Benchmarking COTS Projects Using Data Envelopment Analysis

Ingunn Myrtveit
The Norwegian School of Management
P.O. Box 580, N-1301 Sandvika, Norway
+47-6757 0576
ingunn.myrtveit@bi.no

Erik Stensrud
Ernst & Young Management Consulting
P.O.Box 6834 St. Olavs plass, N-0130 Oslo, Norway
+47-9228 1903
erik.stensrud@ey.no

Abstract

In Ernst & Young and Andersen Consulting, two of the "big five", there is a continuous search for better methods to measure and compare project performance of multi-dimensional COTS software projects. We propose using Data Envelopment Analysis (DEA) with a Variable Returns to Scale (VRS) model. First, we discuss and illustrate this method by analyzing Albrecht-Gaffney's two-dimensional dataset. Next, we review previous empirical studies using DEA showing that several studies have used DEA where simpler methods could have been used. Finally, we apply DEA to a multi-dimensional dataset of 30 industrial COTS software projects extracted from a benchmarking database in Andersen Consulting. Our main conclusion is that DEA is an applicable method, albeit not without shortcomings, for comparing the productivity of COTS software projects, and that it, therefore, merits further research. However, for two-dimensional datasets this method is unnecessary complex, and there exists other, simpler alternatives. Also, the results support our assumption of increasing as well as decreasing returns to scale for this dataset. Thus, the VRS model provides more reasonable and fair comparisons of project performance than a Constant Returns to Scale (CRS) model. Finally, this study suggests that DEA used together with methods for hypothesis testing may be a useful technique for assessing the effect of alleged process improvements.

Keywords

Benchmarking, multi-dimensional efficiency measurements, data envelopment analysis, software development, COTS, software size metrics, economies of scale, sensitivity analysis.

1. Introduction

In Ernst & Young and Andersen Consulting, two of the "big five", there is a continuous search for better methods to benchmark the productivity of COTS software projects. Benchmarking in this context means to measure the project productivity against some established performance standard, or alternatively, against a best practice frontier. Productivity is generally defined as the output over input ratio. In general, the more output per unit input, the more productive the project is. The usual input is effort, and the output is the amount of product delivered. The amount of product must be represented by some size measure.

For traditional custom development projects the most common size metrics are SLOC and Function Points. For COTS software projects there exists no international standardized size measure similar to SLOC or Function Points, and these existing measures cannot be used because they measure *software* size, only, and COTS software projects do not only deliver software products. Rather, COTS software projects differ from traditional custom development projects by being part of *business transformation* initiatives and not stand alone software development projects. This implies that the projects not only deliver developed software but also deliver new business processes and changed organizations with new roles and jobs. These changes are partly performed independently of the COTS software package to improve organizational performance and partly necessitated by the COTS package because the functionality of a package to some extent dictates how you have to do your business[1].

[1] Of course, one could also perform business process reengineering activities in connection with custom development projects. The difference is that you do not *have to* since you can always customize the functionality to an existing organization and the way it does its work. On the other hand, the functionality of the COTS software packages

Now, the size of the non-software deliverables are generally not correlated with the size of the software deliverables. As a consequence, software size metrics such as SLOC and Function Points cannot be used to size the deliverables of COTS software projects.

Therefore, it would not make sense to define productivity as, say, the number of Function Points produced per unit effort. Of course, one could envisage using SLOC or Function Points to size the *software* deliverables of these COTS projects for benchmarking purposes, but this would not remove the problem of having multi-dimensional outputs since the other deliverables must still be sized. Though theoretically feasible, the fact is that these size metrics are used neither in Ernst & Young nor in Andersen Consulting for benchmarking the productivity of COTS software projects[2]. Since, to our knowledge, there are no international, inter-organizational standards similar to Function Points for sizing COTS software projects, Ernst & Young and Andersen Consulting are developing their own intra-organizational standards to enable comparisons of project performance within the companies.

There are several groups interested in intra-organizational productivity benchmarking of COTS software projects. *Customers* of COTS software projects within the Enterprise Resource Planning (ERP) market (e.g. products like SAP, PeopleSoft, BAAN and Oracle) demand that productivity benchmarks are included in proposals. Therefore, COTS project *bidders* must provide benchmarks to stay competitive. *Organizations* use benchmarks internally to evaluate projects. *Project managers* and *methodologists* need benchmarks to identify best practice processes and technologies. The need is clear. It is, however, less than clear how to benchmark COTS software projects since i) they are characterized by having multi-dimensional outputs and thus, require using multi-dimensional size measures, and ii) there likely is (dis)economies of scale.

At a first glance, the problem of measuring productivity may seem trivial using a simple measure defined as the ratio:

$$P = \frac{y}{x}$$

Equation 1. A simple productivity measure

where P is the productivity, y is the output and x is the input. For example, if x is the project effort in workdays and y is the amount of product delivered, measured in function points, then P measures the productivity as the number of function points produced per workday. We have illustrated this for the Albrecht-Gaffney [1] dataset in Figure 1 (see also Table 2) where we observe that project 23 has the highest productivity (P=199/0.5=398). Alternatively, we may present the productivity results on a *normalized* scale, i.e. a scale from zero to one, by dividing all numbers with the highest, P_{MAX}. For the Albrecht-Gaffney dataset P_{MAX} =398 i.e. the productivity of project 23. Project 23 thus has a productivity equal to 1. Using this normalized scale, the productivity of e.g. project 20 is:

$$\frac{P_{20}}{P_{MAX}} = \frac{\frac{1572}{61.2}}{\frac{199}{0.5}} = \frac{25.7}{398} = 0.06$$

