

**Comparative Study of
Project Effort Estimation Methods by using
Public Domain Multi-organizational
and Organization-specific Project Data**

Melanie Ruhe

Software Engineering Research Group,
Department of Computer Science,
University of Kaiserslautern

Performed at
the Centre for Advanced Empirical Software Research (CAESAR) and the School of
Information Systems, University of New South Wales, Sydney

In cooperation with
the Fraunhofer Institute for Experimental Software Engineering (IESE), Kaiserslautern

August 1999

Abstract

Over the past ten to fifteen years the use of software has increased exponentially. Software development became a critical process. Software products need to be developed within time and budget and, furthermore, to an agreed level of quality. Therefore, the development process has to be well planned, which includes estimates of time, effort and personnel requirements.

In the software industry it is usual to produce a product each time rather than produce the same product again and again as in manufactory. Furthermore, the opportunities for developing software, e.g. tools and programming languages, become more and more various. This leads to difficulties in estimating the costs for a software project, because there is no general rule for how to derive an estimate in an environment of marked change.

This research examines the International Software Benchmarking Standards Group (ISBSG) repository, which is a large database of completed software projects from different organizations that can be used for estimating the required effort for new software projects. The accuracy of the estimates based on this repository is compared with the results obtained from using a one-company data set from a company called Megatec. Therefore, this study mainly investigates the question: Is there any difference in accuracy between using multi-company data and data from one company as a base in the estimation process?

Ordinary least squares (OLS) regression and an analogy-based tool are applied on the two data sets and, hence, another question is, which of the applied techniques yields more accurate results when using the ISBSG repository. It is also examined whether the number of analogues or size adjustment has a positive influence on the accuracy. Furthermore, the metrics of the ISBSG repository are searched through for potential cost factors.

The results show that the accuracy of the estimates based on the Megatec data is significantly higher than the accuracy of estimates based on the ISBSG repository. In addition, it was found that OLS regression yields better results than analogy when using the ISBSG data. Neither linear size adjustment nor using two analogues could improve the accuracy for the ISBSG based estimates. The Megatec based estimates, however, got more accurate when using the linear size adjustment and also by using two analogues to predict the effort.

Acknowledgements

Grateful acknowledgement goes to my Australian supervisor Ross Jeffery, whose personal help and encouragement I always appreciated. I want to thank him for giving me the opportunity to study at the School of Information Systems at the University of New South Wales and to experience Australia not only academically, but also socially and culturally, which has been invaluable.

I extend my thanks to Fiona Walkerden for her generous help in the beginning of the study.

I especially want to thank my German supervisors: Isabella Wiczorek, whose encouragement and useful comments at each stage of the project has progressed for my work a lot. And Dieter Rombach, who made the student exchange possible and encouraged me to perform my project thesis in Australia. This was an outstanding experience, which will always be a memorable part of my life.

I extend my thanks to Catherine Mulligan who provided me with the necessary advice for my English and also to Rasmus Hofmann for his feedback.

Finally, I would like to thank my parents for their support they have ever made in order to provide me with the education that I have.

Table of contents

1. Introduction	5
2. Motivation and Research Objectives of this study	6
2.1. Motivation	6
2.2. Research Objectives and Questions	7
3. Background	8
3.1. Estimation by Analogy	8
3.2. Related Work	9
4. Research Methods	12
4.1. The data sets	12
4.1.1. The Megatec data set	12
4.1.2. The ISBSG repository	14
4.2. Study about the possible cost factors in the ISBSG data set	17
4.2.1. Correlation analysis	17
4.2.2. Stepwise linear regression	20
4.2.3. ANOVA	23
4.2.3.1. The metric <i>business area type</i>	23
4.2.3.2. The metric <i>development platform</i>	26
4.2.4. Binary variables	27
4.2.5. The metric <i>resource level</i>	29
4.3. The reduced ISBSG data set used for the estimations	30
4.3.1. The final subset of variables	30
4.3.2. The final subset of projects	31
4.4. Data splitting	31
4.4.1. The metric <i>development type</i>	32
4.4.2. The metric <i>development platform</i>	32
4.4.3. The metric <i>Client/Server</i>	34
4.5. Ordinary least square regression	34
4.6. ACE	36
4.7. Estimations in this study	41
4.8. Data Analysis	42
4.8.1. Evaluation criteria	42
4.8.2. Statistical tests	42
5. Results and Comparisons	43
5.1. Megatec based estimates	43
5.1.1. Ordinary least square regression	43
5.1.2. Analogy-based estimates	44
5.2. ISBSG based estimates	46
5.3. Comparison between Megatec and ISBSG based estimates	50
5.4. Comparison of ranking and selecting	51
5.4.1. The metric <i>development type</i>	51
5.4.2. The metric <i>Client/Server</i>	53
5.4.3. The metric <i>development platform</i>	55
5.5. Further results	55
6. Summary and conclusions	56
References	59

1. Introduction

Software has become a part of our normal life. Our dependence upon computer technology has become apparent as evidenced by the excitement caused by the Year-2000 problem. This indicates the great need for reliable software systems. One of the biggest challenges in the software development process is the prediction of time, personnel, and effort needed for the project. Significant over- and underestimates can be very expensive for the company and can destroy their reputation. Hence, estimation is a critical step in the development process.

There have been many research investigations into the evaluation of possible cost estimation models. These models have been applied to several data sets in different ways. Generally, a distinction is made between parametric and non-parametric models. The former ones are more common, e.g. the well-known COCOMO model [Boehm 81] or regression, and have been the subject of many studies. These models perform poorly when applied uncalibrated to other environments [Briand et al. 98]. Non-parametric modeling techniques, e.g. machine learning algorithms or analogy, do not assume a specific functional relationship between cost and influential factors on cost; they have become the focus of research during recent years [Shepperd and Schofield 97, Briand et al. 98, Mukhopadhyay et al.92]. Analogy, as a means of predicting project effort, has also been discussed recently and shows promising results. This method is easier to understand and apply than parametric methods, thus it has more practical importance for project managers.

There are two main problems associated with software cost estimation research. Perhaps the largest problem is the lack of data from completed software projects in order to draw conclusions that are applicable in a more general way [Heemstra 92]. Both parametric and non-parametric estimation models are based on data from previous projects. It takes a long time to collect a reasonable amount of data on completed software projects for using it as support for project management. Especially when using analogy, there is a need for a comprehensive database of metrics from completed software projects, because the probability to find a similar completed software project rises with the number of stored projects. The International Software Benchmarking Standards Group (ISBSG) started to collect project data in 1991; it is possible to become member of the ISBSG and put company data into the repository or purchase it, thus benefit from it by using it for building cost models. Deriving a suitable cost model, which matches the collected data, is also a challenge. The problem with cost models is that they do not perform equally when changing environments. In other words, an accurate cost model developed for one organization will not necessarily be as accurate in another one. Most of the time, companies develop their own model or use standard cost models without calibrating them to their own data. Therefore, they often get unreliable estimates.

The objective of this project is to study the fitness of the ISBSG data set for cost estimation purposes. This will be done by applying several cost models and comparing their performance to each other and former studies. In the current study an analogy-based model is compared to regression analysis when using this ISBSG repository as a base for the estimation. The comparison is based on the prediction accuracy.

The first part of this research report is a short overview about problems in cost estimation and an explanation of the research objectives. Following this is background information

about analogy-based estimation and about former studies that are related to this project. After this details about the research methodologies and the data itself are presented. The results of the estimates, their discussion and conclusions complete the work.

2. Motivation and Research Objectives of the current study

2.1. Motivation

Research on software cost estimation models started more than 20 years ago [Brooks 75]. Nevertheless, general advice about which cost model fits to a certain organization or environment could not be drawn. What makes software cost estimation so difficult and still a focus for research?

The conundrum associated with software estimation is that it is essential to accurately estimate the effort and the time needed for a new software project, whilst it is quite impossible to precisely predict the effort required in advance.

Estimates are often done in a hurry because they need to be presented as early as possible to customers [Heemstra 92]. Also, the estimation often has to be done without clear specifications and with ever changing requirements during the development process; for instance, when the customer realizes new constraints or wants to change already specified requirements. If such changes are made, cost and time plans have to be adapted as well.

Furthermore, there are difficulties in determining tailored cost factors, which have an influence on the predictions for a certain project in advance [Heemstra 92]. There is also a lack of past project data. If the data set is not representative for the target projects, general conclusion cannot be drawn about the usefulness of the model.

The rapid pace of change in Information Technology also poses a problem when trying to develop a reliable cost model. New programming languages, development tools, and strategies are steadily introduced. Their influence on cost estimates has to be investigated as well [Heemstra 92].

Previous cost estimates show that software is often more expensive than estimated and the project takes longer than expected:

In a survey of 112 information systems managers in the US [Lederer and Prasad 93]:

- 63% of projects significantly overran their estimates.
- 14% significantly underran their estimates.

In a survey of estimation in Netherlands [Heemstra 92]:

- 80% of projects overran budgets and duration.
- The mean overrun was 50%.
- 35% of the organizations made no formal estimates.

Moreover, there are examples that show that software does not meet the demands of the customer or includes mistakes that make the software unreliable in use and can even

represent a risk for human life. Well-known examples are the Ariane 5 failure [Ariane] and the accident of a Lufthansa A 320 in Warsaw in 1993 [Ladkin 98].

These and other examples show that there is still a need for an improvement in the software development process. Cost estimation represents one of the critical steps at the beginning of the development process. Therefore, there is still a motivation for research to improve estimation models in order to simplify the management of software projects and to derive more accurate predictions for cost, time, and personnel. The current study is a contribution to the research discussion about the certain cost models and the appropriateness of the ISBSG repository for use in these models.

2.2. Research Objectives and Questions

The main objective of this research study is to assess the fitness of the ISBSG repository for its use in software project effort estimation. A review of the cost estimation literature yielded no relevant research on the data in the ISBSG repository. It is a large data set, which is more likely to be representative for the possible target projects.

Walkerden and Jeffery [98] investigated the data that was collected from the Megatec organization. The data is used for the current study as well. It is source of target projects, on one hand, and on the other hand, the Megatec projects are used as source projects as well. We also obtained ISBSG based estimates for certain Megatec target projects. Therefore, the main research question is:

Is estimation of Megatec projects as accurate using ISBSG as using the Megatec data set?

Hypothesis: Effort estimation using ISBSG projects is not as accurate as estimates made using Megatec data.

The Megatec data can be categorized as a one-company data set whereas the ISBSG repository includes data from different organizations. Therefore, the question could also be:

Is there any difference in accuracy between using multi-company data and data from one company as a base in the estimation process?

Besides answering this main question, several other issues are studied:

The estimation techniques used are analogy and OLS regression. Therefore, we want to know:

1. Is there any difference in accuracy between using ACE¹ and OLS regression for the estimates?

This question is important in order to guide the choice of a suitable modeling technique. Previous researches have compared regression and analogy for different data sets [Walkerden and Jeffery 98, Briand et al.98 and Shepperd and Schofield 97]. The results

¹ ACE stands for Analogical Cost Estimation and is a tool that uses analogy to estimate the required effort of a software project.

of their studies are outlined in section 3. This study aims to discover whether the ISBSG data confirms the previous results or not.

2. What are the cost factors of the projects of the ISBSG repository?

Cost factors such as staff experience, product quality and use of tools influence the cost of a project. It is important to know whether there are cost factors collected within this data set. The ISBSG repository is investigated on potential cost factors with the help of statistical tests.

3. What metrics should be used to find analogue projects?

The choice of metrics for the analogy-based estimation is part of a discussion about the full ISBSG data set and is also dependent on the availability of metrics in the Megatec data that were collected in a comparable way.

3. Background

3.1. Estimation by Analogy

The basis of analogy is to find one or more completed projects that are similar to a new (or target) project. In order to find such analogous projects, completed projects have to be characterized in as much detail as possible. In general, good analogy-based estimates require very detailed information about the project, the software development team, and the development methods [Shepperd and Schofield 95].

After deciding about the variables that are to be used for the estimation, the evaluation of the new project can be started. This means software size and other metrics are predicted on the base of the data and information known about the project at the time of quotation. An expert usually does this.

The actual estimation involves a kind of case-based reasoning (CBR). CBR is a methodology to model human reasoning and thinking [CBR]. Previous experiences are used in order to solve a new problem. In the case of analogy, cost data from completed software projects is used for the effort prediction. Depending on the definition of similarity among software projects, the database is searched for matching or analogous projects. The effort for a new project is estimated based on one or more analogous projects. The effort can be derived based on adopting the effort of the similar project or adjusting it by system size or other weights. Adjusting the effort takes into account the differences between the target and completed projects; size adjustment is utilized in order to address differences in software size for example.

An expert may perform the steps of analogy as well. The valuable knowledge of experts is usually not captured in a repository. In order to address this issue, data about the experiences of completed projects has to be stored. Then, people can access this data and learn how to use it for cost estimation. Analogy is a very easy and understandable method and shows promise for use in industry. Furthermore, the estimation is not dependent on one person's knowledge anymore. This is demonstrated in figure 1.

