ERP Implementation Effort Estimation Using Data Envelopment Analysis

Stefan Koch
Institute for Information Business, Vienna University of Economics and BA
stefan.koch@wu-wien.ac.at

Abstract

ERP implementation projects have been encountered as very risky and complex, and frequently exhibit time and cost overruns. Therefore both a comparison of their efficiency, and especially an estimation of effort and costs associated with new projects are of high interest. Since ERP implementation differs from classical software development projects, as the main focus lies on adapting and customizing a complex system, and even changing the organisation, traditional models like COCOMO can not easily be applied. The non-parametric DEA on the other hand has been developed to compare the efficiency of units with multiple inputs and outputs with different measurement scales. In this paper, we will describe the application of DEA using a data set of 39 projects, drawing on the results of applying DEA for effort estimation of future projects.

Keywords: Effort estimation, Enterprise Resource Planning, Data Envelopment Analysis

1. Introduction

In the last years an increasing number of enterprise resource planning (ERP) systems like SAP R/3, BaaN or Oracle Applications have been implemented. This implementation of the selected solution is risky and complex for individual companies, since on the one hand the software and especially the necessary manpower is relatively expensive and on the other hand in this process possibly the internal business processes need to be adapted as well. This results in the desire to compare the efficiency of implementation projects in order to identify best practices, and especially to estimate effort and costs associated with new projects. These informations would be beneficial for project managers involved in such projects, but also for consulting companies and ERP vendors. However, this proves to be difficult, since ERP implementation differs from the development and implementation of individual software solutions in some important points. It actually changes the enterprise and the business processes in many cases, and a majority of the necessary expenditure flows not into the production of new program code, but into adjustment work (so-called customizing). Thus traditional metrics for the size of software products as for instance function points [1] or lines-of-code [2] are not easily applicable which leads in further consequence to problems in computing productivity and efficiency. The data envelopment analysis (DEA) [3] on the other hand allows for such a comparison in particular even if several input and output factors are present. Furthermore, the results of a DEA can also be used to arrive at effort and cost estimates for future projects. In this paper, we will describe the application of DEA in this context of ERP implementation projects.

2. ERP Implementation Effort Estimation

ERP systems are application programs which are used for the enactment and support of all business functions as well as for the administration of all information necessary for these tasks. ERP systems feature a single database, in which all information from the enterprise is stored and administered. Therefore all data must be stored only once, and is available to all modules (like financials, cost accounting, logistics or human resource management), which results in a standardized and faster data flow. In addition, redundancies are prevented. The main characteristics of ERP are further process orientation by execution of function chains, implementation of all business processes, the internal integration of all parts of the software, the applicability in many industries as well as the adaptability to national characteristics, a uniform GUI and the possibility of adapting the software within pre-defined limits to the requirements of a given enterprise.

The implementation of an ERP system in an enterprise represents a special topic of high concern. This area entails among other things the customizing, the adjustment of the enterprise-neutrally delivered standard software to the specific business requirements of an enterprise. In [4], a fine-grained understanding and classification of these tailoring activities is developed, resulting in a typology of 9 different types. Due to the high integration of the underlying software and possible changes to the organisation resulting from implementation, the resulting projects are characterised by high complexity (and most often also high costs and long running times) [5,6,7,8,9]. In some cases such projects can lead also to major problems up to the bankruptcy of the enterprise [10,11]. Therefore the literature (and also suppliers of ERP systems as for example SAP with ASAP - Accelerated SAP [12]) offers diverse suggestions and guidelines for performing and managing such projects [6,8,9], critical success factors [13,14] as well as for the interplay of adjustment of the organization in contrast to the software by customizing or even additional programming in the product language [11]. Depending on the selected process model, an ERP implementation project usually entails an analysis phase of the business processes, information flows and data model with the result of an current state analysis, followed by a design phase with the development of a target concept, a selection phase of the most suitable standard software system [15] and finally an implementation phase with the various means of adaption [4].

These challenges and the data on cost and schedule overruns result in the need for assessing and comparing the efficiency of ERP implementation projects, and in a second step for estimating costs and effort for future projects. For both tasks, the relevant inputs and outputs of such a project need to be defined. While in the simple (and idealised) case of software development, input can be given by the developer effort, e.g. in person-months, and the output characterised by some software product metric like function points or lines-of-code, the situation in this case is more complex. While inputs are not much more difficult to capture, outputs seem problematic. Due to the fact that often no new code is produced, but customizing within the software takes place, together with possible changes to the business processes and the organisation itself, a set of output metrics needs to be defined in order to completely assess the output produced. Traditional software product metrics can, for the case of programming effort, have a place in it. For the resulting case of several outputs (and possibly several inputs), in addition most often measured on completely different scales, traditional approaches for computing the productivity and efficiency can not be used. DEA on the other hand has been developed to deal with this problem, and also offers some additional benefits. Also Kitchenham and Mendes [16] agree that productivity measures need to be based on multiple size measures.