Equation 2: A simple normalized productivity measure

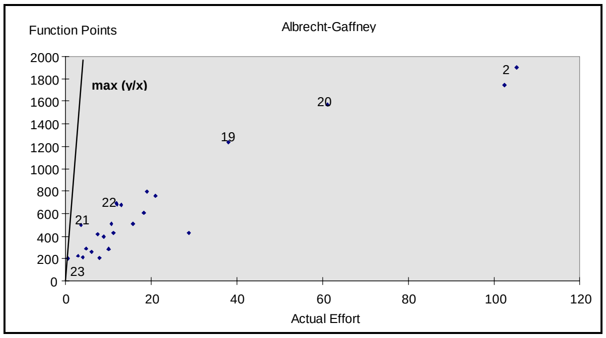

Figure 1. Benchmarking Albrecht-Gaffney projects assuming constant returns to scale (CRS). The straight line is the CRS frontier.

An obvious objection against this approach is that it is probably not fair to compare a small (0.5 workmonths) project with a large (61 workmonths) project. It seems more fair to compare a project with other projects of similar size since there may be economies as well as diseconomies of scale in software projects [4][6]. Assuming variable returns to scale (VRS), one pragmatic approach is to define a non-parametric best practice frontier in this two-dimensional space. This idea is illustrated in Figure 2 where the dotted line represents the constant returns to scale (CRS) best practice frontier, and the solid line represents the VRS best practice frontier. In this VRS scheme, project 20 is on the front with P=1.0 in stead of being highly unproductive (P=0.06) in the CRS scheme. Similarly, e.g. project 10 is benchmarked against

imposes some limitations on how the organization may perform their work, unless you rewrite major portions of the software. Thus, a software package enforces some changes in work processes. That is, there is always *some* business process reengineering activities carried out in these projects.

[2] Function Points is, however, used in these companies for inter-organizational benchmarking of custom development projects since it is a standard.

the line segment between projects 19 and 22 in stead of against the dotted CRS line.

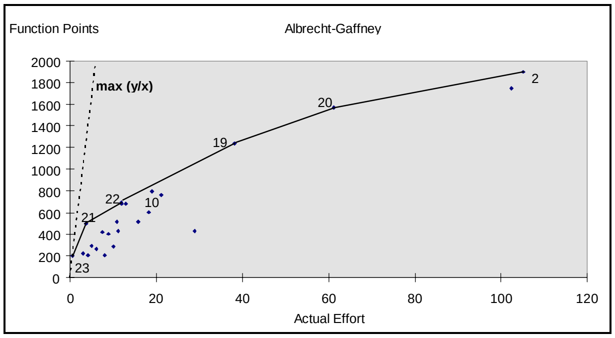

Figure 2. Benchmarking Albrecht-Gaffney projects assuming variable returns to scale (VRS) and using a non-parametric frontier. The dotted straight line is the CRS frontier. The broken line is the VRS frontier.

A second problem arises with the simple ratio measure in Equation 1 when there are multiple inputs and outputs. This is the case with our COTS dataset which uses a multi-dimensional size measure. See Table 1. In this case, it seems reasonable to construct a productivity measure similar to Equation 3:

$$P = \frac{\sum_{j=1}^{n} a_j Y_j}{\sum_{k=1}^{m} b_k X_k}$$

Equation 3: A multi-dimensional productivity measure

In Equation 3, a_j and b_k are weights reflecting the relative importance of the different outputs and inputs, respectively. The normalized productivity can still be defined in a way similar to Equation 2, i.e. P/P_{MAX}.

In this paper we propose to use Data Envelopment Analysis (DEA) to benchmark software projects because DEA addresses the problem of comparing similar projects with each other (i.e. using a VRS model) in a normalized, multi-dimensional space.

To our knowledge, many papers have been published on DEA[3], but only four papers have used DEA to analyze software projects [4][5][6][14]. It is surprising that DEA has not gained more widespread use in the software engineering community given its popularity in other disciplines. We show in section 3 that the four papers using DEA to analyze software projects partly suffer from methodological flaws and partly use DEA where simpler methods could have been used. Therefore, we see the need to provide an intuitive introduction to DEA from a practitioner's point of view. We do this by providing a tutorial before applying DEA to benchmark the productivity of 30 industrial COTS software projects. The tutorial emphasizes the strengths as well as the limitations of DEA in the context of benchmarking COTS software projects. We believe it is the first time DEA is used to analyze COTS software projects. Furthermore, we believe it is the first time DEA is used to test hypotheses and where significance levels are reported when analyzing software projects.

The paper is organized as follows. Section 2 presents DEA in a tutorial fashion and discusses its strengths and limitations. Section 3 presents related work using DEA to analyze software projects emphasizing the flaws of this work. Section 4 briefly describes the COTS data used in the analysis. Section 5 presents the results of analyzing the Albrecht-Gafney dataset using DEA as well as the results analyzing the COTS data. The main purpose of analyzing the Albrecht-Gaffney dataset is to provide an intuitive, tutorial example of the use of DEA. Section 6 concludes.