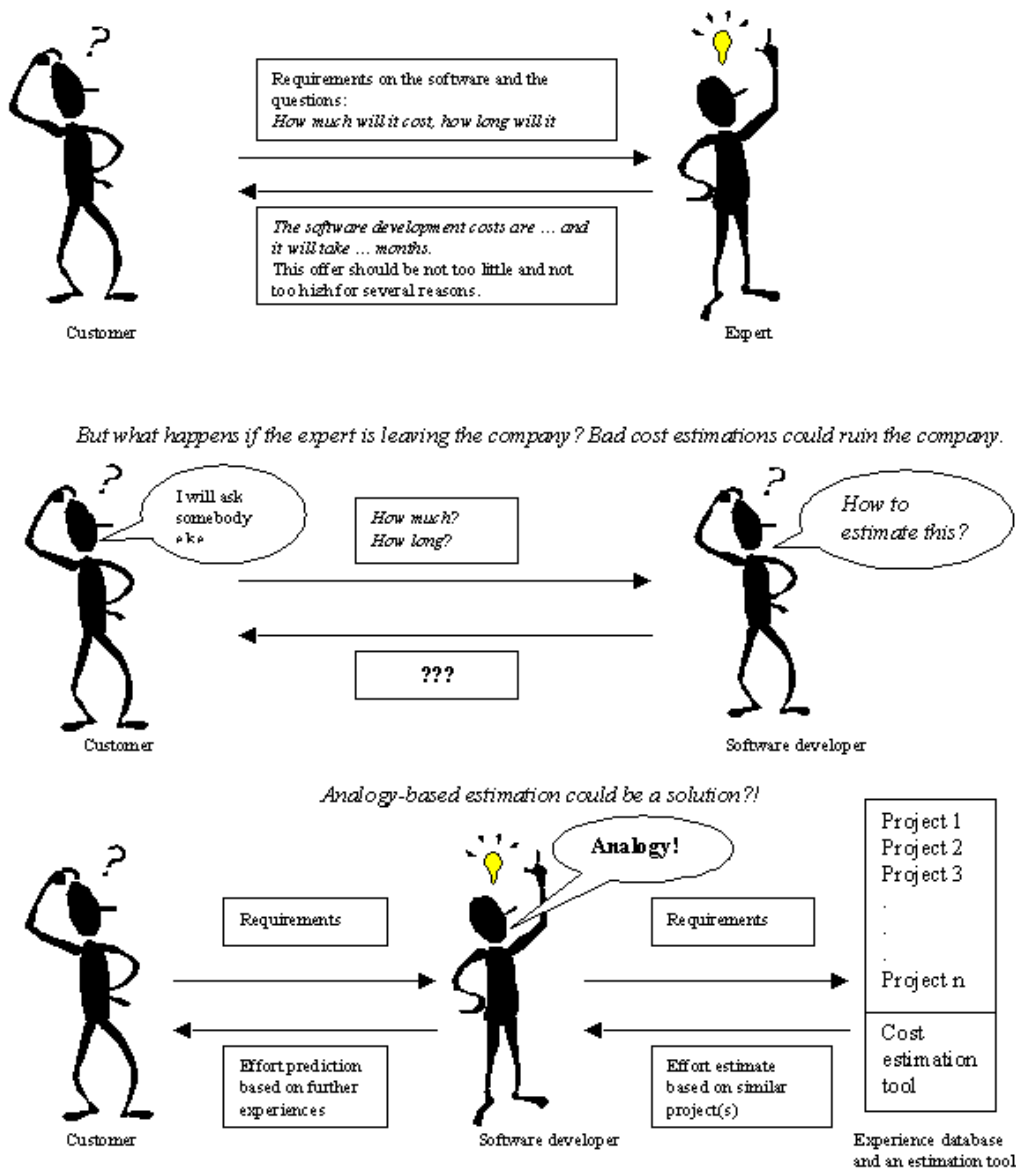


Figure 1. The advantage of Analogy!

Walkerden and Jeffery [98] summarize the steps for estimation by analogy as follows:

1. Measuring or estimating the values for cost factors for the target project;
2. Searching a repository of completed projects for projects similar to the target and selecting one or more projects as source projects;
3. Using the effort of the source analogue(s) as an initial estimate for the target project;
4. Comparing the known metric values for the target and source projects; and
5. Adjusting the effort estimate to compensate differences between target and source projects.

These steps are discussed in the same order in the study. The first step is applied by using data of Megatec projects as target projects.

3.2. Related Work

The work of Briand et al. [98] was based on the so-called “Laturi-database”, which includes 206 business software projects from 26 companies in Finland. This multi-

company database has the advantage that it includes a company-code identifying the source of each project. Therefore, it was possible to investigate the following two questions:

- (1) What modeling techniques are likely to yield more accurate results when using typical software development cost data?
- (2) What are the benefits and drawbacks of using organization-specific data as compared to multi-organization data set?

The project metrics that have been used in this study are:

- *total effort*² in hours,
- *unadjusted experience function points* (see [Briand et al. 98] for a description of this measure),
- *organization type*,
- *application type*,
- *target platform*, and
- *15 productivity factors*.

The following techniques were considered:

- Ordinary least squares regression and stepwise ANOVA, which are common parametric techniques [Conte et al. 86].
- Classification and regression trees (CART). They are a kind of binary tree with conditions in each node that split the data into small subsets fulfilling all the corresponding stipulations. The conditions are based on the metrics of the data set, e.g. “if system is distributed, then” ...or “if *effort* ≤ 10000 hours”. Each branch of the tree corresponds to one possible value, e.g. distributed or non-distributed system and yes or no. A stopping criterion determines the leaves (terminal nodes) of the tree. The average productivity of all projects within a terminal node is calculated and used for the effort prediction [Breiman et al. 84].
- Analogy-based estimation based on a distance function of [ANGEL]³, which basically measures the Euclidean distance in n-dimensional space, where each dimension corresponds to one variable.
- Combination of CART with regression analysis and CART with Analogy-based estimation

Magnitude of relative error (MRE), mean MRE, and prediction at level k (PRED(k)) were chosen for the evaluation. Results show that the performance of the modeling techniques were not significantly different, with the exception of the analogy-based models, which appear to be less accurate when predicting effort using multi-organizational data [Briand et al. 98]. No size adjustment was reported in detail in this study when using analogy. The authors mention that after performing size adjustment, the results did not change

² The names of metrics are always written in Italic.

³ ANGEL stands for *ANalogy softwarE tool* and is also a tool that uses analogy to estimate the required effort of a software project.

significantly. This is different to the results of Walkerden and Jeffery [98]. Furthermore, the use of organization specific models did not yield better results than generic multi-organization models when using standard cost factors.

Walkerden and Jeffery [98] also compared analogy and linear regression. They investigated the performance of people using analogy for the predictions. The performance of the two analogy-based tools ACE and ANGEL was also compared. They also distinguished between the performance of ANGEL, with and without size adjustments.

The data for the study came from the Australian software development organization “Megatec”; i.e. it is a one-company data set. The data and its method of collection are outlined in section 4.

The project metrics that were used in this experiment are:

<u>Metric</u>	<u>Measurement Scale</u>
• <i>total effort in hours</i>	ratio
• <i>unadjusted function points</i>	ratio
• <i>maximum team size</i>	ratio
• <i>distributed system</i>	nominal
• <i>programming language</i>	nominal
• <i>design experience</i>	ordinal
• <i>language experience</i>	ordinal
• <i>application experience</i>	ordinal

Table 1. Project metrics in the Megatec data set

The evaluation criteria used were absolute relative error (ARE) and mean ARE. ACE performed best on average of the four tools, when looking at the mean ARE. It also had the highest proportion of cases with the minimum ARE of the four tools, and the lowest proportion of cases with the maximum ARE [Walkerden and Jeffery 98]. Furthermore, the results showed that the subjects are better than tools when selecting an analogous project.

There are two studies that utilized the ISBSG data set: [ISBSG] and [Lokan 99]. The first one is a descriptive study done by the ISBSG itself. Examples of the areas it analyzes are *project size*, *project effort*, and other metrics, e.g. their range, distribution, and their relationship. Lokan [99] investigated the relationship between the five types of elements in function point analysis. His work has no influence on the current study, because he did not include *effort* when studying relationships within the ISBSG repository. Furthermore, his data varies from the one we use, because it includes only Australian software projects and different metrics, as for instance the five elements of function point analysis and as well a key for each organization that put data in the repository.

There are some reports that document the performance of ANGEL in comparison with linear regression for different data sets [Shepperd and Schofield 95, Shepperd et al. 96 and Shepperd and Schofield 97]. Their suggestions for further research are used for this study, thus they are outlined below:

Firstly, they show that for each data set the analogy-based approach outperformed the algorithmic approach using linear regression to calibrate the model to the particular data set. When using ANGEL, the weakness is its inability to cope with very small data sets and great variations in data sets [Shepperd and Schofield 95]. In addition, Shepperd et al. [96] recommend studying the effect of individual variables in finding analogies to analyze the individual contribution of each variable.

Based on former results relating to the applied methods, ordinary least squares regression and ACE were chosen as modeling techniques for the current study. ANGEL was not chosen, because there was no significant difference when compared with ACE [Walkerden and Jeffery 98]. ACE performed better and, therefore, seems to be more promising. The other different methods of the [Briand et al. 98] study are not used because they performed similarly to each other in the prior research.

4. Research Method

4.1. The data sets

The Megatec data is used for the evaluation of estimation using the ISBSG data. It is either used to obtain a target project or as source projects. The ISBSG projects are only used as source projects. Hence, it is possible to compare the results of estimates. Both data sets have certain properties: How are the software projects characterized in the ISBSG and the Megatec data set? What metrics are available in these data sets? Which of them are used for the estimations and why? The answers to these questions are part of the following sections.

4.1.1. The Megatec data set

The source for this project data is an Australian software development organization with about 50 employees that develop and distribute a range of computer products in Australia and overseas. It was one of the first software companies in Australia to gain Australian Standard 3563 (IEEE-Std.-1298) or in other words: a company that was highly motivated to provide good quality data and that was also interested in research results [Jeffery and Stathis 96]. Clients were charged by the hours spent on their projects. The reporting system breaks down how many hours were spent on which task and stage of each project. Effort expended by staff of the client organizations was not counted. Hence, the measurement of actual effort is very accurate. Furthermore, the projects include a broad range of different data-processing applications from a variety of industries, which gives the possibility for more general conclusions [Jeffery and Stathis 96].

The Megatec data is of high quality, which resulted from the method of data collection: The data collection was split into three parts. Firstly, all available documentation was used as base for the function point counting. Two independent raters with experience in the Albrecht standard [Albrecht and Gaffney 83] counted each of the 19 systems, which were all developed between 1990 and 1993. One of them was a paid external consultant, the other person was one of the researchers of the [Jeffery and Stathis 96] study. Hence, the reliability and also the accuracy of the counts can be ensured. Secondly, the experience of the project members was rated on an ordinal scale between 1 and 5. Thirdly, project managers were interviewed for better understanding of measured effort and function points.

The Megatec database [Jeffery and Stathis 96] includes the metrics mentioned in table 1 and, furthermore, the five function types and the fourteen general application characteristics as used in the Function Point Model [Albrecht and Gaffney 79]. The experience metrics are very useful to evaluate the amount of effort required and for determining analogues as well. The software projects were mainly written in C, COBOL or Powerhouse.

A scatter plot depicts the relationship between *effort* and *function points*. Three projects can be categorized as outliers. Another scatter plot shows the ID of each project where the outliers are excluded.

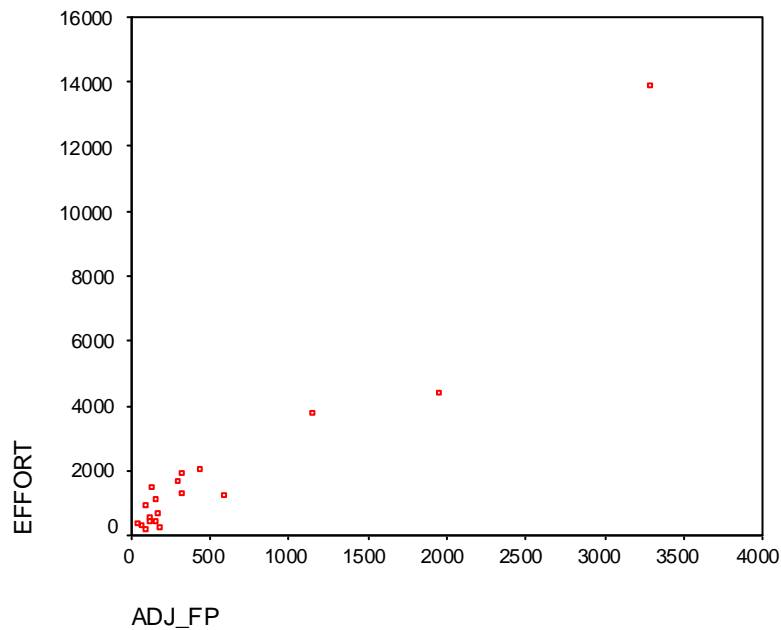


Figure 2. Scatter plot of effort against function points

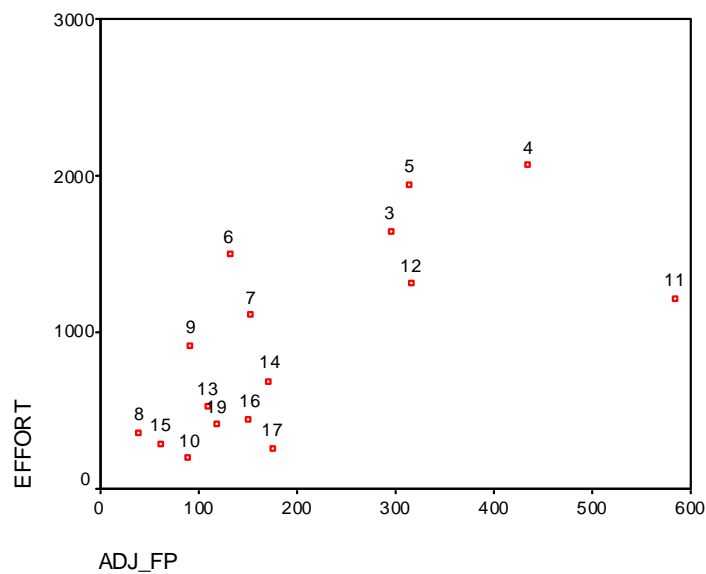


Figure 3. Scatter plot of effort against function points without projects 1,2 and 18

The full description of the data set can be obtained from [Jeffery and Stathis 96].

4.1.2. The ISBSG repository

A primary goal of ISBSG is to provide experience data and, a common language for the data collection in order to get data that is comparable and applicable for cost estimation. The projects have been voluntarily submitted by software practitioners, which could, therefore, reflect a higher productivity in the ISBSG repository than the industry averages [ISBSG].

The ISBSG repository (release 5, March 1998) consists of 451 projects from fourteen countries around the world; Australia is the largest contributor. The ISBSG repository contains a relatively large number of projects and, therefore, could manage the problem of small data sets. Furthermore, there are 38 metrics collected that describe each project. This does not indicate that the results may improve; models that are composed of many metrics do not have to be better than models of only a few metrics. Metrics can be strongly related to each other, thus it has to be investigated first, which metrics are really relevant [Conte et al. 86].

