counted, (ii) how to identify the components to be counted within the boundary of the project, (iii) how to classify these components, and (iv) how to assess the complexity of each component that is counted. Data collection for FP counting was done by using paper forms [3].

(2) The COCOMO II model was locally calibrated by an internally hired effort estimation specialist. The company had already experimented with a variety of estimation techniques, one of which was COCOMO. This alone made it easier to use a calibrated version of COCOMO II for this replication study. The process of calibration itself included applying (i) the linear regression approach [30] and (ii) ERP effort data collected from the first ten ERP projects that were implemented. Given this number of projects, it meant that we met the recommendation for local calibration that at least five projects be used for calibrating the multiplication A and at least 10 projects be used to calibrate the cost drivers and scale factors [30]. The data used to calibrate the effort multipliers A, B, and EM in Eqs. (1) and (2) and the scale factors SF were taken from the company-specific project reporting database. We must note that our choice of which parameters to calibrate is consistent with the recommendations made by the COCOMO II developers [22].

(3) FP-to-line-of-code conversion: As in [2], in this replication we used the average number of lines of code (ALC) in the SAP particular language, ABAP/4, which are required to implement 1 FP. (According to [31], 1 FP equals to 16 ALC.) The estimated code size was computed as follows: Code Size = ALC × UFP.

We must note that we are extremely conscious about that the FP-to-lines-of-code conversion may pose a threat to validity as it might be grossly inaccurate fraught with pitfalls [31, 32]. Jones [31], who created and now updates the conversion tables, recommended by the COCOMO II authors [22], acknowledges that the relationship between FP and lines of code is much more complex than backfiring gives credit. For example, in (6), a linear relationship is assumed, but if the system to be delivered is being developed in a new and very different environment (even when using the same language) that assumption might no longer be justified [38]. In this case study we take the position of that transitioning from one size metric to another is useful, as long as the organization understands “the purposefulness of such an endeavor”. In this case study, the purpose of conversion is to make an approximate estimate without undue effort.
We follow the recommendations by Dekkers and Gunter [32] that the published conversion ratio should be viewed “as no more than indicative” and its use makes sense for rough calculations when large sample size “will even out the discrepancies between types of applications on a portfolio-wide basis”. In our case study, we target an early (and therefore “rough”) estimation and have a relatively large sample of projects treated as a portfolio, which led us to the choice to use the conversion Eq. (6). The issues surrounding the conversion from FP to lines of code in ABAP/4 form part of the discussion on validity in Sec. 6.

(4) Monte Carlo Simulations: As we had the ratings of the cost drivers and scale factors only and no knowledge about the uncertainty of the ratings, we assigned to each factor its distribution type and its parameters of probability distribution (namely center, min and max) based on previously published experiences and recommendations by other authors [26]. For example, this case study used McDonald’s [36] default “high” levels of uncertainty associated to the ratings of the RESL, DATA, ACAP and PCAP cost drivers [22]. (In Sec. 4.2. Table 2 presents descriptions of the COCOMO II cost drivers. For more detailed definitions of the cost drivers, we refer readers to reference [22]). The level of uncertainty determines — in turn, the distribution type to be assigned to each cost driver: normal, triangular, and uniform for low, medium and high uncertainty, respectively. We also opted to use a lognormal distribution for functional size, which was motivated by the observations of Chulani et al. [33]. These researchers investigated the size distribution and indicate that its skew is positive and that log(size) is likely to be a normal distribution.

With this input data (namely, the COCOMO II factors and uncertainty values), we run Monte Carlo simulations which gave us samples of (i) effort, expressed in person-month, and (ii) time, expressed in months. Generally, a Monte Carlo simulation consists of many — often thousands of, trials, each of which is an experiment where we supply numerical values for input variables, evaluate the model to compute numerical values for outcomes of interest, and collect these values for later analysis. In this case study, we used 10,000 trials and generated the samples of effort and time, as presented in Figs. 4 and 5, respectively. In these histograms, the Y-dimension shows the frequency with which a value was observed in the sample of 10,000 trials. The X-dimension shows the value range. Because the average sub-project involved four professionals (two business users, one external consultant and one internal IS team members), we adopted the assumption for E to be 4.

Based on the observation that COCOMO II provides time estimation as in (2), we formulated the following condition for portfolio management in terms of time constraints:

$$\begin{pmatrix} T_1 \\ T_1 + T_2 \\ \vdots \\ T_1 + T_2 + \cdots + T_n \end{pmatrix} \leq \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{pmatrix},$$
4.2. Results

This sections reports on the results with respect to: (i) what we observe when adjusting COCOMO II cost drivers, and (ii) what we learnt from the probability of success of highly-uncertain projects when managing them as a portfolio.