2. Data Envelopment Analysis

The Data Envelopment Analysis (DEA) method was originally developed by Charnes, Cooper and Rhodes [9] handling CRS (constant returns to scale), only. Førsund and Hjalmarson [12] enhanced the method to handle VRS (variable returns to scale) productivity comparisons. When performing DEA, the first step is to decide whether to use a CRS or a VRS model. For software projects it is reasonable to assume VRS. For small projects it is reasonable to assume increasing returns to scale. A software project consists of fixed costs and variable costs. Fixed costs are independent of the size of the software, e.g. the technical infrastructure like development environment, standards, communication protocols and database system. On the other hand, for larger projects we observe decreasing returns to scale, meaning that average productivity is decreasing, due to the well known communication and complexity problems [7][8]. The VRS assumption is also supported by Banker, Chang and Kemerer's [4] findings.

There are two alternative ways to calculate the VRS efficiency[4] using DEA, input reducing or output increasing efficiency. These two measures are illustrated

[3] We found 285 hits in the INSPEC Electronics & Computing database 1989 - Oct 97 using the search term «data envelopment analysis» of which a large majority were in operational research journals.

[4] In the DEA convention, we use the terms efficiency and inefficiency in stead of the term productivity in the DEA context. Otherwise, we use the terms as synonyms.

in Figure 3 for project C where AB/AC and EC/ED are the input decreasing and output increasing efficiencies, respectively. Both are reasonable measures. We can either measure how much less effort that could have been used to produce the same amount of software, or alternatively, we can measure how much more software that could have been produced with the same amount of effort. In this paper we use the input reducing efficiency measure to illustrate the DEA method.

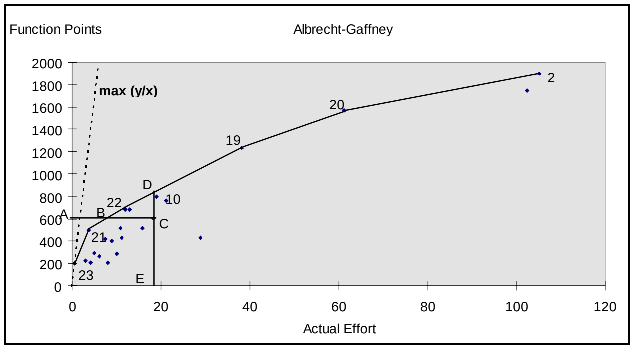

Figure 3: Measuring VRS efficiency using either input reducing or output increasing measures.

The idea behind the DEA approach is to calculate the front and the efficiency simultaneously. Using project C as example, we attempt to find the *minimal* effort required to produce the same amount of output as C produces. That is, we ask how much effort it would take for a best practice project to produce just as much output as C. This minimal effort is the effort at the point B which is a linear combination of the two frontier projects 21 and 22. These latter are termed *reference projects*. Thus, the idea is to move horizontally from C and towards the left until we hit the line segment at B. This is a minimization problem which can be solved using linear programming.

The formal problem thus becomes to minimize the objective function:

$$E_i = \min \theta_i \qquad (1)$$

subject to the constraints:

$$\sum_j \lambda_{ij} Y_{kj} \geq Y_{ki}, \forall k \qquad (1.1)$$

$$\theta_i X_{mi} \geq \sum_j \lambda_{ij} X_{mj}, \forall m \qquad (1.2)$$

$$\sum_j \lambda_{ij} = 1 \qquad (1.3)$$

$$\lambda_{ij} \geq 0, \forall j \qquad (1.4)$$

The constraint in (1.3) is the VRS constraint, and furthermore:

- i - is the current observation
- j - is all the other observations with which observation i is compared
- m - is the number of inputs, in our case effort, only
- k - is the number of outputs, i.e. the multi-dimensional size metric for the COTS projects

The technicalities for solving the DEA problem in a computationally efficient manner on a computer is beyond the scope of this paper and is thus not discussed here.

2.1 Limitations of DEA

The main weakness with DEA is that extreme observations highly impact the frontier. Therefore, some kind of sensitivity analysis is required to detect outliers and assess the robustness of the frontier.

Also, a considerable number of observations are characterized as efficient unless the sum of the number of inputs and outputs is small relative to the number of observations. Specialized units will typically be on the frontier. For example, a COTS project that produces reports, only, will probably produce more reports than any of the other projects that are producing both reports, interfaces and conversions, and therefore this atypical project will be deemed efficient by DEA.

The output from DEA, the efficiency measures, does not have a normal distribution that lends itself to simple statistical analysis since the distribution is truncated at 1. There are, however, more advanced techniques that may be used such as e.g. Tobit regression analysis.

As with most other multi-dimensional measures, DEA does not solve the problem of weighting the dimensions. All dimensions are normalized, i.e. have equal weights. Function Point based productivity measures suffer from similar problems since their weights were based on analyzing one single dataset.

3. Related work

Banker and Kemerer [6] use DEA to test whether there are economies as well as diseconomies of scale in software projects and to identify the optimal project size with respect to maximizing productivity. They apply the CRS variant of DEA on eight single input - single output datasets, including the Albrecht-Gaffney [1] dataset. For the Albrecht-Gaffney dataset they find that the most productive project is project 23 in Figure 1 (199 function points, 0.5 workmonths. See Table 2). The merit of Banker and Kemerer is that they introduce DEA to the software engineering community.

However, for this trivial two-dimensional case, we observe that the same result could have been found with simpler methods than DEA such as visual inspection of the scatter plot in Figure 1 or by calculating all the simple y/x ratios and then sorting them in a spreadsheet.

Also, we observe in Figure 1 that the most productive project in the Albrecht-Gaffney dataset is the smallest project thereby contradicting their own conjecture of economies as well as diseconomies of scale. We recognize, however, that the Albrecht-Gaffney dataset probably is an exception among their eight datasets.