The ISBSG data set was collected with the help of questionnaires. The questionnaire is separated into six parts:

- Project attributes
- Project work effort data
- Project size data (function points)
- Project quality data
- Project cost data
- Project estimation data

This survey had to be answered and sent back. There was neither any interview nor were all the people familiar with the cost estimation research area. Each project submitted to the ISBSG repository was validated against specific quality criteria and rated as "A" (295 projects), "B" (101) or "C" (55), where "A"-rated projects are those which satisfy all quality criteria. The table below gives a short overview about available metrics, their definition and kind of scale [ISBSG]:

Table 2. Project metrics used in the ISBSG database

No.	Variable/Metric	Definition	Kind of Scale	Number of categories
1	<i>ASMA_Project ID</i>	primary key for identifying projects	nominal	-
2	<i>Data Quality Rating</i>	validation of the quality of the data	ordinal	3
3	<i>Count Approach</i>	description of the technique used to count the function points	nominal	9
4	<i>Function Points</i>	adjusted function point count number	ratio	-
5	<i>Value Adjustment Factor</i>	adjustment to the function points	ratio	-
6	<i>FP Standards</i>	function point counting standard used for the project	nominal	5

7	<i>Counting Technique</i>	method used to count the FP	nominal	6
8	<i>Reference Table Approach</i>	describes the approach used to handle counting of tables of code or reference data	comment field	-
9	<i>Summary Work Effort</i>	total effort in hours recorded against the project	ratio	-
10	<i>Recording Method</i>	method used by the project team to determine the project effort	nominal	7
11	<i>Resource Level</i>	description of the team or (development, supporting team, ...) effort	ordinal	4
12	<i>Max Team Size</i>	maximum number of people on the project	ratio	-
13	<i>Development Type</i>	new development, enhancement or re-development etc.	nominal	3
14	<i>Development Platform</i>	primary platform (PC, Mid Range or Mainframe)	nominal	3
15	<i>Language Type</i>	defines the language type	nominal	4
16	<i>Primary Programming Language</i>	actual language used for the majority of the development	nominal	21
17	<i>DBMS Used</i>		Y/N	-
18	<i>Upper CASE Used</i>		Y/N	-
19	<i>Lower CASE (with code gen) Used</i>		Y/N	-
20	<i>Lower CASE (no code gen) Used</i>		Y/N	-
21	<i>Integrated CASE Used</i>		Y/N	-
22	<i>Used Methodology</i>	Use structured methodology?	Y/N	
23	<i>How Methodology Acquired</i>	whether the methodology was purchased or developed	nominal	-
24	<i>Development Techniques</i>	techniques used during development	numeration	-
25	<i>Project Elapsed Time</i>	total elapsed time for project in months	ratio	-
26	<i>Project Inactive Time</i>	number of months in which no activity occurred	ratio	-
27	<i>Implementation Date</i>	actual date of implementation	interval	-
28	<i>Total Defects Delivered</i>	defects reported in the first month of system use	ratio	-
29	<i>User Base - Business Units</i>	number of business units that the system services	ratio	-
30	<i>User Base - Locations</i>	number of physical locations being serviced by the system	ratio	-
31	<i>User Base - Concurrent Users</i>	number of users using the system concurrently	ratio	-
32	<i>Country</i>		nominal	14
33	<i>Business Area Type</i>	Manufacturing, Personnel, Finance etc.	nominal	13
34	<i>Application Type</i>	Office information, Decision support etc.	nominal	7
35	<i>Client/Server?</i>		Y/N	-
36	<i>Architecture Description</i>		comment field	-
37	<i>Package Customization</i>	Was the project based on a package?	Y/N	-
38	<i>Degree of Customization</i>	How much customization was involved in the project?	comment field	

The function points were almost all counted by using the [IFPUG] standard. There is a wide range of *programming languages*; the systems are mainly written using ACCESS, COBOL, NATURAL, PL/1, and TELON. The range of *business area types* is also quite wide. The ISBSG software projects were performed in many different areas, whereas the Megatec data is only obtained from projects performed in the software development company Megatec. The *application types* of ISBSG are mainly management information systems, office information systems or transaction & production systems.

Unfortunately, there is neither any data about the experience of the software developers, nor any metric that identifies the company or gives information about the organization type of the company in the repository. Therefore, research as done by [Briand et al. 98], which compared estimations for multi-organization and organization-specific data, cannot be repeated with the ISBSG data only. Nor are there any indications or measures of the experience level of the development team. This makes it more difficult to find analogues.

Project delivery rate (PDR) as a measure of productivity was calculated. It is defined as:

$$PDR = \frac{Hours}{FP}$$

A high number indicates more work hours per function point and, therefore, shows that the productivity is low and vice versa in a small number. In manufactory productivity is usually defined as:

$$productivity = \frac{output}{input}$$

In this case output is the number of *function points* and input the hours of *effort*. In industry it is not as common to use this definition, because it is difficult to imagine what a function point actually is and how many of them can be developed in one hour. Therefore, we decided to use *project delivery rate* as measurement for productivity in this study.

If it is necessary, the *unadjusted function points* (UFP) can be calculated; *function points* and the *value adjustment factor* are given in the database and can easily be transformed with the following equation, which shows their relationship:

$$FP = VAF * UFP$$

Generally, *function points* are used in the current study. This decision was made because the ISBSG data set, where both measures are collected, contains a lot of missing values for the *value adjustment factor*.

For the estimation of *effort* as dependent variable not all of the 38 metrics were used. The choice of the independent variables is explained later on. At first, the full data set is described with the help of some statistics. The statistics are done using the SPSS tool (version 8.0), which is able to run most of the common tests, regression and all descriptives.

4.2. Study about the possible cost factors in the ISBSG data set

In order to use the ISBSG data for estimation, more information is needed to understand the data. In general, it is useful to look at the variables and their influence on *effort*. When looking at the data from different angles and exploring this, conclusions about possible cost factors can be drawn. For instance, it is important to know whether a relationship exists between *effort* and *software size* or *effort* and other metrics. If there is a certain relationship that can be put into a formula, predictions can be done very easily.

The ratio scaled metrics in the data set are shown in the following table for the full data set (451 projects):

Table 3. Statistics for the ratio metrics of ISBSG

Variable	Mean	Median	Std. deviation	Min	Max
<i>Function Points</i>	710	308	1266	9	17518
<i>Summary Work Effort</i>	6223	2362	12330	5	106480
<i>Max Team Size</i>	7	4	8	1	65
<i>Total Defects Delivered</i>	3	0	12	0	151
<i>Project Elapsed Time</i>	11	8	9	1	78
<i>Project Inactive Time</i>	1	0	2	0	12
<i>User Base - Business Units</i>	20	4	143	1	2000
<i>User Base - Locations</i>	65	5	973	1	2000
<i>User Base - Concurrent Users</i>	203	10	12	1	8000

The mean and median are measures of central tendency; the standard deviation measures the variability within the data [Fenton and Pfleeger 96]. Min and Max indicate the range of each metric with the minimal and maximal occurring value in the data set.

The normal distribution, which is indicated by equal values for mean, median and mode cannot be found for any variable [Fenton and Pfleeger 96]. Median and mode are always smaller than mean. The range for *function points* and *work effort* in ISBSG is large in comparison to the Megatec data. Especially the mean effort differs a lot; Megatec projects only required 1947 hours of *effort* on average. *Team size* also shows a large difference compared with a range of 1 to 10 people for Megatec projects. Brooks [75] says that a high number of team members usually does not improve the productivity because of time used for the group communication and organization. Hence, the relationship between *team size* and *PDR* or *effort* is not linear. The other documented metrics have no counterparts in the Megatec data set, but also show a quite broad spectrum.

4.2.1. Correlation analysis

A common first step of data analysis that involve many metrics is to run a correlation matrix [Fenton and Pfleeger 96]. This matrix can be examined for significant relations. As mentioned before, it is important to investigate whether the metrics that are contained

in the ISBSG repository, are related to each other. This is necessary because we want to examine whether a relationship exists between *effort* and other potential cost factors stored in the database. Therefore, we need to understand the data in general at first.

Correlation analysis is performed in order to see if there are metrics in the ISBSG repository which are highly associated [Fenton and Pfleeger 96]. Correlation measures how variables are related. There are different measures available depending on the distribution of the data. We already found that the ISBSG data is not normally distributed. Therefore, the Spearman rank correlation coefficient was chosen [Fenton and Pfleeger 96]. The value of the correlation coefficient ranges from -1 to 1, the sign of the correlation indicates the direction of the relationship, and the absolute value indicates the strength of the relationship. The full ISBSG data was used for determining the coefficients for the metrics of table 2.

It can be seen from table 4 (next page) that the independent metrics are significantly related to *effort*. The relationships are mainly positive ones, and the highest correlation can be found for *function points*. This means that *effort* increases with rising number of *function points*, which fits with the normal expectations. High correlation also exists between *effort* and both *maximum team size* and *project elapsed time*. This also doesn't surprise, because more team members cost more and the same applies for the elapsed time: as more time the project takes as more does it cost. The association of *effort* to *project inactive time* has to be a negative one. *Total defects delivered* are not highly correlated with *effort*. This also seems to be reasonable; defects should be kept as low as possible and not positively correlated to *effort*. There is also an association between *effort* and the number of users, counted in business units, locations, and concurrent users. The complexity of the software usually rises with a higher number of concurrent users. Therefore, *effort* also increases.

These three metrics, *user base - business units, locations, and concurrent users*, are also related to each other by showing a quite high correlation coefficient. Furthermore, they are slightly associated with *maximum team size*. This is probably also reasoned by the higher complexity of the software project. *Project elapsed time* shows quite high correlation to *function points* and to *team size*, which conforms to the normal expectations and also to Brooks [75], who also says that the elapsed time increases when more people work for a software project, as mentioned before. *Function points* and *team size* are also positively correlated, but the Spearman rank coefficient is quite low.

Table 4. Spearman rank correlation coefficients

<i>Summary Work Effort</i>	<i>Function Points</i>	<i>Max Team Size</i>	<i>Project Elapsed Time</i>	<i>Project Inactive Time</i>	<i>Total Defects Delivered</i>	<i>User Base - Business Units</i>	<i>User Base - Locations</i>	<i>User Base - Concurrent Users</i>		
1.000	.732	.633	.640	-.292	.104	.293	.204	.320	Correlation Coefficient	<i>Summary Work Effort</i>
	.000	.000	.000	.003	.027	.000	.004	.000	Sig. (2-tailed)	
451	451	280	378	103	451	204	202	199	N	<i>Function Points</i>
	1.000	.384	.610	.024	.124	.204	.054	.230	Correlation Coefficient	
		.000	.000	.808	.008	.003	.445	.001	Sig. (2-tailed)	<i>Max Team Size</i>
	451	280	378	103	451	204	202	199	N	
		1.000	.320	-.209	.098	.286	.367	.379	Correlation Coefficient	<i>Project Elapsed Time</i>
			.000	.062	.103	.002	.000	.000	Sig. (2-tailed)	
		280	275	80	280	113	111	109	N	<i>Project Inactive Time</i>
			1.000	.243	.088	.017	-.040	.140	Correlation Coefficient	
				.013	.087	.826	.613	.079	Sig. (2-tailed)	<i>Total Defects Delivered</i>
			378	103	378	162	161	158	N	
				1.000	.194	-.181	-.184	-.087	Correlation Coefficient	<i>User Base - Business Units</i>
					.050	.067	.065	.389	Sig. (2-tailed)	
				103	103	103	102	101	N	<i>User Base - Locations</i>
					1.000	-.159	-.080	-.065	Correlation Coefficient	
						.023	.258	.365	Sig. (2-tailed)	<i>User Base - Concurrent Users</i>
					451	204	202	199	N	
						1.000	.552	.455	Correlation Coefficient	<i>Summary Work Effort</i>
							.000	.000	Sig. (2-tailed)	
						204	202	199	N	<i>Function Points</i>
							1.000	.547	Correlation Coefficient	
								.000	Sig. (2-tailed)	<i>Max Team Size</i>
							202	198	N	
								1.000	Correlation Coefficient	<i>Project Elapsed Time</i>
									Sig. (2-tailed)	
								199	N	<i>Project Inactive Time</i>

4.2.2. Stepwise linear regression

One of the general questions is, whether a relationship exists between *effort* and other potential cost factors stored in the database. Stepwise linear regression is applied to examine which ratio scaled variables have an important influence on *effort* in the ISBSG repository.

At each step of the regression, variables in the equation are evaluated according to the selection criteria for removal; variables not in the equation are evaluated for entry. This process repeats until no variable in the block is eligible for entry or removal [Norusis 98].

When applying a stepwise linear regression, *effort* is the dependent variable and the block of independent variables is the same as in table 2. The order of entered variables shows their importance on *effort*. The full ISBSG data set was used for the stepwise linear regression.

Table 5. Variables entered in the stepwise linear regression (451 projects)

Model	Variables Entered	R square	Method
1	<i>Max Team Size</i>	0.635	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	<i>Function Points</i>	0.739	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	<i>User Base - Concurrent Users</i>	0.787	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
4	<i>Total Defects Delivered</i>	0.809	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
5	<i>Project Elapsed Time</i>	0.825	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
6	<i>Project Inactive Time</i>	0.851 □	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

Dependent Variable: *Summary Work Effort*

It can be seen that *maximum team size* enters the regression model at first. The value of R^2 is 0.635; this means that 63.5 % of the variability in *effort* can be explained only by *team size*. Including *function points* in the model increases the value of R^2 by 10%; the other variables don't improve the value that much. Finally, the regression model can explain 85 % of the variability in effort.

This order of entering the regression model does not follow the order of the correlation coefficients, where *function points* showed a higher correlation to *effort* than *maximum team size*. The number of projects used for the analysis could reason this; *maximum team size* has got many missing values, thus, there are only 280 software projects left for determining the coefficient, whereas the number of *function points* is available for all 451 projects. The subset of 280 projects obviously differs somehow from the full data set; there are probably outliers included in the full data set that are not included in the subset. Thus, they influence the regression and the coefficients. This is investigated with a scatter plot.