To understand how cost drivers and scale factors make a difference in terms of project success, for each one of them we constructed two portfolios: the first one had this driver/cost factor rated “very high” for all projects and the second
portfolio had it rated “very low” for all projects. Given these two portfolios, we run the portfolio management process. The results of using the two portfolios with respect to each cost driver are presented in Table 1. (For reader’s convenience, after the results in Table 1 below, we provide Table 2 to explain the meanings of the acronyms used for the cost factors in [22]). In Table 1, for each of the 17 COCOMO II cost drivers, we present results in two rows: the one showing the probability of success under effort constraints and under time constraints, when the factor is rated “very low” and the second row showing these probability percentages when the factor is rated “very high”. For example, we found that when selective reuse [3]

<table>
<thead>
<tr>
<th>Cost driver/Scale factor rating</th>
<th>Probability of success</th>
<th>Probability of success</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Under effort constraints</td>
<td>Under time constraints</td>
</tr>
<tr>
<td>DATA: Very low</td>
<td>60.72</td>
<td>62.11</td>
</tr>
<tr>
<td>DATA: Very high</td>
<td>60.91</td>
<td>69.11</td>
</tr>
<tr>
<td>CPLX: Very low</td>
<td>99.01</td>
<td>95.95</td>
</tr>
<tr>
<td>CPLX: Very high</td>
<td>34.73</td>
<td>51.00</td>
</tr>
<tr>
<td>TIME: Nominal</td>
<td>91.00</td>
<td>86.71</td>
</tr>
<tr>
<td>TIME: Very high</td>
<td>93.11</td>
<td>82.19</td>
</tr>
<tr>
<td>STOR: Nominal</td>
<td>81.21</td>
<td>80.01</td>
</tr>
<tr>
<td>STOR: Very high</td>
<td>79.18</td>
<td>77.12</td>
</tr>
<tr>
<td>RUSE: Very low</td>
<td>51.18</td>
<td>63.22</td>
</tr>
<tr>
<td>RUSE: Very high</td>
<td>99.01</td>
<td>92.05</td>
</tr>
<tr>
<td>DOCU: Very low</td>
<td>91.00</td>
<td>84.01</td>
</tr>
<tr>
<td>DOCU: Very high</td>
<td>51.43</td>
<td>48.48</td>
</tr>
<tr>
<td>PVOL: Very low</td>
<td>95.02</td>
<td>93.44</td>
</tr>
<tr>
<td>PVOL: Very high</td>
<td>94.99</td>
<td>93.99</td>
</tr>
<tr>
<td>SCED: Very low</td>
<td>89.32</td>
<td>92.12</td>
</tr>
<tr>
<td>SCED: Very high</td>
<td>56.90</td>
<td>54.994</td>
</tr>
<tr>
<td>RELY: Very low</td>
<td>76.00</td>
<td>76.01</td>
</tr>
<tr>
<td>RELY: Very high</td>
<td>77.15</td>
<td>78.99</td>
</tr>
<tr>
<td>TOOL: Very low</td>
<td>61.94</td>
<td>74.55</td>
</tr>
<tr>
<td>TOOL: Very high</td>
<td>89.34</td>
<td>88.99</td>
</tr>
<tr>
<td>APEX: Very low</td>
<td>43.22</td>
<td>39.90</td>
</tr>
<tr>
<td>APEX: Very high</td>
<td>94.33</td>
<td>87.66</td>
</tr>
<tr>
<td>ACAP: Very low</td>
<td>66.00</td>
<td>62.02</td>
</tr>
<tr>
<td>ACAP: Very high</td>
<td>79.22</td>
<td>91.00</td>
</tr>
<tr>
<td>PCAP: Very low</td>
<td>56.99</td>
<td>62.89</td>
</tr>
<tr>
<td>PCAP: Very high</td>
<td>89.99</td>
<td>94.01</td>
</tr>
<tr>
<td>PLEX: Very low</td>
<td>59.09</td>
<td>63.05</td>
</tr>
<tr>
<td>PLEX: Very high</td>
<td>68.11</td>
<td>69.01</td>
</tr>
<tr>
<td>LTEX: Very low</td>
<td>71.50</td>
<td>56.55</td>
</tr>
<tr>
<td>LTEX: Very high</td>
<td>89.04</td>
<td>90.11</td>
</tr>
<tr>
<td>PCON: Very low</td>
<td>58.88</td>
<td>61.01</td>
</tr>
<tr>
<td>PCON: Very high</td>
<td>83.66</td>
<td>89.22</td>
</tr>
<tr>
<td>SITE: Very low</td>
<td>61.99</td>
<td>70.88</td>
</tr>
<tr>
<td>SITE: Very high</td>
<td>89.90</td>
<td>82.00</td>
</tr>
</tbody>
</table>

*The rating of the TIME cost driver ranges from “nominal” to “extra high”.
†The rating of the STOR cost driver ranges from “nominal” to “extra high”.