Banker, Chang and Kemerer's paper [4] is an extension of [6] employing the DEA-based F-test of Banker and Chang verifying their previous results of economies as well as diseconomies of scale in software projects.

Banker, Datar and Kemerer [5] employ a variant of basic CRS DEA that is extended in two orthogonal directions. The first extension is called Stochastic DEA (SDEA). SDEA is stochastic in the sense that in addition to productivity related deviations, it also allows for the impact of random errors in model specification and measurement. The second extension extends DEA to analyze the effects of several alleged productivity factors (such as using or not using «peer reviews».). This doubly extended DEA model is used to evaluate the effect of five productivity factors on 65 software maintenance projects.

SDEA is a conceptually appealing model. In practice, however, the problem of how to distinguish random errors from inefficiency is not solved. We still need to use common sense to evaluate if the frontier and the individual ranking of projects is reasonable and fair. Regarding the first type of error, model specification error, we find it more intuitive to perform sensitivity analysis by removing one project at a time from the frontier and study the effect on the average productivity rates. Regarding the second type of error, measurement errors, we still need to use common sense and judgment in ranking individual projects taking into account, say, the limited inter-rater reliability of function point counts and effort figures.

Parkan, Lam and Hang [14] use DEA to measure the performance of individual projects where DEA is used as a part of an organization's reward structure. They apply the VRS model on one dataset with eight projects. The dataset has four inputs and one output. However, they have not commented on the fact that they use a VRS model and why. With four inputs and only eight projects, three out of the eight projects are efficient. The robustness of this result is not commented.

Below follows a summary of other uses of DEA in the IT business (i.e. other than software projects).

Fisher and Sun [11] use DEA to evaluate the individual performance of 22 e-mail packages using the VRS model. The dataset has five inputs and four outputs. Using all inputs and outputs they find four efficient e-mail packages. One project is in the reference set for all but two of the 22 packages. Fisher and Sun do not comment on the rationale for choosing a VRS rather than a CRS model. Also, they do not comment on why one single package serves as reference for almost all other packages, nor do they do a sensitivity analysis by removing this package which obviously is extreme in one or more of the output dimensions.

Thore, Phillips, Ruefli and Yue [17] use DEA to rank the efficiency of 44 U.S. computer companies using six inputs and three outputs. They find that 11 companies are efficient using both CRS and VRS models. The robustness of this result is not discussed. Sensitivity analysis is not done.

Mahmood [13] use DEA to evaluate organizational efficiency of IT investments using a dataset with 81 firms and eight inputs and ten outputs per firm. The results indicate that two-thirds of the firms are efficient. It is not documented whether a CRS or a VRS model is used. However, using any of these two models, there will likely be many firms on the frontier because of the large number of dimensions. The robustness of the results are not discussed. Mahmood also compares the efficient group of firms with the non-efficient group based on differences in means but without testing the significance of these results.

Doyle and Green [10] use DEA to benchmark 22 microcomputers using one input and four outputs. The merit of their paper is in providing an excellent presentation of DEA and a comparison of DEA with regression analysis.

In summary, previous studies suffer from several major flaws.

- They use DEA for two-dimensional datasets where we have shown that simpler methods could have been used.
- When using a VRS model in multi-dimensional datasets, it is applied to too small datasets compared to the number of dimensions. In such a case, a VRS model does not make sense as too many projects will be on the DEA frontier.
- Sensitivity analysis is not a routine part of DEA analysis. Sensitivity analysis must be done when using DEA because the method is extremely sensitive to outliers.

4. COTS data

The data set used for this validation consists of 48 completed COTS[5] projects. The data have been gathered since 1990, and it is an ongoing effort. All the projects are industrial projects spanning from 100 to 20.000 workdays, and there are 10 factors for sizing the product. See Table 1. These 10 factors constitute the intra-

[5] All the COTS projects in the sample are SAP R/3, i.e. it is a homogeneous data set.

organizational size metric standard[6] in Andersen Consulting for benchmarking the productivity of COTS software projects. A more detailed description and explanation of these size metrics is beyond the scope of this paper since the main purpose is to demonstrate and apply the DEA method. Interested readers are referred to [15] or [16].

Table 1: Descriptive statistics for COTS data set

Variable	N	Mean	Min	Max
Users	**48**	**346.5**	**7**	**2000**
Sites	**48**	**10.25**	**0**	**98**
Plants	**48**	**7.35**	**0**	**98**
Companies	**48**	**2.833**	**1**	**35**
Interfaces	**46**	**13.07**	**0**	**50**
EDI	**35**	**1.857**	**0**	**10**
Conversions	**37**	**18.38**	**1**	**93**
Modifications	**39**	**9.74**	**0**	**30**
Reports	**44**	**44.16**	**0**	**100**
ModulNo	**48**	**4.500**	**1**	**8**

We observe that we have a relatively large number of size dimensions compared to the number of projects. In addition, there were missing values for some of the observations. Therefore, we had to i) reduce the number of dimensions (as discussed in section 2.1) and at the same time ii) to use variables giving us the largest possible sample. We decided to use best subset regression analysis plus expert knowledge to determine which variables to include in the model. The benefit of a best subset regression model is that it identifies the minimum set of variables having high explanatory power of the variation in effort and simultaneously removing those variables that are correlated with the variables in the best subset model. Therefore, the variables in a best subset regression model seem representative as a multi-dimensional size metric of these COTS projects.