Besides comparing productivity, estimating the effort for future projects has a high priority. Naturally, it follows from the reasons given above that estimation approaches developed for new software development, like most notably COCOMO [17] or others, relying on standard software product metrics, are not applicable. Literature currently does not yield any approaches specifically targeted at ERP implementation projects. Given a large and diverse enough database, machine-learning approaches like for example regression trees [18] or analogy-based estimation [19] might prove helpful. Myrtveit and Stensrud [20] have detailed the usage of analogy-based estimation and multiple regression in this context, but could not prove the first one to generally outperform the latter. In this paper, we will describe how DEA can be applied to this problem.

3. Data Envelopment Analysis

The principle of the border production function was introduced by Farell [21] for measuring the technical efficiency and enhanced by Charnes, Cooper and Rhodes into the first DEA model [22,3]. The object of analysis is generally the Decision Making Unit (DMU [3]). This term includes relatively flexibly each unit which is responsible for the transformation of inputs into outputs, for example hospitals, supermarkets, schools or bank branches.

The basic principle of DEA can be understood as a generalization of the normal efficiency evaluation by means of the relationship from an output to an input into the general case of a multi-output, multi-input system without a any given conversion rates or same units for all factors. In contrast to other approaches, which require the parametric specification of a production function, DEA measures production behavior directly and uses this data for evaluation. The DEA derives a production function from mean relations between inputs and outputs (whereby it is only assumed that the relation is monotonous and concave), by determining the outside cover of all production relations, and identifies "best practicing" DMUs on this production border. A DMU is understood as being efficient if none of the outputs can be increased, without either or several of the inputs increasing or other outputs being reduced, as well as vice versa.

For each DMU an individual weighting procedure is used over all inputs and outputs. These form a weighted positive linear combination, whereby the weights are specified in such a way that they maximize the production relationship of the examined unit, in order to let these become as efficient as possible. The efficiency of an examined unit is limited with 1. That means that no a-priori weightings are made by the user, and that the weights between the DMUs can be different. For each evaluation object the DEA supplies a solution vector of weighting factors and a DEA efficiency score with a value of 1 denoting DEA-efficiency. This means that within the selected model variant no weighting vector could be found which would have led to a higher efficiency value. For each inefficient DMU the DEA returns a set of efficient DMUs, which exhibit a similar input/output structure and lie on the production border near to the inefficient DMU (reference set or DEA benchmark) to give an idea in which direction an increase in efficiency is possible.

The first model of the DEA was introduced by Charnes, Cooper and Rhodes [3] and is designated with their initial letters as CCR model. The different models of the DEA can be divided on the basis of two criteria: the orientation of the model and the underlying assumption regarding the returns to scale of the production process. With input-oriented models the reduction of the input vector maximally possible is determined, whereas with output-oriented models the maximally possible proportional increase of the output vector is determined. With constant returns to scale size-induced productivity differences are considered into the efficiency evaluation, with variable returns to scale the differences are neutralized by the model. The most common example of a model with variable returns to scale is a modification of the CCR model by Banker, Charnes and Cooper, the BCC model [23]. This includes an additional measuring variable in the fundamental equation to capture rising, falling or constant returns to scale.

In the area of software development, DEA was so far only rarely applied. Banker and Kemerer use this approach in order to prove the existence of both rising and falling returns to scale [24]. First arise in small, the others in larger software development projects. Banker and Slaughter use the DEA in the area of maintenance and enhancement projects [25]. It can be proven that rising returns to scale are present, which would have made a cost reduction of around 36 per cent possible when utilized. Mayrhauser et al. [26] also report applying DEA on a data set consisting of 46 software projects from the NASA-SEL database to analyze objective variables and their impact on efficiency. An investigation of ERP projects was done by Myrtveit and Stensrud [27]. They used 30 SAP R/3-projects of a consulting firm for the application of the DEA. As inputs the costs, as outputs the number of the users, EDI and conversions are used. The authors point out large differences in productivity between projects, as well as the presence of variable returns to scale. Kitchenham [28] gives an in-depth discussion on the application of DEA in software development. She argues that several assumptions of DEA are violated in most software project data sets. In addition, it is difficult and not agreed which variables are to be included, and as a further problem a project might not be considered a DMU because of its limited autonomy. Also measurement errors or noise are problematic, which could be handled better with a stochastic frontier. In our data set, measurement errors are not impossible, but for many variables highly unlikely, as the number of users or modules of an ERP software is easy to assess. In addition, we perform a sensitivity analysis. For inclusion of variables, we use correlation coefficients to chose those exhibiting a relationship with effort. Kitchenham and Mendes [16] employ stepwise regression for this task in constructing an adjusted size measure. ERP implementation projects as considered here most often enjoy quite large autonomy on the parameters given by Kitchenham, i.e. relation of contractors to own employees, investment in technology and training, but also process model and others. Therefore they might be considered DMUs to a larger degree than traditional software development projects.