Table 1. Analysis of the probability of success for the 17 COCOMO II cost factor under effort constraints and under time constraints.
Table 2. Cost drivers and descriptions as provided in [22].

<table>
<thead>
<tr>
<th>Cost driver acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Database size</td>
</tr>
<tr>
<td>CPLX</td>
<td>Solution complexity</td>
</tr>
<tr>
<td>TIME</td>
<td>Execution time constraint</td>
</tr>
<tr>
<td>STOR</td>
<td>Main storage constraint</td>
</tr>
<tr>
<td>RUSE</td>
<td>Required level of reuse</td>
</tr>
<tr>
<td>DOCU</td>
<td>Documentation match to life-cycle needs</td>
</tr>
<tr>
<td>PVOL</td>
<td>Platform volatility</td>
</tr>
<tr>
<td>SCED</td>
<td>Scheduling factor</td>
</tr>
<tr>
<td>RELY</td>
<td>Required reliability</td>
</tr>
<tr>
<td>TOOL</td>
<td>Use of ERP implementation tools</td>
</tr>
<tr>
<td>APEX</td>
<td>Application experience</td>
</tr>
<tr>
<td>ACAP</td>
<td>Analyst capability</td>
</tr>
<tr>
<td>PCAP</td>
<td>Configuration Specialist/Programmer capability</td>
</tr>
<tr>
<td>PLEX</td>
<td>Platform experience</td>
</tr>
<tr>
<td>LTEX</td>
<td>ERP-specific language and workbench experience</td>
</tr>
<tr>
<td>PCON</td>
<td>Personnel continuity</td>
</tr>
<tr>
<td>SITE</td>
<td>Multisite implementation</td>
</tr>
</tbody>
</table>

was practiced in ERP projects, the probability of success was higher under both time and effort constraints. For the purpose of illustrating this point, we report on the results (see Table 1) yielded when constructing two portfolios of subprojects, namely the first one with the factor of RUSE (meaning reuse) rated as very high for all subprojects and the second one with RUSE rated very low for all subprojects. We make two notes: First, that low level of reuse in an ERP project indicates massive customization of the standard components and that a high level of reuse indicates limited customization [3]. Second, we ruled out the rating “extremely high” as that high levels of reuse are relatively rarely to be observed in a ERP project context [3]. Table 1 suggests that when a project is composed of subprojects all of which have RUSE rated very high, the probability of success is greater under both time and effort constraints.

We observed that 11 out of the 17 factors from the COCOMO II model can be adjusted in a way that maximizes the probability of success. These 11 factors are: CPLX, RUSE, DOCU, ACAP, PCAP, PCON, APEX, LTEX, use TOOL, SITE, SCED. In this case study, our results are inconclusive with respect to the ability to adjust four out of the 17 cost factors. These are: DATA, STOR, TIME, PVOL, RELY and PLEX. We also observed that the scale factors do not seem to let be adjusted in a way that maximizes chance of project success.

Regarding our second group of results, our observations suggest that bundling ERP projects as a portfolio had the advantage over managing projects separately in terms of ability to explicitly and systematically approach uncertainty. We compared the probability of success for projects under effort constraints and for projects under time constraints, respectively (Tables 3 and 4). They indicate that the probabilities of success for projects with high uncertainty ratings are greater when those projects
are managed as a portfolio. In Table 4, the ratio of increase in the probability of success for highly uncertain projects is above seven times (7.37, see the rightmost column).

5. Discussion

5.1. The possible validity threats

The possible threats [25] to validity of this replication study are assessed as follows:

First, we make the note that COCOMO II (as any other model of this family) is a mathematical representation of an “idealized” real-life project relationship only. This is to acknowledge the fact that the 23 cost drivers and cost factors can lead to strong collinearity, heteroscedasticity, and highly variable prediction accuracy [22]. As in [2], in this replication, for the sake of simplicity and pragmatism, we did not investigate the possible ways in which dependencies among cost drivers of different nature might impact our analysis. We, however, deem this an important issue and plan to research it in the future. For the time being, we could only offer an advice to those practitioners and researchers interested in replicating our approach that they should be conscious about this particular under-researched feature of the underlying techniques, being used.