We ended up with one input (effort) and three outputs (Users, EDI, Conversions) resulting in a dataset with 30 observations due to missing values in these three variables.

5. Results

In this section, we provide the results for the Albrecht-Gaffney dataset as well as for the COTS dataset. We have included the Albrecht-Gaffney dataset mostly because it is instructive to discuss the results of the DEA method using a single input - single output dataset which presumably is familiar to most readers. Also, it is interesting to compare our VRS frontier results to Banker and Kemerer's [6] CRS frontier results.

From a benchmarking perspective, there are several alternative measures one can use to evaluate and compare projects. In Table 2, we show three alternative measures:

- Simple productivity which we define as the number of function points per unit of effort (P)
- CRS efficiency (E_{CRS})
- VRS efficiency (E_{VRS})

The first measure can only be used for single input - single output datasets. The other two measures can be calculated using DEA when one has to deal with multiple input - multiple output datasets. Common sense plus data analysis is required to determine which measure, CRS or VRS, is most appropriate.

Table 2: Efficiency results for Albrecht-Gaffney dataset

Project ID	Actual Effort	Function Points	P	E_{CRS}	E_{VRS}
1	**102,4**	**1750**	**17**	**0,042714**	**0,829427**
2	**105,2**	**1902**	**18**	**0,045226**	**1**
3	**11,1**	**428**	**39**	**0,09799**	**0,25752**
4	**21,1**	**759**	**36**	**0,090452**	**0,708999**
5	**28,8**	**431**	**15**	**0,037688**	**0,100325**
6	**10**	**283**	**28**	**0,070352**	**0,136512**
7	**8**	**205**	**26**	**0,065327**	**0,070224**
8	**4,9**	**289**	**59**	**0,148241**	**0,291206**
9	**12,9**	**680**	**53**	**0,133166**	**0,868856**
10	**19**	**794**	**42**	**0,105528**	**0,876914**
11	**10,8**	**512**	**47**	**0,11809**	**0,380298**
12	**2,9**	**224**	**77**	**0,193467**	**0,261198**
13	**7,5**	**417**	**56**	**0,140704**	**0,366024**
14	**12**	**682**	**57**	**0,143216**	**0,941065**
15	**4,1**	**209**	**51**	**0,128141**	**0,147071**
16	**15,8**	**512**	**32**	**0,080402**	**0,25995**
17	**18,3**	**606**	**33**	**0,082915**	**0,441553**
18	**8,9**	**400**	**45**	**0,113065**	**0,288775**
19	**38,1**	**1235**	**32**	**0,080402**	**1**
20	**61,2**	**1572**	**26**	**0,065327**	**1**
21	**3,6**	**500**	**139**	**0,349246**	**1**
22	**11,8**	**694**	**59**	**0,148241**	**1**
23	**0,5**	**199**	**398**	**1**	**1**
24	**6,1**	**260**	**43**	**0,10804**	**0,184957**
Mean	**21,9**	**648**	**60**	**0,15**	**0,56**

P - Productivity defined as number of Function Points per unit of Actual Effort.

From Table 2, (P column) we observe that the most productive project in the Albrecht-Gaffney dataset is ID 23 (P=398). The productivity is defined as the simple y/x ratio of Function Points over Actual Effort (as in Equation 1). ID 23 is the same project that Banker and Kemerer

[6] This is the standard as per 1997. However, there is continuous research to improve these size metrics.

found to be the most efficient project using DEA. We further observe that the productivity for this project largely exceeds any of the other projects. Also, this project definitely is the smallest project in terms of effort (Actual Effort = 0.5). Therefore, project 23 is an outlier. In two dimensions, outliers like this project can be easily detected by visual inspections of Table 2 and Figure 1. However, in multiple dimensions one has to use sensitivity analysis to detect such outliers and other sources of error.

The CRS efficiency (column E_{CRS} in Table 2) in this single input - single output dataset is identical to making the simple ratios as in Equation 2, i.e. dividing all P's with P_{MAX}=398. For the Albrecht-Gaffney dataset we thus find only one project with E_{CRS}=1, namely ID 23, defining the frontier in a CRS best practice frontier. This CRS measure is the efficiency measure that Banker and Kemerer [6] used.

Table 3: Reference set for Albrecht-Gaffney dataset

Project ID	2	19	20	21	22	23
1	**0,54**	**0,00**	**0,46**	**0,00**	**0,00**	**0,00**
2	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
3	**0,00**	**0,00**	**0,00**	**0,76**	**0,00**	**0,24**
4	**0,00**	**0,12**	**0,00**	**0,00**	**0,88**	**0,00**
5	**0,00**	**0,00**	**0,00**	**0,77**	**0,00**	**0,23**
6	**0,00**	**0,00**	**0,00**	**0,28**	**0,00**	**0,72**
7	**0,00**	**0,00**	**0,00**	**0,02**	**0,00**	**0,98**
8	**0,00**	**0,00**	**0,00**	**0,30**	**0,00**	**0,70**
9	**0,00**	**0,00**	**0,00**	**0,07**	**0,93**	**0,00**
10	**0,00**	**0,18**	**0,00**	**0,00**	**0,82**	**0,00**
11	**0,00**	**0,00**	**0,00**	**0,94**	**0,06**	**0,00**
12	**0,00**	**0,00**	**0,00**	**0,08**	**0,00**	**0,92**
13	**0,00**	**0,00**	**0,00**	**0,72**	**0,00**	**0,28**
14	**0,00**	**0,00**	**0,00**	**0,06**	**0,94**	**0,00**
15	**0,00**	**0,00**	**0,00**	**0,03**	**0,00**	**0,97**
16	**0,00**	**0,00**	**0,00**	**0,94**	**0,06**	**0,00**
17	**0,00**	**0,00**	**0,00**	**0,45**	**0,55**	**0,00**
18	**0,00**	**0,00**	**0,00**	**0,67**	**0,00**	**0,33**
19	**0,00**	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**
20	**0,00**	**0,00**	**1,00**	**0,00**	**0,00**	**0,00**
21	**0,00**	**0,00**	**0,00**	**1,00**	**0,00**	**0,00**
22	**0,00**	**0,00**	**0,00**	**0,00**	**1,00**	**0,00**
23	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**1,00**
24	**0,00**	**0,00**	**0,00**	**0,20**	**0,00**	**0,80**