A scatter plot views the relative positions of pairs of data points and the likelihood of an underlying relationship between them can be seen [Fenton and Pfleeger 96]. The

influence of the outliers on the regression and the regression line is demonstrated in the following pictures:

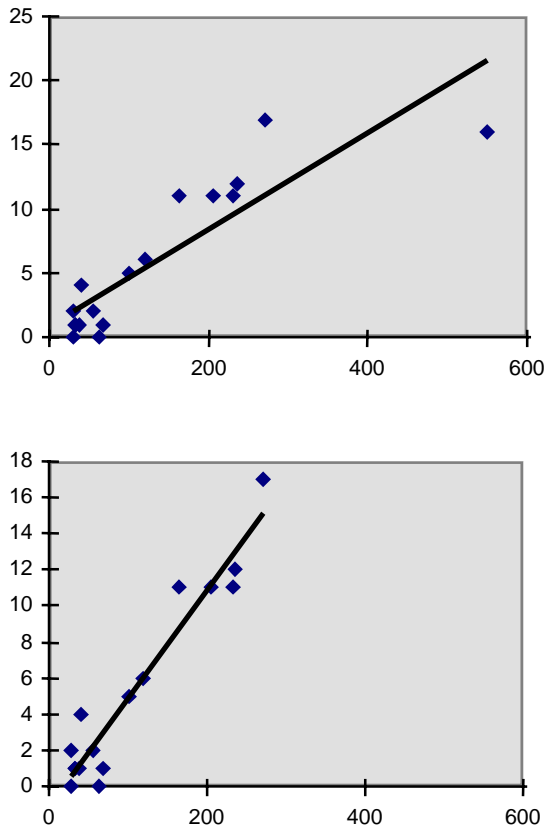


Figure 4. Comparison of regression lines with (upper picture) and without outlier

The pictures show that the regression lines are quite different. The regression line is much steeper when the outliers are excluded. This means, that if the data set includes many outliers, they flatten or steepen the regression line. Therefore, the regression coefficients, which determine the regression line, differ as well.

A scatter plot for the whole data set (Figure 5) depicts the relationship between *effort* and *function points*.

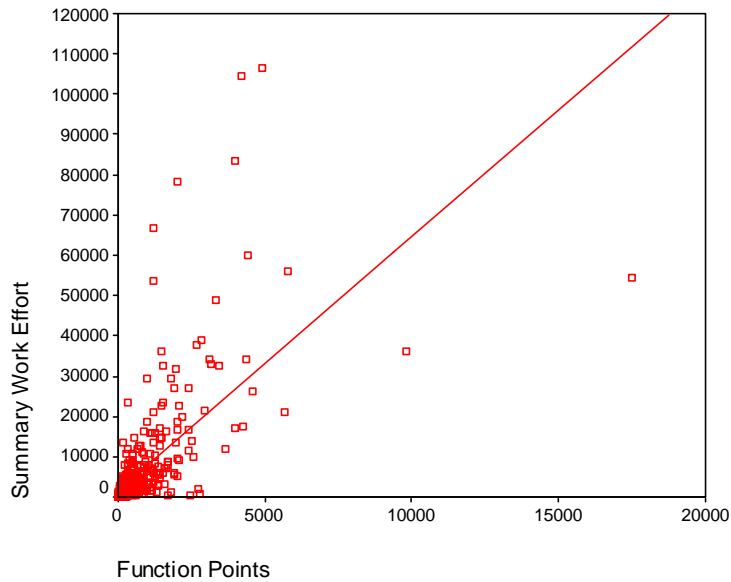


Figure 5. Scatter plot of effort against function points (full data set)

The scatter plot shows that the majority of projects have less than 2500 function points and 10.000 hours of effort. The distribution is quite skewed and there are a lot of extreme outliers. Hence a linear relationship cannot be seen. The only conclusion drawn from this picture is that in most of the cases effort rises as the number of function points increase.

This scatter plot is compared with the one of the subset of projects that doesn't include missing values for the metric *maximum team size* (Figure 6).

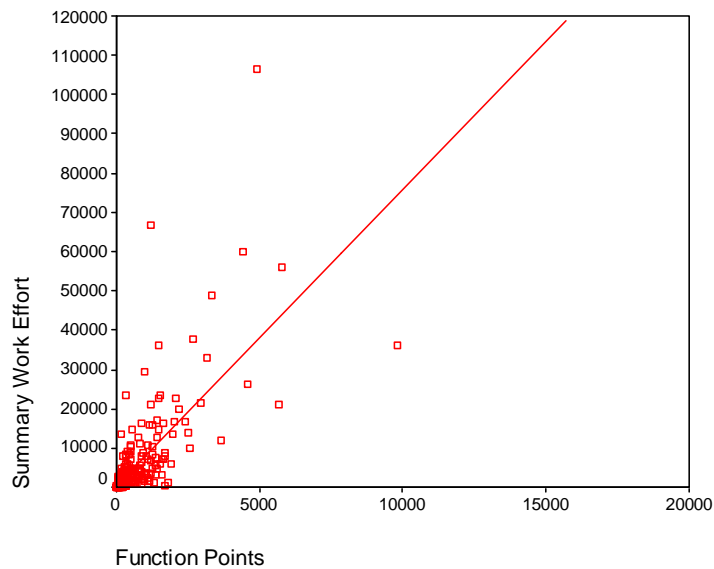


Figure 6. Scatter plot of effort against function points (280 projects)

It can be seen that there is a difference between the two regression lines. The one for the subset of 280 projects is steeper than the one for the full data set. This could explain that *maximum team size* enters the regression model at first, although the metric *function points* shows a slightly higher correlation to *effort*.

4.2.3. ANOVA

ANOVA stands for analysis of variance and is a statistical test that is applied if the mean of a certain dependent variable needs to be compared between more than two groups. It tests whether the means of the different groups are equal or not. It is called analysis of variance because it examines the variability of the sample values [Norusis 98].

We use the one-way ANOVA in this study to test whether certain categories of categorical variables in the ISBSG repository have an impact on *effort* or *PDR*. It is called one-way ANOVA because cases are assigned to different groups based on their values for one variable. If the Null-Hypothesis that the group means are equal is rejected another test can be applied to see which groups differ. We used the Bonferroni test that performs pairwise comparisons between group means.

Now we look at different categorical variables independently to see whether they have any influence on effort.

4.2.3.1. The metric *business area type*

The different business area types are investigated to see whether there is any difference between them and in their mean *effort* and *PDR*. The hypothesis is:

Projects from the *business area type* telecommunications need more or greater effort and are less productive than the other groups. This is expected because of the higher requirements for software for telecommunication systems concerning all kinds of software faults.

Faults can be very expensive as seen in 1996, when the German telecommunication company Telecom introduced a new tariff system. The first day of use, 1st of January, was not categorized as public holiday and, therefore, caused a lot of trouble because the phone bills were too high or wrong. In order to keep their reputation they spent millions of money for free calls for their customers.

The table below shows the categories of business area types in the ISBSG repository, their percentage, and the required effort.

Table 6. Business Area Types and their *effort*

<i>Business Area Type</i>	Percent	Mean <i>effort</i>	Std. dev. of <i>effort</i>	Minimum	Maximum
Missing value	23.7	-	-	5.00	104690
Accounting	6.4	9239	18189	191.00	78472
Banking	21.5	3674	5788	5.00	36046
Engineering	6.7	6008	8970	5.00	34004
Financial (excluding Bank)	12.4	8633	18999	10.00	104690
Insurance	2.4	10698	11890	1174.00	32760
Inventory	4.2	6564	7629	281.00	33028
Legal	4.2	5625	5968	190.00	19699
Logistics	0.4	8608	10433	1230.00	15985

Manufacturing	8.0	4781	10653	97.25	55960
Personnel	3.8	13062	22065	180.00	66600
Research & Development	2.0	2723	5268	17.00	16514
Sales	2.2	3502	2946	391.00	8290
Telecommunications	2.0	2640	2504	544.00	8520

It can be seen that mean *effort* is less for Banking, Manufacturing, Research & Development, Sales and Telecommunications (mean *effort* less than 5000 hours). The highest mean *effort* can be found for the group Personnel. Projects of the business area type Personnel also show the highest variability in *effort*, which means that their standard variation is very high.

The one-way ANOVA shows that there is no difference in mean effort between the business area types. The observed significance level is 0.141. This means the Null-Hypothesis, which says that all means are equal, can not be rejected. Therefore, it can not be concluded that Telecommunication projects required more effort than other projects. These projects even used the lowest amount of *effort* in comparison with the other groups.

The same examinations were performed for the *PDR*, which is a measurement of the productivity.

Table 7. Business Area Types and their *PDR*

<i>Business Area Type</i>	Mean <i>PDR</i>	Std. dev. of <i>PDR</i>	Minimum	Maximum
Missing value	-	-	.02	78.07
Accounting	10.78	9.89	.78	39.31
Banking	13.30	9.99	.02	78.07
Engineering	8.63	11.45	.07	59.43
Financial (excluding Bank)	8.20	6.17	.73	26.56
Insurance	15.34	19.87	3.18	70.30
Inventory	7.85	5.24	1.28	21.94
Legal	7.04	6.07	.91	21.27
Logistics	8.17	6.56	3.52	12.81
Manufacturing	6.37	3.75	1.18	18.35
Personnel	13.20	16.89	.24	54.46
Research & Development	5.56	5.75	.19	15.03
Sales	11.04	7.03	2.59	27.51
Telecommunications	9.29	5.59	1.97	17.60

The mean *PDR* is greater than 10 hours per function point for projects from the groups Accounting, Banking, Insurance, Personnel and Sales, which indicates a low productivity. The lowest value for *PDR* can be found for Research and Development projects. They only needed 5.5 hours per function point, thus have the highest productivity. Insurance projects show the highest variability in mean *PDR*.

The one-way ANOVA rejects the Null-Hypothesis at a significance level of 0.003, which means there is a difference between the groups. With the help of the Bonferroni test it was found that there is only a significant difference between Banking and Manufacturing.

Therefore, the relationship between *effort* and *function points* is expected to be similar for the different groups as well. We only investigated the projects that are outliers when looking at a scatter plot of *effort* against *function points* in order to see whether these projects belong to a certain business area type or not. When outliers are defined as projects with more than 20.000 hours of effort, the relationship between *effort* and *function points* is depicted in Figure 7 with markers for the different *business area types*.

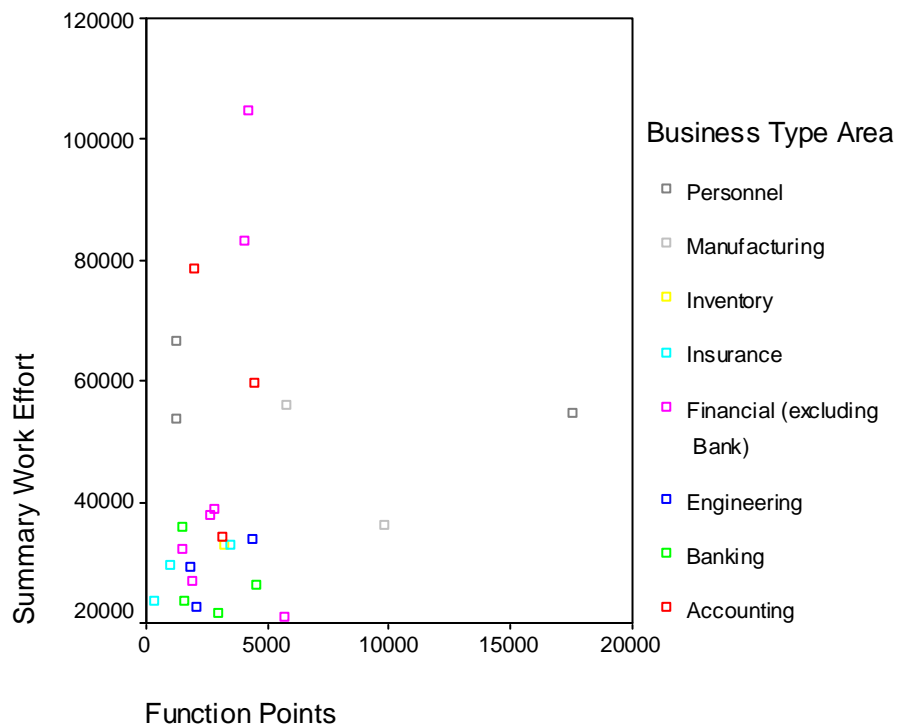


Figure 7. Scatter plot of Effort against Function Point only for outliers with labels for the Business Area type

Telecommunication is not in this outlier-category. There is no business area type that could be highlighted concerning outliers. Eight of the thirteen business area types include outliers. From this point of view, there is no indication for a different behavior for certain business area types, thus this metric is not a cost factor in the ISBSG repository.

4.2.3.2. The metric *development platform*

There is no data stored about the target platform but about the *development platform* in the ISBSG repository. The pie chart presents the distribution of projects by development platform.

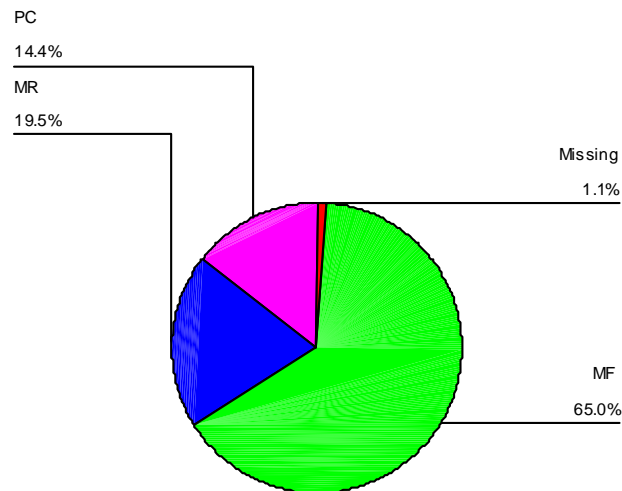


Figure 8. Distribution of projects by development platform

The proportion of projects with mainframe (MF) as development platform is very high. This has to be kept in mind, because all the Megatec projects are developed on midrange (MR) machines or PCs and, as already mentioned, projects developed on mainframes normally require more effort. The mean effort is much higher for mainframe (6689 hours) and midrange (7049 hours) than for PC (2392 hours).