A consequence of using COCOMO represents our second threat to validity that is due to two fundamental difficulties particularly experienced when using algorithmic models (such as COCOMO II):

1. Conversion of FP to lines of code is easy to carry out, but is often inaccurate [38]. In ERP projects, inaccuracies may arise also because the code size is affected by package configuration and design decisions, which have not been made so early in the requirements stage. Specifically, the SAP package uses a commercial database (e.g. Oracle or Sybase) and, thus, the code size might be
small but additional efforts might be needed to overcome the performance limitations of the database product. For example, currently in an SAP solution it is impossible to distinguish between read-only reports and interfaces or utilities that update business data, as the SAP programming language (ABAP) treats both as “objects” of the same type.

(2) The estimates of the context characteristics contributing to E and EM are subjective (see Eq. (1)). We acknowledge that variation may exist from a team member to a team member, depending on the depth and the breadth of the team members’ practice and experience. We, however, believe that the threat to validity in this respect is reduced because we used rules for determining the rates of the cost drivers specifically elaborated for the specific organization and the specific project type. In this case study, the external COCOMO II expert made sure that these rules are consistently applied in the local calibration process of the COCOMO II model.

Third, as in [2], in this study we think that a threat to validity might be posed because of the use of the conversion ratio, ALC, in Eq. (6), which helps to transform the size in FP into lines of code in the COCOMO II model. We are aware of the recommendation in [31, 32] that it is always a good idea to use local data to identify an organization-specific conversion ratio. In the study, determining a company-specific ratio was not a viable option, as we did not have the data and the resources this would need. Instead, we used the most recent table by Jones [31], in which a range of possible values (that is, the min and the max number of lines of code, corresponding to one FP) is assigned to a programming language. The mode value in this range is also provided, which we used in this study. The decision to use the mode value was made after a discussion with three IFPUG members who also were involved in the development of the conversion ratios in the table.

Fourth, when constructing the portfolio, the author based her choice of ‘very low/very high’ ratings on her own experience in implementing ERP. While for some drivers, as reuse, the author did research on what reuse levels are achievable in an ERP project [3], for others the author set up the ratings in a way that — clearly, could be subjective. However, this design choice was the only possible way to go, given the fact that, to the best of our knowledge, there is no published research on the COCOMO II factor ratings which are more common in ERP context.

Fifth, instead of Monte Carlo simulations, the Latin Hypercube technique could be used [24]. Instead of the portfolio management model by Fewster and Mendes [27], the portfolio approach of Chris Verhoef [34] might be good candidates for inclusion. In the future, we are interested in investigating whether different modeling choices sustain our results or limit the validity of our findings to the subset of the analyzed models. We make the note, however, that when it comes down to the practical applicability of the various combinations of techniques by these three types, it all will depend on the availability of data by certain types at ERP adopter’s sites. So, the choice of techniques should represent an acceptable balance between
data-intensiveness and usefulness. As pointed in Sec. 2, some techniques are more
data-intensive than others and, therefore, we expect that their usefulness will vary.
We expect that those techniques which are more data-intensive will be of more
limited use, as very few ERP organizations have the practice of disciplined project
data collection and reporting. We are interested in improving our understanding of
the effect of assumptions commonly made in uncertainty quantification processes
and would like to compare the possible ways of representing uncertainty for ERP
stakeholders. The latter two items form our immediate research agenda.

5.2. What we learnt from this replication study

This section compares our replication study results and the results from our first
study [2] for the purpose of identifying points in which the two studies agree and
disagree. The results of the two studies seem to converge on the following:

(1) Both studies suggest that:

(a) a single cost uncertainty analysis method should not be stipulated for ERP
projects; instead, a combination of approaches can possibly be more valu-
able for project managers,

(b) when managed as a portfolio, highly-uncertain ERP projects have a greater
chance to succeed under time and under effort constraints,

(c) subprojects with high uncertainty ratings would have greater advantages
from portfolio management than projects with low uncertainty ratings
would do, and

(d) it is possible to adjust cost drivers so that a project manager increases
the probability of success for highly uncertain ERP projects that the ERP
adopter might have to implement.

(2) The two case studies shared similar limitations. The validity concerns discussed
in Sec. 5.1 are similar to those reviewed in the first study [2].