Assuming a VRS frontier in stead of a CRS frontier, we find six efficient projects (i.e. where E_{VRS}=1) as opposed to one (See Table 2 column E_{VRS}). The least efficient project is ID 7 (E_{VRS}=0.07). Among the 6 efficient projects, two are at the very end of the frontier, the smallest (project 23) and the largest (project 2). We also observe that two other frontier projects (projects 19 and 20) do not have any other projects in the neighborhood. We observe that the cluster of projects is between 0 and 20 function points. Only in this area are the results reasonably robust. For the other projects the results are less trustworthy. We also observe that visual inspection of scatter plots (see Figure 2) still can be used to identify the VRS frontier for this simple single input - single output dataset.

The assumption of VRS seems likely from the average efficiency numbers of E_{CRS} and E_{VRS} in Table 2. It is more reasonable that the average efficiency is around 56% than 15% for a group of homogenous projects conducted by the same organization. Also, a large project like ID 2 was highly inefficient when compared with the frontier line determined by ID 23 in the CRS model. In the VRS model, ID 2 has become efficient.

Table 3 shows which are the reference projects for each project in the Albrecht-Gaffney datset. The column headings show the IDs of the reference projects which are the same six projects constituting the frontier (IDs 2, 19, 20, 21, 22 and 23). Reading a row we can identify which are the reference projects for a given project. Taking e.g. ID 7, we find it has two projects in its reference set (IDs 21 and 23). The figures in the cells are weights indicating the relative importance of the reference projects. For ID 7, ID 23 is a more important reference project than ID 21 (98% vs. 2%). The practical benefit of this information is that the project manager of project 7 can identify which projects he ought to consult to improve his performance, in this case project 23, in particular.

Finally, we also observe that for this single input - single output dataset, the reference projects could have been just as easily identified by visual inspection of the scatter plot in Figure 3 in stead of using DEA, and the weights could have been determined by measuring with a ruler on the scatter plot diagram.

The full potential of DEA first becomes apparent for multi-dimensional datasets where visual inspections no longer can be used to detect the frontier nor simple y/x ratios can be used to calculate efficiency scores. Our COTS dataset is such a multi-dimensional dataset having 10 outputs (See Table 1).

To make reasonable and fair comparisons of the efficiencies of projects, the outputs must be related to the inputs. Therefore, the first step in a DEA is to select the inputs and outputs (See section 4). The average efficiency (E_{MEAN}), standard deviation (SD), minimum efficiency (E_{MIN}) and the number of efficient projects (N_{EFF}) for Albrecht-Gaffney and the COTS dataset are shown in Table 4. We observe that the figures are almost identical for the two datasets except that the COTS set has nine efficient projects vs. six for Albrecht-Gaffney. This is as we would expect since there are more dimensions for the COTS than for the Albrecht-Gaffney dataset. (See section 2.1 for explanation).

From a process improvement perspective, these average efficiency figures tell us that there is a potential for improvement of such projects close to 40%.

Table 4: Average efficiency results using DEA

	N	E_{MEAN}	SD	E_{MIN}	N_{EFF}
Albrecht-Gaffney	**24**	**0.56**	**0.36**	**0.07**	**6**
COTS	**30**	**0.56**	**0.36**	**0.06**	**9**

N - total number of projects, E_{MEAN} - mean VRS efficiency, SD - standard deviation of efficiency, E_{MIN} - efficiency of the least efficient project, N_{EFF} - number of efficient projects.