The one-way ANOVA rejects the Null-Hypothesis at a significance level of 0.024, which means that there is a difference between the development platforms. The Bonferroni test resulted a significant difference in mean effort between mainframe and PC.

The *PDR* is also examined. The box plot in Figure 9 shows the distribution of *PDR* that measures the productivity in hours per function point.

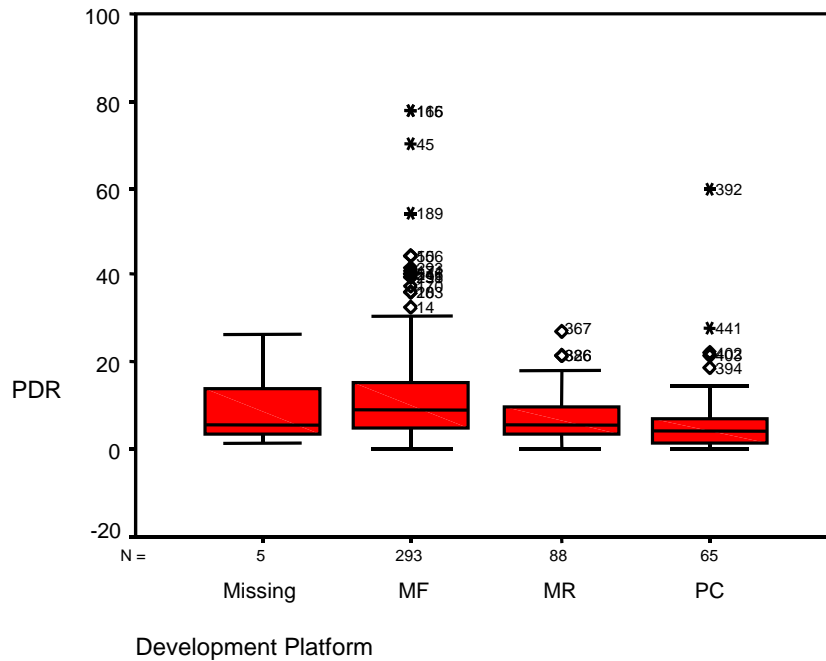


Figure 9. Box plot of PDR for the different development platforms

The *PDR* is 10 hours per function point in mean for mainframe, but only 7 hours for midrange and 6 hours when a PC was used. The range of *PDR* is large for mainframe and PC, which can be seen by the number of outliers.

The one-way ANOVA and the Bonferroni test show a significant difference in mean *PDR* both between mainframe and midrange and between mainframe and PC.

4.2.4. Binary variables

The data set contains several variables that indicate the use of CASE tools, e.g. *upper CASE*, *lower CASE*, and *integrated CASE*, or whether the system is a client/server application. These binary variables could have an influence on *effort* or *PDR*, too. With the help of t-tests, it is possible to determine whether there is a significant difference in a dependent variable, e.g. *PDR* or *effort*.

Firstly, the difference in mean *PDR* for the metric *Client/Server* is investigated. Table 8 presents the values of *PDR* in dependence on the values of the *Client/Server* metric.

Table 8. The values of *PDR* depending on the metric *Client/Server*

<i>PDR</i>	Client/Server?	N	Mean	Std. Deviation	Std. Error Mean
	No	112	11.2686	12.4457	1.1760
	Yes	35	7.1793	5.4939	0.9286

The values of mean *PDR* are quite different. Astonishingly, Client/Server systems required on average 4 hours less to be developed when comparing it with non-Client/Server systems. Usually, Client/Server applications require more development time. Their requirements are normally much higher and more complex than the

requirements for non-Client/Server systems, because the interactions between Clients and Server also need to be realized.

It can also be seen that the values highly deviate from the mean *PDR* for non-Client/Server applications. The t-tests show that there is a significant difference in mean *PDR* between the two groups (p-value: 0.07).

Secondly, the influence of the use of CASE tools is studied. Some of the projects stored in the database used more than one CASE tool, e.g. upper CASE and lower CASE, others used only one kind of CASE tool or none. Table 9 includes the number of projects (N) that used the different CASE tools. When there were two different CASE tools used, an explanation for this might be that different CASE tools were used in different parts of the project.

Table 9. CASE tool metrics and their mean *PDR*

<i>PDR</i>	CASE	N	Mean	Std. Deviation	Std. Error Mean
	upper CASE	55	8.42	8.76	1.18
	lower CASE no code generator	15	15.42	8.96	2.31
	lower CASE with code generator	23	17.96	20.36	4.25
	integrated CASE	15	5.75	6.22	1.61
	upper & lower (with code generator) CASE	10	8.04	7.99	2.53
	upper & lower (no code generator) CASE	8	6.33	6.17	2.18

Projects using the lower CASE with code generator have the highest *PDR*. They also have a high standard deviation, which means the values are widely distributed. The lowest *PDR* can be found when integrated CASE tools were used.

The t-tests compare the mean *PDR* of these groups against each other in order to see whether there is a significant difference in *PDR* to realize between them.

Table 10. Results of the t-tests between the different groups

Comparison	Significant difference
upper CASE v lower CASE with code generator	Yes (0.040)
upper CASE v lower CASE no code generator	Yes (0.008)
upper CASE v integrated CASE	No (0.273)
Lower CASE with code generator v integrated CASE	Yes (0.012)
Lower CASE no code generator v integrated CASE	Yes (0.002)
Lower CASE no code generator v lower CASE with code generator	No (0.653)
Upper & lower (no code generator) CASE v upper & lower (with code generator) CASE	No (0.626)

The answer ‘Yes’ means that a significant difference between the means of the two groups was found. Answer ‘No’ means that no significant difference can be concluded based on this test. The numbers in brackets are the significance value for the t-test.

A significant difference can be seen between upper and lower CASE tools and between lower and integrated CASE tools. No significant difference in results can be found for lower CASE between using a code generator or not. Normally, a code generator should reduce the development effort. When looking at table 9, the opposite can be seen. The mean PDR is always higher when a code generator was used.

When looking at these binary variables, it can be concluded that the ISBSG data is not homogenous and does not follow the common rules for effort predictions for Client/Server systems and for the use of CASE tools.

4.2.5. The metric resource level

The *resource level* is a measure of the kind of recorded effort. The *resource level 1*, for instance, means that development team effort has been recorded. Level 2 means that effort for development team plus the supporting team was recorded and so on:

- 1: development team effort
- 2: development team and people supporting the development team
- 3: the above plus computer operations involvement (e.g. computer operators network admin)
- 4: the above plus effort expended by end user or client.

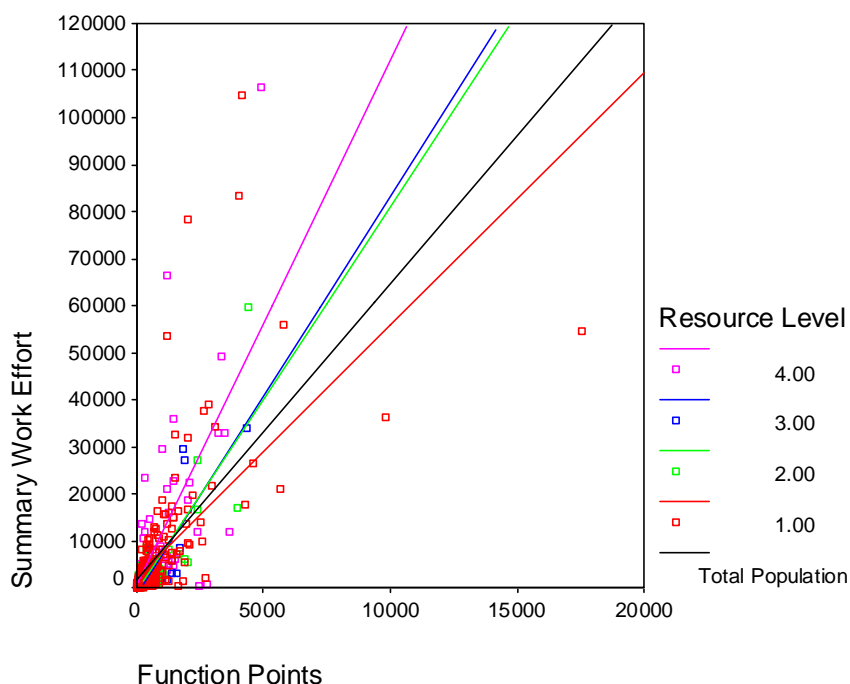


Figure 10. Scatter plot of Effort against Function Point with labels for the different resource levels

The scatter plot presents the least-squares regression lines for each resource level and the total population. The black one is the line that best fits all data points in total and the other four colored ones are best fit for each corresponding resource level. As it can be seen, the red line is closest to the black one. This means that these projects from level 1 represent the full data set properly compared to the three other levels.

The kind of recording *effort* for Megatec projects conforms to level 1 and 2. This is also necessary in order to choose a suitable data subset of ISBSG for the estimation. It is important for the comparison of accuracy, because we want to estimate the effort required for Megatec target projects by using two different data sets as source projects. Therefore, a subset of projects from resource level 1 or 2 was chosen. This is the first step in creating a subset of ISBSG.

The results of the stepwise regression change when only using the subset of projects of *resource level 1* or *2*. The order of entered variables is different: the metric *function points* enters at first, *maximum team size* at second. Furthermore, *project elapsed time* and *project inactive time* enter the model. The value of R^2 , however, is not as high as in the model built on the base of the full data set:

Table 11. Variables entered in the stepwise linear regression (348 projects)

Model	Variables Entered	R square	Method
1	<i>Function Points</i>	0.488	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	<i>Max Team Size</i>	0.601	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	<i>Project Inactive Time</i>	0.636	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
4	<i>Project Elapsed Time</i>	0.685	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

Dependent Variable: *Summary Work Effort*

The reason for this is probably the definition of how to record the required effort. When recording the *effort* for projects of *resource level 3* and *4* it included effort required for administration and even for the end users and clients. It may be that the number of maximum team size also included administrative staff. This is not included when counting the effort for projects of level 1 or 2. Thus, the number of *maximum team size* might be smaller on average for these projects and its influence on *effort* becomes less. This also fortifies the decision to use only a subset of the ISBSG projects, whose *effort* was recorded in the common way.

4.3. The reduced ISBSG data set used for the estimations

4.3.1. The final subset of variables

For the estimates a subset of metrics had to be chosen that is available in both data sets, on one hand, and, furthermore includes only metrics that are collected in a comparable way. This has to be ensured in order to be able to compare the prediction accuracy afterwards. The Megatec data, for instance, contains metrics which indicate the experience of the development team, whereas the ISBSG repository has no corresponding variables for this. The ISBSG repository includes many metrics that have no counterpart in the Megatec data set, thus these metrics were excluded from the analysis, firstly; for instance *total defects delivered* and *DBMS used*. Secondly, many metrics have a lot of missing values, which means the question wasn't answered. Therefore, these metrics couldn't be used, either.

Effort is always used as the dependent variable. *Function points* as a measure of software size was chosen as independent variables, firstly. The second variable is *team size* that is defined as maximum number of people working on the project both for Megatec and ISBSG project data. Thirdly, *development type* was chosen, because of its availability in the two data sets, the fact that Megatec and ISBSG collected this metric in the same way, and also because it is a potential cost factor.

There are 21 different programming languages in the ISBSG database and a lot of them are not very common. This makes it difficult to find analogues, for instance. Therefore, it seemed to be more appropriate to choose the metric *language type* instead of *programming language*. The Megatec data included only *programming language* as metric, but it can easily be mapped into *language type*.

The Megatec data includes only the metric *distributed system* and not *Client/Server*. *Client/Server* was used as a surrogate for *distributed system*, because often client-server systems are distributed systems. This is done because both data sets have only a few variables in common, and in order to be able to look for analogues there have to be as many variables as possible that characterize the projects. It is known that there is usually a difference in the definition of distributed and client-server systems. Finally, the following independent variables were included in the subset:

- *function points*,
- *maximum team size*,
- *development platform*,
- *language type*, and
- *Client/Server*.

4.3.2. The final subset of projects

As mentioned before, projects of *resource level* 1 or 2 are included in the subset. According to the missing values for the metric *team size* or *function points* some projects had to be excluded, but this is a normal step [Fenton and Pfleeger 96]. The remaining subset is called the reduced data set and consists of 225 projects. This is still a large data set compared to both [Walkerden and Jeffery 98] and [Shepperd and Schofield 97].

4.4. Data splitting

This part of the study examines the influence of some metrics on *effort* by splitting the ISBSG data into subsets and using these subsets for the effort estimations. In order to investigate this, the accuracy of the estimates when including the metric in the set of independent variables is compared with the results when excluding the metric from the variable set.

The technique splits the data into subsets in dependence on a certain metric and its values. These subsets of the ISBSG data were used for estimating the Megatec target projects in the way that the subset of ISBSG projects always corresponded to the Megatec target project that had to be estimated for this certain metric.

Firstly, metrics had to be chosen that are potential cost factors. Again, this choice is dependent on the availability of metrics in both data sets, which are only a few.

Furthermore, this splitting technique is only applicable for categorical variables, because we want to use subsets of ISBSG projects that agree with the conditions of the target project for a certain metric, for example target and source projects have the same development type.

We chose the following metrics: *development type*, *development platform* and *Client/Server*. The reasons for this choice and the technique itself are explained in the next sections.

4.4.1. The metric *development type*

Information about the development type could also be found in the Megatec data specification. It documents that all Megatec projects were new developments. In the ISBSG data set there was the corresponding metric *development type*. With the help of the one-way ANOVA and the Bonferroni test it was found that there is a significant difference in mean effort both between enhancements and re-developments and between new development and re-development. Therefore, a difference in estimation accuracy was expected when choosing subsets of ISBSG of a certain development type. When comparing the mean *effort* (table 12), it can be seen that new developments required a higher mean *effort* than re-developments and enhancements.