4. Data Set

4.1. Description

For the empirical validation of DEA for effort estimation, data on a number of ERP implementation projects was necessary. Therefore a questionnaire was developed, which was after a pretest sent to Austrian companies which had already introduced an ERP system. Altogether 300 enterprises of different industries were addressed, using customer lists of several different suppliers including SAP and BaaN which had already been used in a preceding study on ERP system selection processes [15]. Out of 43 questionnaires returned, 39 could be included into the analysis.

For each enterprise, data characterising the organisation itself, and the implementation project were asked. In the first group, industry and number of employees were collected. For industries within the data set, a considerable number of enterprises is present only from trade (8) and production (14). Results regarding the business size are given in Table 1.

Table 1. Size of enterprises in data set (N=39).

Number of Employees	Frequency	Percentage
0-250	15	38.5
251-500	8	20.5
501-1000	5	12.8
1001-2000	3	7.7
More than 2000	8	20.5

Next it was explored which modules (or parts) of the integrated software system were implemented. The area of finance and accounting was most strongly represented: 87% of the regarded enterprises introduced the financial module , controlling 79%. Furthermore 64% implemented distribution, 44% manufacturing, 28% project, 13% transport, 15% service and 51% others. In the mean, an enterprise implemented 3.82 different modules (standard deviation 1.48). In order to more exactly specify the ERP solution resulting from the implementation project, four additional variables were explored (see Table 2 for descriptive statistics). These are based the number of employees in the enterprise who use the standard software after final implementation, the extent of the modifications to the software in lines-of-code (which according to the typology developed in [4] constitutes a form of ERP tailoring), the number of interfaces to other software systems (another form of ERP tailoring according to [4]), and the number of locations in which the solution was introduced.

Table 2. Characteristics of implemented ERP solutions (N=39).

	Max	Mean	St.Dev.
Number of Modules	7.00	3.82	1.48
Users	1,500.00	217.46	364.23
Modifications (in LOC)	5,000.00	227.95	841.70
Interfaces	100.00	12.10	20.39
Locations	62.00	4.26	9.98

The last group of variables are concerned with costs and duration of the ERP implementation project (see Table 3). The effort is measured in person-years, the costs are both taken in total and several sub-components. These include the cost of the software, the acquisition of any necessary hardware, as well as any costs associated with employing external consultants (whereby only 3 enterprises, or 8% did not use any consultants) which show up to be the largest position.

Table 3. Effort and costs of ERP implementation projects (N=39).

	Max	Mean	St.Dev.
Duration of Implementation (in Weeks)	156.00	43.05	29.45
Effort (in Person-Years)	1200.00	65.68	245.58
Total Costs (in EUR)	14,535K	1,477K	2,718K
Costs for Software (in EUR)	5,814K	361K	977K
Costs for Hardware (in EUR)	4,360K	267K	743K
Costs for Consultants (in EUR)	5,814K	519K	997K

As a first preliminary analysis preceding DEA, correlations between the described variables are evaluated. All variables used for characterising effort and costs are positively correlated, with coefficients of 0.67 between total costs and effort up to 0.95 (significance level in all cases 0.01). The duration of the project is however only correlated with the total costs at 0.44 ($p<0.01$), but not with the effort in person-years. Regarding the characteristics of the project, the number of the users and interfaces show an influence on total costs, duration and effort, the extent of modifications and number of locations however no statistically provable one. In particular a high correlation to the costs exists, for the number of the users at 0.91 and interfaces at 0.81 ($p<0.01$). For the effort in person-years the correlation-coefficients are 0.5 (for users) and 0.54 (for interfaces), for the duration 0.52 respectively 0.33. Also the number of modules implemented has a significant positive correlation to the total costs (coefficient of 0.42 with $p<0.01$). If a non-parametric correlation coefficient like Kendall's tau is applied due to the skewness of the data, the results do not change, with the single exception of a significant correlation between duration and total costs.