Table 5: VRS efficiency and reference set for COTS dataset

ID	E_{VRS}	48	101	111	133	137	140	142	158	168
1	**0,13**	**0,52**	**0,00**	**0,48**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
2	**0,41**	**0,00**	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
47	**0,24**	**0,29**	**0,00**	**0,58**	**0,00**	**0,00**	**0,00**	**0,13**	**0,00**	**0,00**
48	**1,00**	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
63	**0,18**	**0,47**	**0,00**	**0,53**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
73	**0,28**	**0,62**	**0,00**	**0,26**	**0,00**	**0,00**	**0,00**	**0,13**	**0,00**	**0,00**
101	**1,00**	**0,00**	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
109	**0,48**	**0,21**	**0,00**	**0,00**	**0,00**	**0,16**	**0,57**	**0,00**	**0,00**	**0,05**
110	**0,90**	**0,31**	**0,00**	**0,44**	**0,00**	**0,00**	**0,00**	**0,25**	**0,00**	**0,00**
111	**1,00**	**0,00**	**0,00**	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
112	**0,22**	**0,04**	**0,15**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,81**
113	**0,15**	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
127	**0,40**	**0,95**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,05**
133	**1,00**	**0,00**	**0,00**	**0,00**	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
136	**0,84**	**0,00**	**0,29**	**0,00**	**0,14**	**0,57**	**0,00**	**0,00**	**0,00**	**0,00**
137	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**
140	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**1,00**	**0,00**	**0,00**	**0,00**
142	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**1,00**	**0,00**	**0,00**
145	**0,33**	**0,33**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,67**
146	**0,11**	**0,47**	**0,00**	**0,53**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
147	**0,06**	**0,74**	**0,00**	**0,26**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
151	**0,72**	**0,00**	**0,00**	**0,00**	**0,18**	**0,31**	**0,00**	**0,00**	**0,00**	**0,51**
154	**0,29**	**0,22**	**0,00**	**0,78**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
155	**0,40**	**0,25**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,63**	**0,00**	**0,13**
158	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**1,00**	**0,00**
159	**0,59**	**0,50**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,37**	**0,00**	**0,12**
163	**0,51**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,13**	**0,73**	**0,13**
168	**1,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**1,00**
172	**0,19**	**0,74**	**0,00**	**0,26**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
174	**0,23**	**0,83**	**0,00**	**0,17**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**

Table 5 shows the individual efficiency scores as well as the reference projects for each project in the COTS dataset. Nine projects are fully efficient (E_{VRS}=1.00). The least efficient project is ID 147. We observe that in multiple-dimension datasets an inefficient project may have more than two projects in its reference set, e.g. ID 47 has three reference projects (IDs 48, 111 and 142).

5.1 Sensitivity analysis

DEA identifies best practice, not average or say the best 10 %, which makes the technique very sensitive to extreme observations. It is, therefore, necessary to do a sensitivity analysis of the frontier. There are several techniques (e.g. superefficiency [2] and analysis of reference units [18]) each with their strengths and limitations depending on the purpose of the analysis. The purpose of our analysis is twofold, first to determine the average efficiency of the COTS projects to quantify the potential for productivity improvement and second, to assess individual projects and identify their respective reference projects. For this double purpose, the simplest, and probably most reasonable, sensitivity analysis is to remove all the frontier projects one by one and study the effect on the mean efficiencies as well as study the effect on the efficiencies and stability of reference projects for individual projects. We concentrate on the first part, presented in Table 6. The other part should be fairly obvious.

The COTS dataset has 9 units on the front. We do the sensitivity analysis by removing each of the frontier projects one by one. We observe that none of the frontier projects are extreme, in the sense that their removal highly influences the average efficiency. That is, there is still a potential improvement of around 40%.

Table 6: Results of sensitivity analysis of COTS dataset

Project ID	E_{MEAN}	New ID
48	**0.61**	**None**
101	**0.54**	**None**
111	**0.55**	**None**
133	**0.56**	**151,136**
137	**0.54**	**None**
140	**0.54**	**None**
142	**0.55**	**None**
158	**0.54**	**None**
168	**0.59**	**145**

Project ID – ID of removed project, E_{MEAN} – mean of E_{VRS}, New ID - New projects on the frontier

5.2 Hypothesis testing

Apart from telling us how much room there is for efficiency improvement, average figures of DEA results (such as in Table 4) may be used to identify what characterizes the most efficient projects. This can be used to test hypotheses about the alleged superiority of a certain technology or a certain process improvement technique. For example, one can test whether a certain programming language or a database product or a process technique such as peer reviews improves efficiency or not.

We demonstrate the hypothesis testing technique used in conjunction with DEA by investigating whether the average efficiency vary with industry[7]. If it does, it might be unfair and unreasonable to compare the performance across industries for evaluation purposes. On the other hand, this information may be used to discover what COTS projects in a certain industry have in common that make them more efficient. The industries in our sample are shown in Table 7.

Table 7: Average Efficiency per industry in COTS dataset

Industry	N	Mean	Median
Manufacturing	**11**	**0.56**	**0.51**
Energy	**3**	**0.46**	**0.19**
Process	**7**	**0.36**	**0.28**
Consumer	**7**	**0.79**	**0.9**
Unclassified	**2**	**0.57**	**--**

The preliminary results in Table 7 suggest that projects in the Consumer industry are the most efficient (E_{VRS}= 0.79) and that projects in the Process industry are the least efficient (E_{VRS}= 0.36).

The significance tests in Table 8 confirm the preliminary result that the Consumer industry is more efficient than the other industries combined. We have used analysis of the variance (ANOVA) of the mean and Mann-Whitney of the median. Both tests are significant at the 5% level.

A problem with hypothesis testing with the DEA measures is the fact that they are truncated (at 1). ANOVA assumes a normal distribution and is therefore not completely correct, and the results from ANOVA should therefore be treated with care. Mann-Whitney does not have any such requirement concerning the distribution and is therefore more suited in this case. A third alternative is a DEA adjusted F-test developed by Banker [3]. This test, however, requires a large number of observations in the sample.

The significance tests in Table 9 confirm the preliminary result that the Process industry is significantly less efficient than the other industries combined. Process industry projects should therefore be compared with projects from other industries with caution.