Table 12. Statistics for *effort* of the different development types

<i>Effort</i>	N	Mean	Std. Deviation	Std. Error	Minimum	Maximum
Enhancement	142	3834.02	10304.49	864.73	5	104690
New Development	287	6720.83	10986.62	648.52	5	83250
Re-Development	21	15811.52	28355.47	6187.67	330	106480

In order to take the information into account, a further subset of 145 ISBSG projects that are all new developments was chosen. The estimations are based on the same kind of *development type* for both the target and the source projects. This kind of selection can improve the estimates, because the development type matches and differences between the three development types were realized from the statistics. It is discussed in section 5, whether there is a significant difference in estimation results obtained from using only projects that are new developments instead of projects that are of mixed development types as base for the estimates.

4.4.2. The metric *development platform*

The second metric we examined is *development platform*. It was mentioned before, that there is a significant difference in mean *effort* between mainframe and PC. There is also a significant difference in mean *PDR* between mainframe and midrange and between mainframe and PC. This led us to investigate whether there is a difference in estimation accuracy between using projects with mixed development platforms and projects that have the same development platform as the target project.

The *development platform* for the Megatec projects was either PC or Midrange (HP 9000 or Sun). The adjustment of Megatec projects was done by splitting the ISBSG data into subsets of projects with a certain development platform.

Figure 11 depicts this kind of data splitting as a binary tree, which is comparable to the regression trees used in the study of Briand et al. [98]. There are only 79 of the 225 projects left, because of different development platforms or missing values.

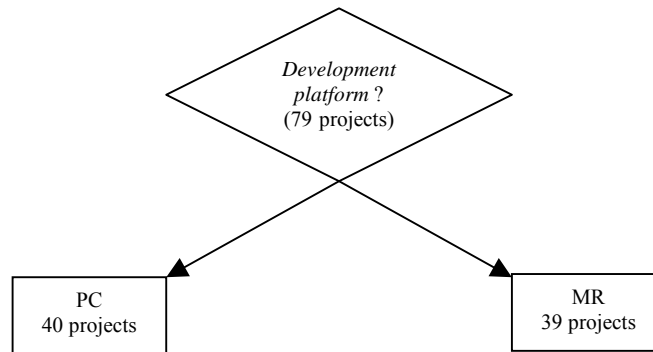


Figure 11. Classification tree based on the variable *development platform*

If the target project has a PC as *development platform*, the subset of the 40 ISBSG projects is chosen as base for the estimate. Otherwise, the subset of projects that were developed on midrange is chosen. This procedure is done by hand and is called selecting, because the matching subset is selected before the actual estimation is started. The estimates were obtained from applying regression and ACE in the following manner:

- Regression was applied separately on the projects belonging to the two subsets as well as on the 79 projects altogether.
- When ACE was applied, the subset of 79 projects is used for the ranking, whilst the variable *development platform* was still included in the variable set. In a second run, *development platform* was excluded from the variable set and in dependence on the *development type* of the target project the matching subset of ISBSG was chosen as a base for the estimation of ACE.

The first step, when using the 79 projects for estimating effort, is called ranking, whereas the second step, when using the two different subsets, is called selecting.

4.4.3. The metric *Client/Server*

The same was done for the variable *Client/Server*. This is reasoned by the expectation that usually there is a difference in required effort between projects that develop client-server software and other projects. In the ISBSG data there are 62 projects with a certain answer to this question. The data was split into two subsets as done before for *development platform*.

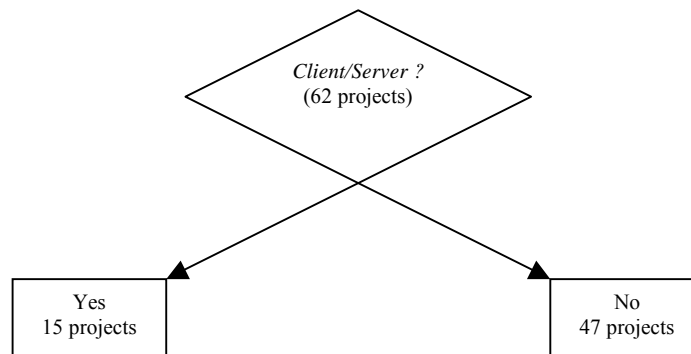


Figure 12. Classification tree based on the variable *Client/Server*

It is interesting to look at the *PDR* of the projects within these two subsets. Software projects that developed client-server software required less time per function point and, hence, are more productive than software projects that were developing none-client-server software. This stands in conflict with normal expectations; more time is needed to develop client-server systems, because there are much more requirements to fulfill for these systems. But this is probably due to the heterogeneity of the ISBSG data.

4.5. Ordinary least square regression

The research question about the existence of a relationship between effort and certain other variables can be investigated using correlation analysis. Linear regression is a popular method for expressing an association as a linear formula, but this does not mean that the determined formula will fit the data very well. Regression is based on a scatter plot, where each pair of attributes (x_i, y_i) corresponds to one data point when looking at a relationship between two variables. The line of best fit among the points [Fenton and Pfleeger 96] is determined by the regression. It is called the least-squares regression line and is characterized by having the smallest sum of squared vertical distances between the data points and the line.

Linear regression can express the relationship between two (simple linear regression) or more than two (multiple linear regression) variables in a linear formula that determines the regression line:

$$\text{effort} = a + b * x_1 \quad (\text{simple linear regression})$$

$$\text{effort} = a + b * x_1 + c * x_2 + d * x_3 + \dots \quad (\text{multiple linear regression})$$

Effort is the dependent variable. The x_i are the independent variables. These variables are used for building the regression model, which calculates the coefficients and the intercept a . The intercept and the coefficients b , c , and so on determine the regression line in either

the two-dimensional space (simple linear regression) or the n-dimensional space (multiple linear regression).

When using regression, normal distribution of the data is not required. If the distribution of the data is very skewed, it is a common method in software engineering to apply regression to transform the data into a scale where the measurements fit more closely to the normal distribution [Fenton and Pfleeger 96]. The logarithmic scale is chosen very often for transforming the data. When the original data is not normally distributed, the ln-transformed data is [Fenton and Pfleeger 96]. Hence, the regression model is exponential. The transformation is also useful to reduce some variance in the data. It has a smoothing effect on the data [Conte et al. 86].

It was already mentioned that the ISBSG data is not normally distributed. Therefore the data of the considered variables has been ln-transformed. Figure 13 shows the relationship between $\ln(\text{effort})$ and $\ln(\text{function points})$ for projects from *resource level 1* and 2.

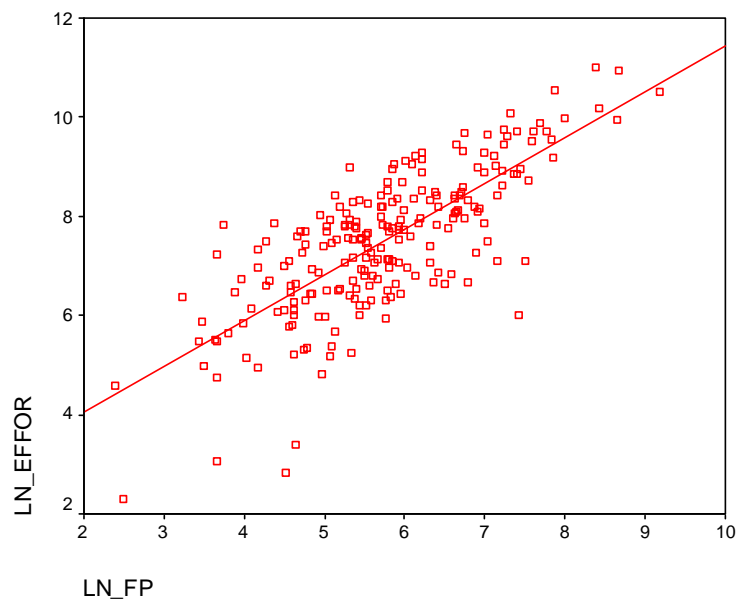


Figure 13. Scatter plot of effort against function points when using the reduced data set and applying the ln-transformation

As can be seen on the scatter plot, the data points are quite close to the regression line apart from a few outliers indicating that a relationship exists.

Furthermore, the linear model of the ISBSG data showed heteroscedasticity, which is a violation when applying linear regression [Briand et al. 98]. Heteroscedasticity was tested with the help of a scatter plot of residuals against predicted effort [Fenton and Pfleeger 96]. Performing the ln-transformation also addresses heteroscedasticity in regression analysis.

If linear regression is applied on the transformed data, the investigated relationship is an exponential one and is determined by one of the following formulas:

$$\ln(\text{effort}) = \ln(a) + b \cdot \ln(x) \text{ for a simple regression or}$$

$$\ln(\text{effort}) = \ln(a) + b \cdot \ln(x_1) + c \cdot \ln(x_2) + d \cdot \ln(x_3) + \dots \text{ for a multiple regression}$$

In order to determine the coefficients for the regression equation, a regression model had to be built. That means dependent and independent variables had to be chosen first. As variables used for this study *function points* and *maximum team size* were chosen as independent variables and *effort* is always the dependent variable. The reason for this is that they entered the stepwise regression first, which means they are mostly correlated to *effort*. Furthermore, both variables are available in the Megatec and the ISBSG data set. After applying the regression the coefficients can be used for predicting the effort. The value for $\ln(\text{effort})$ that we got from the model, can easily be transformed into the predicted effort itself by applying the inverse logarithms (exp.).

For this study different regression models were built to be able to compare the results with the one obtained from ACE. These models use different data subsets that are explained in section 4.4. The results of the several regression model estimates are documented in chapter 5.

4.6. ACE

A prototype of the Analogical and Algorithmic Cost Estimator (ACE), has been developed as a means to explore the benefits of analogy-based estimation. ACE estimates *effort* for a target project by searching through a database of metrics for completed projects and selecting the completed project that it judges most similar to the target project [Walkerden and Jeffery 98].

How does ACE find similar projects?

The tool uses a certain ranking algorithm when searching for similar projects. One part of the ranking algorithm is the calculation of the difference between the target project and each source project for each metric and each source project. According to this difference the completed projects get ranked and this is done for all included variables.

The completed project with the lowest difference is ranked 1 on that metric; the project with the next lowest difference is ranked 2, and so on. If two completed projects differ from the target project by the same amount for a particular metric, they are allocated the same rank, and the rank of the project with the next lowest rank is adjusted accordingly. For example, if the target project has a *maximum team size* (MTS) of 4 people, and two completed projects also have a MTS of 4, then they are both assigned ranks 1. The next most similar project has a MTS of 5 people, and is assigned rank 3.

ACE calculates the average rank of each completed project over the set of independent variables. The project with the lowest average rank is selected as the first analogue for the target project, the one with second lowest average rank is the second analogue and so on. Calculating the average rank standardizes the contribution of each search metric to the

final ranking [Walkerden and Jeffery 98]. If there is more than one project ranked as first or second similar the more accurate one (less ARE) was chosen.

How does ACE handle categorical variables?

The categorical metrics, such as *language type*, are handled equivalently: all projects with the same categorical value as the target project are assigned rank 1; all other projects are assigned the next rank, for example rank 4, if 3 completed projects used the same language type as the target project. The larger the data set than higher are the ranks for the projects that do not match this category. Therefore, the ISBSG data the ranks will be quite high and influence the average rank greatly. It is also possible to exclude the categorical variables, but this leads to a lack of variables, because a lot of project characteristics are categorical ones.

How is a missing value handled?

If there is a missing value, for instance for the function point metric, the project was excluded from the data set in general, because these data are essential for the estimates. The missing values for categorical variables were handled in a different way. A dummy value was introduced. Hence, when looking for analogues in the ranking process, no similarity can be found for this variable.

How many analogues should be selected for the estimation – only one or more?

In the former study, which also used the Megatec database, only one analogue was used for the estimates [Walkerden and Jeffery 98]. Because of the indication that two analogues could improve the estimation quality, this study uses two analogues for the prediction as well. The predicted effort is determined as the unweighted average of both. Shepperd et al. [96] recommend a weighted average, which means that the first analogue is weighted more than the second one as an example. This is not applied here, because of the almost equivalent accuracy reached by the weighting before [Shepperd and Schofield 96].

What kind of adjustment should be chosen in order to respond the differences between the source analogue(s) and the target analogue?

Size adjustment is a response to differences between the most similar project(s) and the target project. The effort value of the completed analogue project(s) is adjusted for the target project to take into account the size difference between target and source project(s). As applied by Walkerden and Jeffery [98], linear size adjustment is used in the current study, because it improved the accuracy of results when compared to not using size adjustment. The linear size adjustment assumes a linear relationship between *effort* and *function points*. This means, if the software size is low, it is assumed that the *effort* is low and vice versa. This rule does not seem to be always appropriate, especially when reusing or re-developing software. In this study the use of size adjustment is also investigated in order to see which method better fits the ISBSG repository.

What has to be done for using the tool for estimates?

The steps for using ACE can be summarized as follows:

1. Project metrics – Select the project metrics that are to be considered in the estimation process.
2. Target project – Given estimated project metrics for a new project.
3. Source projects– Given a set of n software projects from a historical database. Data is available for all of the project metrics.
4. Determine the difference – For each of the project metrics, compare the past projects with the target project metrics and calculate the difference.
5. Rank the source projects – The project with the lowest difference is ranked 1 ... and so on. The ranking gives duplicate numbers for the same rank. However, the presence of duplicate numbers affects the ranks of subsequent numbers. This is applied consistently for all project metrics.
6. Calculate the average rank – Determine the average rank across the project metrics used for each project. The lowest average rank indicates the source analogue as the most similar project and so on.
7. Chose a number of analogues and estimate – Using the effort value from the selected analogues and predict the effort for the target project based on this. If more than one analogue is used, the average effort of these projects is determined in order to predict the effort for the target project.
8. *Optional:* Adjust - an adjustment can be made according to the formula,

$$Effort_{TARGET} = \frac{Effort_{ANALOGUE}}{FP_{ANALOGUE}} \times FP_{TARGET}$$

When using two analogues for the estimation the effort with size adjustment is calculated with:

$$Effort_{TARGET} = \left(\frac{Effort_{ANALOGUE1}}{FP_{ANALOGUE1}} + \frac{Effort_{ANALOGUE2}}{FP_{ANALOGUE2}} \right) \times \frac{FP_{TARGET}}{2}$$

That means at first the average of the *PDR* of both projects is calculated and than multiplied by the function points of the target project.