4.2. Application of DEA

As a first step, the available projects were subjected to a DEA, in order to compare their productivity and efficiency. For this there are different software products available, some of which are freeware. In this case, the program accompanying the book by Cooper, Seiford and Tone [29] was used, which can compute different DEA models, input or output-oriented, as well as with constant or variable returns to scale.

The first choices to be taken concern the definition of input and output factors, as well as the model to be applied. Banker and Kemerer have demonstrated the existence of both increasing and decreasing returns to scale in software projects [24], also Myrtveit and Stensrud recommend to use a model with variable returns to scale [27]. Using a data set of maintenance and extension projects Banker and Slaughter have found increasing returns to scale [25]. Kitchenham [28] gives an overview of research result and reasons for differences on economies and diseconomies of scale in software development. In addition, the size of an ERP implementation project can under most circumstances not be chosen by the decision makers, which would also point to using a variable return to scale model. In order to confirm this, efficiency scores for all projects are computed using both different assumption and the results are compared. Regarding the orientation of the model an input orientation is selected, since the ERP system necessary for an organization, hence the output, is determined by the requirements, and this is to be reached with minimal input factors. Thus BCC-I is applied with variable returns to scale, and CCR-I for constant returns to scale.

Regarding the definition which factors are to be used as inputs and outputs, it is to be considered that with an increase in the number of factors more DMUs are estimated to be efficient, in particular if the database is small in relation. In the available case the total costs offer themselves as inputs as well as the effort in person-years. As outputs the duration of the introduction, the number of users, the number of interfaces and the implemented modules are used. The duration of the project is for many organisations of high interest and further is characteristic for how efficiently and fast the effort could be applied. The extent of modifications and the number of locations do not exhibit significant influence on costs and effort, and therefore are not included.

The results for the application of the selected models are summarized in Table 4. It can clearly be seen that with application of the model BCC-I with variable returns to scale the efficiency scores are higher both in average and median as well as minimum. Likewise the number of DEA efficient projects rises strongly from 6 to 22, which corresponds to 56% of all projects. This increase in the efficiency evaluation can be explained by the fact that size-induced productivity differences are balanced by the model, and thus some small projects are no longer dominant. But even in the BCC-I model an average saving potential of nearly 25% is shown. Compared with the results of Myrtveit and Stensrud on higher average efficiency is found, since they found a value of 0.56 with application of a model with variable returns to scale [27].

Table 4. Results from applying CCR-I and BCC-I models (N=39).

CCR-I	
Average Efficiency Score	0.4472
Std. Dev.	0.3219
Median	0.3726
Minimum	0.0039
Number of Efficient Projects	6
BCC-I	
Average Efficiency Score	0.7552
Std. Dev.	0.3239
Median	1
Minimum	0.0267
Number of Efficient Projects	22

For testing the stability of these results, a sensitivity analysis has been performed by deleting those projects that most often show up in the reference set

[27,30], i.e. those possibly constituting outliers or measurement errors. When removing those 7 projects, the average efficiency score in the BCC-I model changes only between 0.7512 and 0.7682, showing the stability of the results. Regarding whether variable or constant returns to scale are present, a statistical test on equality of the distributions of the efficiency scores from both models is applied. Since the efficiency scores are not normal distributed, a non-parametric test for dependent samples (e.g. Wilcoxon rank) can be used. Another possibility consists of testing the average value of the differences, which are normal distributed according to a Kolmogorov-Smirnov test, by means of a t-test on being 0. In both cases the result at a level of significance of 0.01 is that a significant difference between the efficiency scores is showing up. As a BCC-I model could also apply constant returns to scale if these would best fit the data, this shows that variable returns to scale are present and the BCC-I model is to be used. This result is identical with those of Myrtveit and Stensrud [27], as well as results from software development [24,25].

5. Effort Estimation with DEA

5.1. Approach

We propose to use DEA in effort and cost estimation in a similar way to the application for productivity comparison, in which the input and output factors are known, and the efficiency is to be computed. For effort estimation, the outputs are determined, and a certain efficiency level is set. Determining the outputs will, depending on the different measures included, be possible in advance. In our example, the number of users, modules and interfaces will be relatively clearly defined at the beginning. The duration will be set by management, or depend on other restrictions. Of higher interest is the efficiency level. There are several possibilities for setting a value: One would be to optimistically hope for DEA-efficiency to be reached, thus implying a project on the current production frontier, another would be to set a lower value (e.g. 0.5) in order to be on the safe side. The last possible value would be an orientation on the current status, and use the mean (or median) efficiency achieved by all other projects, or of constructing a confidence interval using mean and standard deviation. Exploring several of these alternatives shows one of the advantages of this method, in that several scenarios can easily be computed. In addition, while the computation itself is complex and not easily communicated, the notion of efficiency score and associated scenarios with resulting estimations will be easily comprehended by decision-makers.