We found the following contradiction which we consider important in motivating
our future research plans. Our replication study yielded a different result regarding
the number of factors that can be adjusted so that the chances for project success
are increased. Our first study [2] has indicated 13 out of 17 COCOMO II factors
as adjustable, while this study has found 11 out of the 17 factors to be adjustable.
The two factors on which the two studies disagreed are DATA (which addresses
database size) and PVOL (which addresses platform volatility). We attempted to
explain why DATA and PVOL differ in our two empirical settings. For this purpose,
we searched the Scopus bibliographic database (http://www.scopus.com) for recent
research studies that explored these two factors in either ERP or non-ERP projects.
We, however, could find no theory published by other authors that could be used
in explaining our observations. This finding motivated us to discuss the question of
why DATA and PVOL appear to be outside the scope of project manager’s control,
with the cost analysis expert in the company and one SAP consultant. While for
PVOL, both the cost analyst and the SAP consultant had no explanation to offer, regarding the DATA cost driver, they both suggested that our observation may well be traceable back to the fact that there may be no relationship between the database size and the functional size in an SAP project. (This is an issue already mentioned in the previous section). In their views, a skillful database consultant can reduce the size of the database, for example, by reducing the history/log size, by implementing a specific (the so-called ‘vardecimal’) data storage format, or by offering direct access to legacy data in the SAP Graphic User Interface for business users. In his words, these approaches were “three possible ways to keep the database at manageable size”. We think that this hypothesis may have a merit, because in the SAP literature on database management [35] there are hints that seem to be in line with this consultant’s reasoning. For example [35] claims that total cost of ownership can be decreased if the database is kept “lean” as this would cut backup and restore time, and limit the resource consumption. We acknowledge, however, that more studies are needed to obtain a clearer answer to why the replication study indicates DATA and PVOL as unadjustable factors. Follow-up replication studies would also reveal whether or not those 11 factors that our two studies deem adjustable would indeed remain adjustable in other similar but different settings.

6. Conclusions

This paper addresses the natural tension in ERP projects between the uncertainty of context and the need for an ERP adopter and a consulting company to agree on a price in a contract. Much of the ERP implementation that follows is colored by this tension. Effective cost uncertainty analysis can help ERP project managers understand the context factors in their organizations and critically evaluate the assumptions they and/or their teams, or senior managers make about these factors. Our empirical replication provides evidence strengthening the position that a joint application of multiple techniques can be of greater value to project managers than using one particular technique alone. While it is easier and may be cheaper to use one effort estimation technique for all types of projects in an organization or live with whatever estimates an ERP vendor or a consulting company might come up with, our study suggests that this choice is not practical. Rather, a combination of cost uncertainty methods seems to be a more promising choice. As in our previously published case study [2], we in no way undermine existing approaches but rather to suggest that more multi-technique-based approaches should be investigated for fit, designed and evaluated in the ERP implementation context.

Our results indicated that the joint application of Monte Carlo simulation, a portfolio management method and a parametric empirical model (COCOMO II) adds value particularly in ERP project organizations that struggle with two well-known difficulties inherent to ERP project contexts: (i) lack of ERP-adopter’s specific historical information about the context and (ii) strong bias of ERP consultants in cost estimation. We found that in these contexts, our approach to
be one good alternative to those ERP-adopters that decisively object to use vendors and consultants' estimates as the only reference point in their project investment decision. The results in this replicated study provided evidence regarding the following:

(i) when managed as a portfolio, highly-uncertain ERP projects have a greater chance to succeed under time and under effort constraints,
(ii) subprojects with high uncertainty ratings would have greater advantages from portfolio management than projects with low uncertainty ratings would do, and
(iii) it is possible to adjust cost drivers so that a project manager increases the probability of success for highly uncertain ERP projects that the ERP adopter might have to implement. We have also shown that 11 out of the 17 COCOMO II cost drivers can be adjusted to increase the chances for success.

The replication study has also these implications: First, we draw the attention that to the fact that in the implementation of ERP projects uncertainty could possibly be an asset. For practicing project managers, understanding the adjustable aspects of ERP projects is important so that they can develop appropriate intervention mechanisms (e.g. training and communication) that can lead to successful ERP roll-out of projects managed together as a portfolio.

The findings of the study need to be interpreted cautiously because it has some limitations, and also because we observed a contradiction with respect to our previously published study [2]. We therefore are involved in an on-going collaboration with ERP adopting companies, so that we carry out more case studies and possibly build a body of evidence about (1) the adjustability of the cost drivers, and (2) the advantages and the disadvantages of the possible ways to deal with the uncertainty of context in ERP projects.

Acknowledgments
The author thanks the company for providing the environment in which the author could carry out the case study. The research in the paper was also partially funded by the CTIT project COSMOS.

References
35. M. Höding, A. Faustmann, G. Klein and R. Zimmermann, SAP Database Administration with Oracle (SAP Press, 2010).