An example will make it more understandable:

The table on the following page is a copy of an estimation template. It is divided into two parts, the upper one, with data about the target project and the lower part with the data of the source projects. Here there are 19 source projects with their effort, number of function points, team size and the language type used for these projects. The columns next to the actual values contain the difference between target and source project for each variable.

In case of a categorical variable it only says whether it matches (diff=1) or not (diff=2). The ranking is done according to these differences and finally the average rank is calculated. The lowest average rank indicates the most similar project and so on.

The size adjustment for this example is calculated as follows:

For one analogue:

$$Effort_{TARGET} = \frac{1113.50}{153.01} * 143.00$$

For two analogues:

$$Effort_{TARGET} = \left(\frac{1113.50}{153.01} + \frac{447.50}{150.40} \right) * \frac{143.00}{2}$$

As can be seen in the example, the ranking in case of categorical variables strongly depends on the distribution of values of source projects. As there are five projects within the same category (4 stands for 4GL), the remaining projects are ranked as 6th. If there are more projects within the same category, the ranking for the categorical variables can get very high. The ISBSG data set is quite large and not homogenous, which probably influences the ranking of projects a lot.

Target
Project ?

143.00

4

4

Real Project ID	Effort (Hours)	Function Points	diff fp	rank fp	Maximum Team Size	diff mts	rank mts	Language Type	diff lt	rank lt	Average Rank
1	3532.00	345.00	202	10.00	6	2	8.00	4	1	1.00	6.3
2	6548.00	421.00	278	13.00	6	2	8.00	4	1	1.00	7.3
3	4545.00	2997.00	2854	18.00	3	1	4.00	4	1	1.00	7.6
4	6475.00	413.00	270	12.00	5	1	4.00	3	2	6.00	7.3
5	5354.00	391.00	248	11.00	5	1	4.00	3	2	6.00	7.0
6	5453.00	179.00	36	2.00	2	2	8.00	3	2	6.00	5.3
7	5145.00	167.00	24	1.00	4	0	1.00	3	2	6.00	2.6
8	5643.00	790.00	647	17.00	2	2	8.00	3	2	6.00	10.3
9	7678.00	89.00	54	4.00	7	3	16.00	3	2	6.00	8.6
10	193.50	302.00	159	8.00	2	2	8.00	4	1	1.00	5.6
11	1214.00	200.00	57	6.00	4	0	1.00	4	1	1.00	2.6
12	8985.00	199.00	56	5.00	4	0	1.00	3	2	6.00	4.0
13	3122.00	189.00	46	3.00	1	3	16.00	3	2	6.00	8.3
14	6577.00	578.00	435	15.00	2	2	8.00	3	2	6.00	9.6
15	657.00	532.00	389	14.00	1	3	16.00	3	2	6.00	12.0
16	432.00	342.00	199	9.00	2	2	8.00	3	2	6.00	7.6
17	4321.00	751.00	608	16.00	3	1	4.00	3	2	6.00	8.6
18	24212.00	3311.00	3168	19.00	10	6	19.00	3	2	6.00	14.6
19	534.00	206.00	63	7.00	2	2	8.00	3	2	6.00	7.0

4.7. Estimations in this study

In order to compare the accuracy of Megatec and ISBSG estimates different steps are performed. Generally, one of the 19 Megatec projects was chosen as a target project. Altogether, there are 19 different estimates to regard for each step. Figure 14 depicts the main steps:

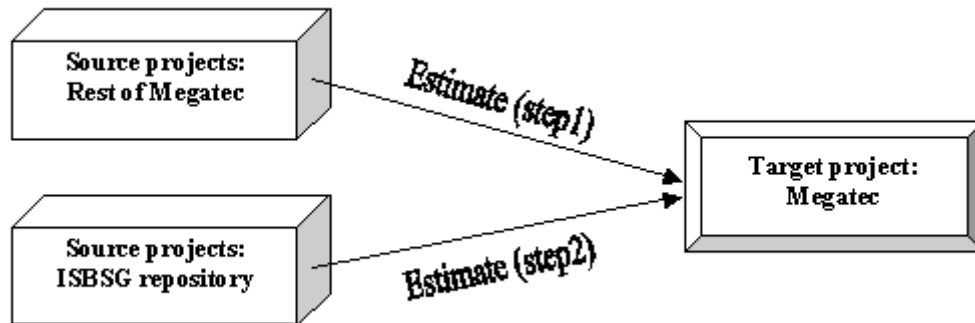


Figure 14. The two steps in the research process

Firstly, each target project was estimated with the Megatec data. This means the remaining 18 of the 19 projects are the basis for each case. Regression and ACE are utilized for doing the estimates.

Secondly, each target project was estimated with the ISBSG data. This means the 225 projects of the reduced data set are the basis for the 19 estimates. Again, regression and ACE are utilized for doing the estimates.

The estimation results of the two steps are used to compare ISBSG and Megatec based estimates.

Furthermore, certain variables are investigated independently. As mentioned before, these variables are *development type*, *Client/Server* and *development platform*. The utilized methods for the ranking and selecting were explained before. Regression and ACE are applied separately on the different data subsets.

The following table summarizes all the methods that are compared. It can be seen that this is basically a comparison of the performance of regression and ACE. Furthermore, the influence of the size adjustment and the number of analogues are studied.

Table 13. Estimation Methods to Compare

Estimation Method	Adjustment Method	No. of Analogues
Regression	None	-
ACE	None	1
		2
	Linear Size Adjustment	1
		2

4.8. Data Analysis

4.8.1. Evaluation Criteria

For the evaluation of the cost estimation models the same criteria Walkerden and Jeffery [98] used were chosen, because these criteria are widely used in the literature, making comparisons easier [Conte et al. 86]:

1. Absolute relative error as a percentage of the actual effort for a project, is defined by:

$$ARE = \frac{100 | (Actual\ Effort - Estimated\ Effort) |}{Actual\ Effort}$$

2. The mean ARE:

$$\overline{ARE} = \frac{1}{N} \sum_i ARE_i$$

3. And the proportion of predictions (PRED). This measure is often used in the literature and is a proportion of a given level of accuracy:

$$PRED(l) = \frac{k}{N}$$

N is the total number of observations, and k the number of observations with an ARE less or equal than l. A common value for l is 0.25, which is used for this study as well.

4.8.2. Statistical tests

For the analysis of the results several statistical tests are used. The results are the predicted effort obtained from regression and ACE for the 19 target projects. Their performance is measured in ARE and different comparisons are made.

With the help of t-tests it can be determined whether there is a significant difference in means between two groups [Norusis 98]. For instance, it can be tested whether there is a significant difference in mean ARE between using Megatec data and the ISBSG data as a base for the estimates. The two groups mentioned are the results for the 19 cases obtained from Megatec, on one hand, and ISBSG based predictions, on the other hand. It is also possible to explore whether there is a significant difference between the two cost models or between the different techniques for ACE as well. For example, it can be tested whether the performance of ACE when using one analogue with size adjustment is significantly different to using no size adjustment.

The t-tests assume that the results are normally distributed. This was tested with the Kolmogorov-Smirnov test. When the data shows normal distribution, the paired t-test was used to determine whether there is a difference in accuracy:

This paired t-test for matched groups is applied when the dependent measure, e.g. *number of defects*, is examined under two different conditions (before and after training of the personnel) [Fenton and Pfleeger 96]. When two sets of data that are related have to be compared, the paired t-test can be applied; for instance, when it has to be assessed whether the *number of defects* are significantly different for several persons before and after training.

In this study the paired t-test is applied to determine whether there is a significant difference in mean ARE between the estimates based on Megatec and the one based on ISBSG. It is also used to test the difference in accuracy of ACE between using one analogue and two analogues, size adjustment and no size adjustment, and between regression and ACE in general.

For this study the t-tests and the Kolmogorov-Smirnov test were utilized for the data analysis. The confidence interval is always 95 %.

5. Results

This section presents the main results starting with the ones obtained from estimates based on Megatec. Secondly, the ISBSG based estimates results are documented. They are compared with the results obtained from Megatec afterwards. Finally, the results of separately examining several variables are presented.

5.1. Estimates based on Megatec

5.1.1. Ordinary least squares regression

The ln-transformed data was used for building the 19 different regression models. The calculated coefficients (a, c_max, c_fp) are documented in the table below, as well as predicted effort, actual measured effort, and *PDR* in hours per function point. The regression model was based on the variables *maximum team size* and *function points*. Hence, the regression equation is:

$$\ln(\text{effort}) = a + c_max * \ln(\text{maximum team size}) + c_fp * \ln(\text{function points})$$

Table 14. Regression coefficients of Megatec data based estimations

ID	a (intercept)	c_max (coefficient for max team size)	c_fp (coefficient for function points)	ln_effort	predicted effort	actual effort	ARE in %	PDR
1	2.920	.471	.635	8.240	3791.033	3777.00	0.37	3.28
2	2.800	.455	.663	8.636	5629.825	4388.50	28.29	2.26
3	2.937	.486	.625	7.024	1123.523	1647.25	31.79	5.59
4	2.923	.463	.635	7.525	1853.378	2069.75	10.45	4.77
5	2.892	.441	.644	7.304	1486.053	1946.80	23.67	6.20
6	2.799	.537	.633	6.258	521.943	1500.00	65.20	11.44
7	2.838	.434	.655	6.735	841.033	1113.50	24.47	7.28
8	2.695	.434	.680	5.487	241.543	362.20	33.31	9.29
9	2.753	.378	.682	6.573	715.306	921.00	22.33	10.01
10	3.157	.458	.603	6.186	485.834	193.50	151.08	2.16
11	2.813	.450	.664	7.666	2135.140	1218.00	75.30	2.09
12	2.923	.473	.634	7.227	1375.527	1317.50	4.40	4.18

13	2.949	.604	.597	5.747	313.367	529.40	40.81	4.88
14	2.921	.474	.634	6.508	670.430	691.25	3.01	4.05
15	2.893	.504	.631	5.488	241.772	290.75	16.85	4.76
16	2.945	.443	.640	6.461	639.429	447.50	42.89	2.98
17	3.107	.505	.606	6.788	886.994	261.50	239.19	1.50
18	3.227	.454	.576	8.937	7610.331	13904.75	45.27	4.23
19	2.963	.458	.632	6.292	539.965	414.75	30.19	3.54

Two projects are totally overestimated, project 10 with more than 150% and project 17 with more than 200%. The reason for these high AREs could lie in the numbers of *PDR* and the intercept:

- The average *PDR* for the 19 projects is about 5 hours per function point. Project 17 has a very high productivity, it used only 1.5 hours per function point. Project 10 required 2.2 hours per function point, which is still low. It is comparable to project 11, which also has a high ARE with 75%.
- The intercept is a measure of effort that is basically used before actually starting to develop the software system and, hence, does not include the effort for developing function points or lines of code. A high value indicates that the basic costs are very high. When looking at projects 10 and 17, the calculated intercept is higher than for all other projects. Regression calculates the intercept by using the remaining 18 of the 19 projects to determine the intercept. Obviously, the determined values don't fit the conditions of these projects very well, which is also reasoned by their low values for *PDR* and actual *effort*. A very high intercept was calculated for project 18, for instance, but its actual effort is very high, thus this project is only overestimated by 45%.

5.1.2. The analogy-based estimates

ACE always looks for similar projects using the 18 remaining projects as source projects for estimating one target project. The following table presents the results obtained from ACE. The second column shows the actual effort for each project, the following ones the predicted effort when applying the four different estimation techniques of ACE.

Table 15. Predicted effort when using Megatec data and ACE with its four estimation techniques

ID	Actual Effort	One analogue without SA ⁴	ARE in %	One analogue with SA	ARE in %	Two analogues without SA	ARE in %	Two analogues with SA	ARE in %
1	3777.00	4388.50	16.19	2601.35	31.13	2803.25	25.78	2502.65	33.74
2	4388.50	3777.00	13.93	6371.83	45.19	2497.50	43.09	5213.66	18.80
3	1647.25	1317.50	20.02	1230.31	25.31	789.50	52.07	836.58	49.21
4	2069.75	1946.80	5.94	2693.79	30.15	1530.15	26.07	2926.61	41.40
5	1946.80	2069.75	6.32	1495.82	23.16	1591.63	18.24	1889.58	2.94
6	1500.00	362.20	75.85	1217.27	18.85	796.00	46.93	785.72	47.62
7	1113.50	1500.00	34.71	1751.09	57.26	1723.40	54.77	1350.24	21.26
8	362.20	1500.00	314.14	446.33	23.23	957.38	164.32	292.14	19.34
9	921.00	1113.50	20.90	669.80	27.27	1530.15	66.14	620.44	32.63
10	193.50	414.75	114.34	317.33	64.00	431.13	122.80	292.13	50.97

⁴ The abbreviation SA stands for size adjustment.

11	1218.00	784.00	35.63	790.58	35.09	1482.38	21.71	2852.45	134.19
12	1317.50	261.50	80.15	474.23	64.01	1104.15	16.19	1215.51	7.74
13	529.40	414.75	21.66	384.05	27.46	431.13	18.56	353.54	33.22
14	691.25	447.50	35.26	507.63	26.56	431.13	37.63	555.57	19.63
15	290.75	529.40	82.08	297.93	2.47	472.08	62.36	257.03	11.60
16	447.50	691.25	54.47	609.37	36.17	553.00	23.58	570.69	27.53
17	261.50	691.25	164.34	704.58	169.44	569.38	117.73	611.00	133.65
18	13904.75	1317.50	90.52	13746.08	1.14	1693.63	87.82	14716.39	5.84
19	414.75	447.50	7.90	348.87	15.89	488.45	17.77	460.30	10.98

Looking at the results in columns 3 to 6, which are obtained by using only one analogue. If no size adjustment is applied (column 3) there are three projects overestimated by more than 100%. When utilizing the size adjustment (column 5), the ARE decreases a lot for project 8, and there is only one project left with more than 100% ARE. Project 18 is better estimated with size adjustment compared to not utilizing size adjustment. The ARE improved by almost 90% for one analogue and for two analogues by more than 80%. The actual effort of project 18 is very high. In the Megatec data there are no projects found with a similar amount of *effort*. Therefore, the prediction without size adjustment is poor, but improves a lot when utilizing size adjustment. The size adjustment improves the accuracy of the predictions in some cases, an exception is made by project 17.