If all of these values have been determined, the resulting set of equations is solved for both the weights associated with the input and output factors, given the usual condition that the same set of weights may not result in an efficiency value of more than one for any project, and in addition setting the efficiency score for the current project at the determined value, and also the input factors themselves. It should be noted that a single solution will most certainly not be possible given multiple input factors, but a certain trade-off between the input factors is resulting from the model. Nevertheless, exactly exploring this trade-off, e.g. between costs for external consultants and in-house costs, might be of interest.

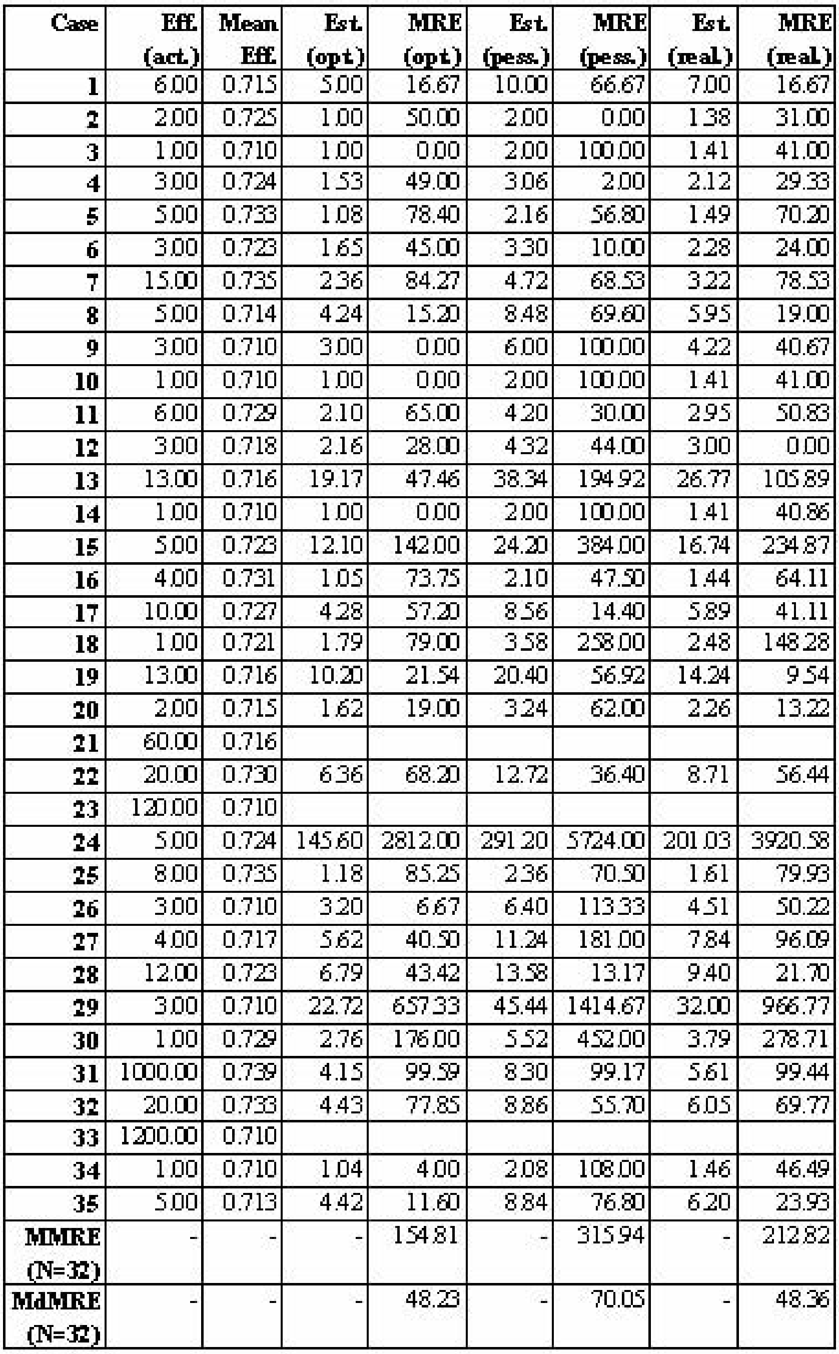
5.2. Validation

For validation, we use the data set presented above. In order to decrease complexity, and also to make comparisons to other approaches easier, we limit the analysis to the single-input case, thus resulting in a single solution for a set efficiency level. This allows to calculate standard measures for estimation quality. To this end, we compute a new BCC-I model in a first step, this time using only effort as input, while keeping all output factors. The decreases the sample size to 35, as 4 projects did not contain effort data, only total costs. The results are not very different from the model described above, and give an average efficiency of 0.718 (median 0.885) and 16 efficient projects.

Then we proceeded to estimate the effort for each single project, using all other projects as data set. This jack-knifing technique offers two distinct advantages compared to partitioning the data set into a training set and a holdout sample, in that more cases can be estimated thus giving a better indication of predictive quality, and that each case can be estimated using a larger sample, thus increasing quality. As a first step, each case was eliminated from the total set one after the other, and the BCC-I model as above was computed using the remaining projects in order to determine the mean efficiency within the respective group. In Table 5, the respective value for each case can be seen, with the very low bandwidth (between 0.710 and 0.739) again underlining the stability of the model results. Then, the effort for each single