Table 8: Efficiency of Consumer industry vs. the others using ANOVA and Mann-Whitney significance tests

Industry	ANOVA Mean	Mann-Whitney Median
Consumers	**0.79**	**0.9**
The Others	**0.48**	**0.33**
Significance level of difference	**0.04**	**0.03**

Table 9: Efficiency of Process industry vs. the others using ANOVA and Mann-Whitney significance tests

Industry	ANOVA Mean	Mann-Whitney Median
Process	**0.35**	**0.28**
The Others	**0.62**	**0.59**
Significance level of difference	**0.09**	**0.05**

6. Conclusion

The conclusions in this paper are of two kinds: i) conclusions on the results of the empirical study and ii) conclusions on the usefulness of DEA.

As for the results, we have shown that there is a large variation in project productivity, and as a consequence, there is a substantial improvement potential compared with the “best in class” projects. This is not surprising since we know that there is a large number of factors impacting on productivity, most notably personnel capability [7] [8].

The preliminary results of the hypothesis testing suggest that there are significant differences in productivity between projects in different industries. If this is the case, one should exhibit caution when benchmarking across industries. This result needs, however, further investigation.

The results support the findings of Banker et al. [6] and Boehm [7] that there are economies as well as diseconomies of scale also in *COTS* software projects. Thus, it is appropriate to use a VRS model rather than a CRS model.

Regarding the usefulness of DEA as a benchmarking method we argue that it is a useful, practical method to evaluate COTS software projects because these projects use multi-dimensional size measures. On the other hand, for custom development projects that base productivity measures solely on Function Points and effort, DEA does not add much value.

Also, DEA is appealing to a practitioner because its basic idea is simple and intuitive (despite the complexities of the algorithm). It makes sense to practitioners to

[7] Results concerning the effect of process improvement techniques are considered sensitive information to Andersen Consulting and can therefore not be published.

compare performance with best practice rather than with some theoretical optimal productivity.

Finally, this study suggests that DEA used together with methods for hypothesis testing may be a useful technique for assessing the effect of alleged process improvements.

The only real objection against DEA is that it normalizes all the dimensions. This may cause a potential bias in the results. Still, we recommend it as benchmarking method provided that the size dimensions are carefully selected.

Acknowledgements

The authors would like to thank Ernst & Young and Andersen Consulting for supporting this research.

References

1. Albrecht-Gaffney, A.J. and Gaffney, J.R. Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. IEEE Trans. Software Eng. 9, 6 (Nov 1983), 639-648.

2. Andersen, P. and Petersen, N.C. A Procedure for Ranking Efficient Units in Data Envelopment Analysis. Man. Sci. 39,10 (1993),1261-1264.

3. Banker, R.D. Maximum Likelihood, Consistency and Data Envelopment Analysis: A Statistical Foundation. Man. Sci. 39, 10 (1993), 1265-1273.

4. Banker, R.D., Chang, H. and Kemerer, C.F. Evidence on Economies of Scale in Software Development. Inf. and Software Tech. 36,5 (1994), 275-82.

5. Banker, R.D., Datar, S.M. and Kemerer, C.F. A Model to Evaluate Variables Impacting the Productivity of Software Maintenance Projects. Man. Sci. 37, 1 (Jan 1991), 1-18.

6. Banker, R.D. and Kemerer, C.F. Scale Economies in New Software Development. IEEE Trans. Software Eng. 15, 10 (Oct 1989), 1199-1205.

7. Boehm, B.W. Software Engineering Economics. Prentice-Hall: Englewood Cliffs, N.J. (1981).

8. Brooks Jr., F.P. The Mythical Man-Month - Essays on Software Engineering. Anniversary Edition, Addison-Wesley, Reading Massachusetts (1995).

9. Charnes, A., Cooper, W.W. and Rhodes, E. Measuring the Efficiency of Decision Making Units. Eur. J. Oper. Res. 2 (1978), 429-444.

10. Doyle, J.and Green, R. Strategic Choice and Data Envelopment Analysis: Comparing Computers across Many Attributes. J. Inf. Tech. 9,1 (Mar 1994), 61-9.

11. Fisher, D.M. and Sun, D.B. LAN-based E-mail: Software Evaluation. J. Computer Inf. Sys. 36,1 (Winter 1995-1996), 21-5.

12. Førsund, F.R. and Hjalmarson, L. Generalised Farrell Measures of Efficiency: An Application to Milk Processing in Swedish Dairy Plants. The Economic Journal 89 (June 1979), 294-315.

13. Mahmood, M.A. Evaluating Organizational Efficiency Resulting from Information Technology Investment: an Application of Data Envelopment Analysis. Inf. Sys. J. 4,2 (Apr 1994), 93-115.

14. Parkan, C., Lam, K. and Hang, G. Operational Competitiveness Analysis on Software Development. J. Oper. Res. Society 48,9 (Sep 1997), 892-905.

15. Stensrud, E. and Myrtveit, I. The Added Value of Estimation by Analogy – An Industrial Experiment. Proc. FESMA'98 (Antwerp Belgium, May 1998), Technologisch Instituut vzw, 549-556.

16. Stensrud, E. and Myrtveit, I. Human Performance Estimating with Analogy and Regression Models: An Empirical Validation. Proc. METRICS'98 (Bethesda MD, Nov 1998) 205-213.

17. Thore, S., Phillips, F., Ruefli, T.W. and Yue, P. DEA and the Management of the Product Cycle: the U.S. Computer Industry. Computers & Oper. Res. 23,4 (Apr 1996), 341-56.

18. Torgersen, A.M., Førsund, F.R. and Kittelsen, S.A.C. Slack Adjusted Efficiency Measures – The Case of Norwegian Labour Employment Offices. Memo, Dept. of Economics, University of Oslo (2,1994).