project was estimated as described in the last section. The output factors were taken from the data, as they are supposed to be easily determined by the organisation. We then used three different scenarios for the efficiency to be achieved by the project: An optimistic scenario with the efficiency on the production border, i.e. DEA-efficiency of 1, a pessimistic scenario with efficiency of 0.5 and a realistic scenario assuming the mean efficiency within the set. For all three possibilities, estimated effort was computed, and compared to the real value. Table 5 gives the relevant data, including magnitude of relative error (MRE) for each case under each scenario. For three projects an estimation was not possible, as they lie on the production frontier due to their high output in a factor, no matter what the input. These cases are discarded from further analysis.

As can be seen, the mean magnitude of relative error (MMRE) is between 154.81 and 315.94, with the optimistic scenario performing best, the median magnitude of relative error (MdMRE) is 48.36 in the realistic and 48.23 respective 70.05 in the optimistic and pessimistic case. Using a Wilcoxon rank test, it can be shown that the differences in MRE between the others and the pessimistic scenario are statistically significant (at level 0.01), while the realistic and optimistic ones are not significantly different. A different quality indicator, pred(0.2), is for the optimistic scenario 31.25%, for the pessimistic and the realistic 15.63%. Using this quality measure, again the optimistic scenario performs best. This performance results from the data set studied, which sports a high number of efficient projects and is not normal distributed. Therefore using the mean efficiency score did not prove to be the best strategy in the realistic scenario, the median, which is nearer to 1 would have performed better. It should also be noted that the pessimistic scenario underestimates the true effort in 9 cases, and the optimistic scenario overestimates in also 9 cases. The first fact shows that 0.5 might be set too high an efficiency score for a worst-case scenario, while the second fact highlights that these projects would have expanded the production frontier as determined using the other projects in the data set.

Table 5. Results from effort estimation.



Case	Eff. (act.)	Mean Eff.	Est. (opt)	MRE (opt)	Est. (pess.)	MRE (pess.)	Est. (real.)	MRE (real.)
1	6.00	0.715	5.00	16.67	10.00	66.67	7.00	16.67
2	2.00	0.725	1.00	50.00	2.00	0.00	1.38	31.00
3	1.00	0.710	1.00	0.00	2.00	100.00	1.41	41.00
4	3.00	0.724	1.53	49.00	3.06	2.00	2.12	29.33
5	5.00	0.733	1.08	78.40	2.16	56.80	1.49	70.20
6	3.00	0.723	1.65	45.00	3.30	10.00	2.28	24.00
7	15.00	0.735	2.36	84.27	4.72	68.53	3.22	78.53
8	5.00	0.714	4.24	15.20	8.48	69.60	5.95	19.00
9	3.00	0.710	3.00	0.00	6.00	100.00	4.22	40.67
10	1.00	0.710	1.00	0.00	2.00	100.00	1.41	41.00
11	6.00	0.729	2.10	65.00	4.20	30.00	2.95	50.83
12	3.00	0.718	2.16	28.00	4.32	44.00	3.00	0.00
13	13.00	0.716	19.17	47.46	38.34	194.92	26.77	105.89
14	1.00	0.710	1.00	0.00	2.00	100.00	1.41	40.86
15	5.00	0.723	12.10	142.00	24.20	384.00	16.74	234.87
16	4.00	0.731	1.05	73.75	2.10	47.50	1.44	64.11
17	10.00	0.727	4.28	57.20	8.56	14.40	5.89	41.11
18	1.00	0.721	1.79	79.00	3.58	258.00	2.48	148.28
19	13.00	0.716	10.20	21.54	20.40	56.92	14.24	9.54
20	2.00	0.715	1.62	19.00	3.24	62.00	2.26	13.22
21	60.00	0.716						
22	20.00	0.730	6.36	68.20	12.72	36.40	8.71	56.44
23	120.00	0.710						
24	5.00	0.724	145.60	2812.00	291.20	5724.00	201.03	3920.58
25	8.00	0.735	1.18	85.25	2.36	70.50	1.61	79.93
26	3.00	0.710	3.20	6.67	6.40	113.33	4.51	50.22
27	4.00	0.717	5.62	40.50	11.24	181.00	7.84	96.09
28	12.00	0.723	6.79	43.42	13.58	13.17	9.40	21.70
29	3.00	0.710	22.72	657.33	45.44	1414.67	32.00	966.77
30	1.00	0.729	2.76	176.00	5.52	452.00	3.79	278.71
31	1000.00	0.739	4.15	99.59	8.30	99.17	5.61	99.44
32	20.00	0.733	4.43	77.85	8.86	55.70	6.05	69.77
33	1200.00	0.710						
34	1.00	0.710	1.04	4.00	2.08	108.00	1.46	46.49
35	5.00	0.713	4.42	11.60	8.84	76.80	6.20	23.93
MMRE (N=32)	-	-	-	154.81	-	315.94	-	212.82
MdMRE (N=32)	-	-	-	48.23	-	70.05	-	48.36

Compared to the results of Myrtveit and Stensrud [20], who have used analogy-based and multiple regression for estimating their dataset of ERP projects, the results are comparable but do not point to DEA as a generally better performing estimation methodology. They report an MMRE of 126 for practitioners using a multiple regression model (MdMRE of 43), of 136 for practitioners using the analogy tool (MdMRE of 51), and for the tools alone of 127 (MdMRE 35) for multiple regression respective 154 (MdMRE 52) for the analogy tool. These numbers are quite near to the results of the optimistic scenario.

6. Conclusion

In this article, we have argued to apply DEA for the problem of ERP implementation project effort and cost estimation. This type of projects differs from traditional software development in a significant way, making it an ideal case for applying DEA, a non-parametrical method which allows for multi-input and multi-output relations, with different units of measurement. As we have shown using a data set of 39 projects from a questionnaire, the efficiency of these can indeed be compared. For effort and cost estimation, DEA has been applied for the first time. From a first preliminary validation we can draw the following insights: DEA can achieve estimation results comparable in predicitve quality to other approaches like analogy-based methods, if properly set up. Using a measure of central tendency suitable to the data set will yield best predictive results. One main advantage is that different scenarios can easily be set up and explored. Another point in comparison with other approaches is that multi-input situations can be analysed, although then a single solution will in most cases not be found, which on the other hand leaves possibilities for exploring trade-offs between different input factors. Further restrictions of this method lie in the need to set an efficiency level, which leads to multiple possibly solutions, the chance that a new project might expand the production frontier and thus be more efficient than those previously encountered which is difficult to adequately reflect, and the need for a diverse data set of projects. Computation of DEA models is in general mathematically complex, but there are numerous tools available to this end. These are not tailored yet for effort estimation purposes, so some effort has still to be expended, but in league with other approaches like analogy-based techniques. Future research will be necessary to develop a set of metrics and measures for quantifying ERP implementations, for example using different ERP tailoring types [4].

7. References

1. Albrecht, A.J., and J.E. Gaffney, “Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation“, IEEE Transactions on Software Engineering, 9(6), pp. 639-648, 1983.

2. Park, R.E., “Software Size Measurement: A Framework for Counting Source Statements”. Technical Report CMU/SEI-92-TR-20, Software Engineering Institute, Carnegie Mellon University 1992.
3. Charnes, A., W. Cooper, and E. Rhodes, “Measuring the Efficiency of Decision Making Units”, European Journal of Operational Research, 2, pp. 429-444, 1978.
4. Brehm, L., A. Heinzl, and M.L. Markus, “Tailoring ERP Systems: A Spectrum of Choices and their Implications“, Proceeding of the 34th Hawaii International Conference on System Sciences, Hawaii, 2001.
5. Davenport, T.H., “Putting the enterprise in the enterprise system”, Harvard Business Review, July-August, pp. 121-131, 1998.
6. Kirchmer, M., Business Process Oriented Implementation of Standard Software, Springer-Verlag, Berlin, 1998.
7. Scheer, A.-W., ARIS – Business Process Frameworks, 3rd edition, Springer-Verlag, Berlin, 2000.
8. Wallace, T.F., and M.H. Kremzar, ERP: Making It Happen: The Implementers' Guide to Success with Enterprise Resource Planning, Wiley, New York, 2001.
9. Welti, N., Successful SAP R/3 Implementation: Practical Management of ERP Projects, Addison-Wesley, Reading, MA, 1999.
10. Scott, J.E., “The FoxMeyer Drugs' bankruptcy: Was it a failure of ERP?”, Proceedings of the Fifth Americas Conference on Information Systems (AMCIS 1999), Milwaukee, pp. 223-225, 1999.
11. Scott, J.E., and I. Vessey “Managing Risks in Enterprise Systems Implementations”, Communications of the ACM, 45(4), pp. 74-81, 2002.
12. Brand, H., SAP R/3 Implementation With ASAP: The Official SAP Guide, Sybex, Alameda, CA, 1999.
13. Holland, C.P., and B. Light, “A Critical Success Factors Model for ERP Implementation”, IEEE Software, 16(3), pp. 30-36, 1999.
14. Bingi, P., M.K. Sharma, and J.K. Godla, “Critical Issues affecting an ERP implementation”, Information Systems Management, 16(3), pp. 7-14, 1999.
15. Bernroider, E., and S. Koch, “ERP Selection Process in Midsize and Large Organizations”, Business Process Management Journal, 7(3), pp. 251-257, 2001.
16. Kitchenham, B. and E. Mendes, “Software Productivity Measurement Using Multiple Size Measures”, IEEE Transactions on Software Engineering, 30(12), pp. 1023-1035, 2004.
17. Boehm, B.W., Software Engineering Economics, Prentice Hall, Englewood Cliffs, NJ, 1981.
18. Selby, R.W., and A.A. Porter, “Learning from Examples: Generation and Evaluation of Decision Trees for Software Resource Analysis”, IEEE Transactions on Software Engineering, 14(12), pp. 1743-1756, 1988.
19. Shepperd, M., C. Schofield, and B. Kitchenham, “Effort Estimation Using Analogy”, Proceedings of the 18th International Conference on Software Engineering (ICSE 1996), pp. 170-178, Berlin, 1996.
20. Myrtveit, I., and E. Stensrud, “A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models”, IEEE Transactions on Software Engineering, 25(4), pp. 510-525, 1999.
21. Farell, M.J., “The Measurement of Productive Efficiency”, Journal of the Royal Statistical Society, Series A 120(3), pp. 250-290, 1957.

22. Charnes, A., W. Cooper, and E. Rhodes, “A Data Envelopment Analysis Approch to Evaluation of the Program Follow Through Experiments in U.S. Public School Education”. Management Science Research Report No. 432, Carnegie-Mellon University, Pittsburgh, PA, 1978.
23. Banker, R.D., A. Charnes, and W. Cooper, “Some Models for Estimating Technical and Scale Inefficiencies in Data Envelopment Analysis”, Management Science, 30, pp. 1078-1092, 1984.
24. Banker, R.D., and C. Kemerer, ”Scale Economies in New Software Development”, IEEE Transactions on Software Engineering, 15 (10), pp. 416-429, 1989.
25. Banker, R.D., and S.A. Slaughter, “A Field Study of Scale Economies in Software Maintenance”, Management Science, 43(12), pp. 1709-1725, 1997.
26. Mayrhauser, A., C. Wohlin, and M. Ohlsson, “Assessing and Understanding Efficiency and Success of Software Production”, Empirical Software Engineering, 5(2), pp. 125-154, 2000.
27. Myrtveit, I., and E. Stensrud, “Benchmarking COTS Projects Using Data Envelopment Analysis”, Proceedings of 6th International Software-Metrics-Symposium, Boca-Raton, pp. 269-278, 1999.
28. Kitchenham, B., “The question of scale economies in software - why cannot researchers agree?”, Information & Software Technology, 44(1), pp. 13-24, 2002.
29. Cooper, W., L. Seiford, and K. Tone, Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software, Kluwer Academic Publishers, Boston, MA, 2000.
30. Wilson, P.W., “Detecting Influential Observations in Data Envelopment Analysis”, Journal of Productivity, 6, pp. 27-46, 1995.