It has to be mentioned that project 17 is overestimated in all cases that were performed during this study. The reason might be the very high number of *function points* compared to *effort*, or its very high productivity. Therefore, the main results would become better when this project is excluded, but in the current study this it is included. It is important to know that it is generally difficult to estimate this project, because it is an extreme outlier.

When using two analogues (columns 7 to 10), the ARE sometimes changes. The choice of analogues is likely to be incorrect once, but it is very unlikely to happen twice. Therefore, the average effort of two analogues should predict *effort* better. This is the case for some of the projects for instance number 8. On the other hand, there are projects that behave in the opposite way. The same can be said about the utilizing size adjustment. There is no general improvement recognizable.

Project 8 has a low number of *function points* and, hence, size adjustment adjusts the predicted effort better to the actual measured *effort*. The ARE for project 11 rapidly deteriorates when using size adjustment, which is probably due to the high number of *function points*. A high number is an indication of higher *effort*, thus the predicted effort increases much more when utilizing linear size adjustment for this project.

The other project estimates show no large differences in ARE when comparing the four techniques. In order to summarize the results of estimations based on Megatec, table 16 compares the mean ARE and the proportion of predictions (PRED) at a level of 25% for the different estimation methods.

Table 16. Mean Absolute Relative Errors and the PRED of the Estimates based on Megatec projects

Estimation Method	Adjustment Method	No. of Analogues	Mean ARE in %	PRED(0.25) in %
Regression	-	-	47	42
ACE	None	1	63	42
		2	54	32
	Linear Size Adjustment	1	38	32
		2	37	47

The PRED is highest for ACE with two analogues and size adjustment, which means 9 of the 19 projects were estimated with less than 25% ARE. This estimation method also performs best in mean ARE in percent. Conte et al. [86] concludes that a good estimation model should have PRED(0.25) of more than 75 %. This is a really high percentage, which was not obtained by either [Briand et al. 98] or [Walkerden and Jeffery 98].

There is a perceivable difference between applying size adjustment and not applying, and also between using one or two analogues. Regression performed better than ACE using no size adjustment, but worse than ACE without size adjustment.

Statistically there is no significant difference in mean ARE between the different techniques according to the t-tests. The results are shown in the table below.

Table 17. Results of the t-tests to compare the different techniques

	Regression	ACE-1 no SA ⁵	ACE-1 with SA	ACE-2 no SA	ACE-2 with SA
ACE-1 no SA	0.354	-	0.158	0.348	-
ACE-1 with SA	0.317	0.158	-	-	0.880
ACE-2 no SA	0.547	0.348	-	-	0.166
ACE-2 with SA	0.299	-	0.880	0.166	-

5.2. Estimates based on ISBSG

The regression model is the same one for the 19 target projects, because the 225 ISBSG projects are always used for the prediction. The regression equation calculated by SPSS is the following one:

$$\ln(\text{effort}) = 2.228 + 0.655 * \ln(\text{maximum team size}) + 0.748 * \ln(\text{function points})$$

This formula is used for the predictions documented in column 3. Furthermore, the results obtained from ACE are presented in table 18.

⁵ The abbreviations after ACE stand for the number of analogues (1 or 2) and the use of size adjustment.

Table 18. Predicted effort when using ISBSG data and applying regression and ACE with its four techniques for the estimates

ID	Actual Effort	Regression	ARE in %	One analogue without SA	ARE in %	One analogue with SA	ARE in %	Two analogues without SA	ARE in %	Two analogues with SA	ARE in %
1	3777.00	5852.82	55	5624.00	49	4713.36	25	11421.50	202	9433.17	150
2	4388.50	8654.63	97	16357.00	273	19342.25	341	16788.00	283	21609.22	392
3	1647.25	1339.67	19	3570.00	117	3469.85	111	2204.50	34	2168.41	32
4	2069.75	2502.56	21	2263.00	9	2634.00	27	3670.50	77	4719.18	128
5	1946.80	1962.86	1	5078.00	161	4917.51	153	3805.50	95	3748.94	93
6	1500.00	560.63	63	616.00	59	651.12	57	796.00	47	785.72	48
7	1113.50	991.12	11	2216.00	99	2216.14	99	1985.00	78	1936.40	74
8	362.20	226.41	37	10.00	97	32.50	91	53.63	85	188.65	48
9	921.00	977.69	6	1544.00	68	2220.47	141	2317.00	152	2111.65	129
10	193.50	422.17	118	17.00	91	16.76	91	100.00	48	88.85	54
11	1218.00	2698.71	122	784.00	36	790.58	35	2179.00	79	2091.81	72
12	1317.50	1702.45	29	1249.00	5	1382.10	5	3163.50	140	3162.42	140
13	529.40	309.26	42	440.00	17	582.57	10	708.00	34	672.45	27
14	691.25	682.84	1	976.00	41	1197.95	73	822.00	19	973.87	41
15	290.75	201.17	31	440.00	51	327.85	13	756.50	160	676.12	133
16	447.50	621.39	39	976.00	118	1056.05	136	822.00	84	858.51	92
17	261.50	903.37	245	668.00	155	764.24	192	1211.00	363	1323.54	406
18	13904.75	17927.10	29	21625.00	56	24169.46	74	15729.50	13	18414.50	32
19	414.75	515.80	24	616.00	49	582.47	40	796.00	92	702.87	69

Firstly, look at the regression results: Projects 10, 11 and 17 have an ARE of more than 100%. Again, these projects along with project 2, are the projects with much higher productivity compared to the ISBSG projects. This strong deviation might be the reason for the overestimation. The regression model does not fit their conditions very well.

Secondly, look at the ACE estimates. There are several estimates with an ARE of more than 100%. The ranking algorithm and the definition of similarity might be reasons for this. The similarity is defined by using five variables. Two of them are continuous and the other three are categorical. There are 225 projects and it is quite likely to find matching projects concerning the categorical values for instance projects that are using the same language type and the same development platform.

Hence, the decision about the most similar project is mainly based on the two ratio scaled metrics. It is likely to find the same number of *maximum team size* within a large data set. Therefore, the ranking is even more dependent on *function points*.

Weighting the ranks for the different variables could be a change in the definition of similarity. That means, averaging of ranks (the right formula) is replaced by a weighted average:

$$rank_{weighted} = \frac{\sum_{i=1}^n w_i * rank(i)}{n} \qquad rank_{average} = \frac{\sum_{i=1}^n rank(i)}{n}$$

The variable w_i indicates the weight of each of the n variables. Whether this improves the performance and which weights should be used have to be investigated at first.

The size adjustment tries to adjust the effort, but the formula is also based on the *PDR*. It uses the *PDR* of the analogue project for adjusting the predicted effort. If the *PDR* of the analogue deviates from the one of the target project, size adjustment deteriorates the ARE. When there is no linear relationship between *effort* and *size* the adjustment has no positive influence on the prediction, either. This is the case for project 2 both when using one and two analogues. The first and second, analogue that are ranked by ACE, required 10 and 12 hours per function point, which is much higher than the actual value of 2.3 hours for project 2. It is the same with project 9 or 16, which have a low *PDR*, but the first ranked project has about half of their *PDR* and, therefore, the adjusted effort again has a much higher ARE.

Generally, the mean *PDR* of the ISBSG projects is higher than the one for Megatec projects:

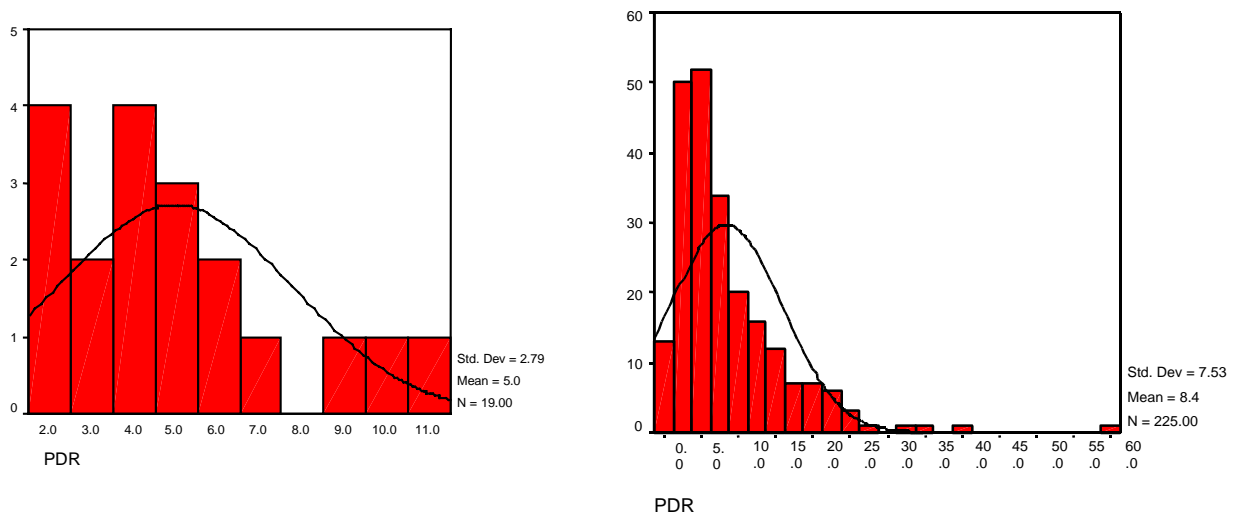


Figure 15. Histograms of *PDR* (in hours per function point) for Megatec (left hand) and ISBSG projects

ISBSG projects required 3.4 hours more per function when comparing the mean *PDR*. This indicates that predictions based on ISBSG are more likely to be overestimated.

In order to address the difference in mean *PDR*, a different adjustment method was explored in this study. Assumed that it is known how many function points the team develops on average, a productivity adjustment can be applied on the effort prediction instead of the size adjustment. The size adjustment did not improve the accuracy for ISBSG based estimates.

In our case we know that the mean *PDR* of Megatec is 5.0 hours per function point compared to ISBSG with 8.4 hours. The adjustment uses these two numbers as follows:

$$Effort_{TARGET} = \frac{Effort_{ANALOGUE}}{PDR_{ISBSG}} \times PDR_{Megatec}$$

This kind of adjustment led to an improvement in accuracy as presented in table 19. It also reinforces the major difference between Megatec and ISBSG data lies in the *project delivery rate* and the required *effort*.

The table below summarizes the results for ISBSG based estimates as done before for Megatec based estimates. The results of both are statistically compared afterwards.

Table 19. Mean Absolute Relative Errors and the PRED of the Estimates based on ISBSG projects

Analogue Selection Method	Adjustment Method	No. of Analogues	Mean ARE in %	PRED(0.25) in %
Regression	-	-	52	37
ACE	None	1	82	16
		2	108	11
	Linear Size Adjustment	1	90	21
		2	114	0
	Productivity Adjustment	1	43	37
		2	48	42

When using the 225 ISBSG projects as base for the 19 estimates, the results that are measured in mean ARE, deteriorate in comparison to the Megatec based estimates. The regression performs better than analogy, which performed best for Megatec. Using one analogue without size adjustment is the best method for ACE, but it is the worst one for Megatec. The size adjustment doesn't improve the accuracy, but the results are much better after applying the productivity adjustment.

Statistically the methods are not significantly different in mean ARE apart from comparing regression and ACE using two analogues. Regression obviously adjusts effort better than ACE, where the effort from the analogue project(s) is just adopted for the prediction. The values of standard deviation also confirm this with higher values for the four ACE techniques than for regression, which means the values for ARE are more widely spread.

Table 20. Results of the t-tests and the standard deviations to compare the different techniques

	ACE-1 no SA	ACE-1 with SA	ACE-2 no SA	ACE-2 with SA	Standard deviation of mean ARE in %
ACE-1 no SA	-	0.200	0.141	-	82
ACE-1 with SA	0.200	-	-	0.222	90
ACE-2 no SA	0.141	-	-	0.643	110
ACE-2 with SA	-	0.222	0.643	-	114
Regression	0.089	0.060	0.002	0.005	52

5.3. Comparison between Megatec and ISBSG based estimates

As mentioned before, the estimates based on ISBSG are not as accurate as the ones based on Megatec. The reasons for this are mainly the difference in the data sets. The Megatec data is much more homogenous than the ISBSG data is. The two data sets are distinct in terms of their business area types, for instance. For the estimation the difference in *PDR* between target project and source projects is important, which badly influences the accuracy of ACE.

Furthermore, the ISBSG data contains a lot of outliers, which influence the regression line or equation a lot. Therefore, the intercept, the coefficients, and thus the predicted effort are also influenced. This influence can improve the estimates when the determined regression model fits the condition of the target project, but this is different for the ISBSG repository.

If the ISBSG data is searched for analogues however, only one or two projects of the 225 ISBSG projects are chosen to predict the *effort* instead of building a regression model based on all 225 projects. It might happen that the analogues are chosen badly, but it can also be a good choice. The choice of analogues is not well determined by using only five variables for the ranking as mentioned before. This could reason that regression outperforms ACE.

T-tests confirm that there is a significant difference in mean ARE between Megatec and ISBSG based estimates when using ACE apart from using one analogue without size adjustment.

Table 21. Independent samples t-tests for the ARE of the different estimation methods

Comparison	Mean ARE in % for Megatec	v	Mean ARE in % for ISBSG	Difference in Mean ARE%	p-Value (2 tailed)
Regression	47	v	52	+5	0.397
One analogue without size adjustment	63	v	82	+18	0.401
One analogue with size adjustment	38	v	90	+52	0.008
Two analogues without size adjustment	54	v	110	+56	0.019
Two analogues with size adjustment	37	v	114	+77	0.004

Therefore, it can be concluded that estimates based on Megatec are more accurate than those based on ISBSG.

It is interesting to note, that the difference in mean ARE rises in each row; at the same time it shows the order of ranking of the different techniques. Regression outperforms ACE and using one analogue outperforms using two. The p-value decreases row by row as well, which is an indication for the degree of significance.

5.4. Comparison of ranking and selecting

5.4.1. The metric *development type*

The estimates are done with the final variable set and the 225 ISBSG projects, which are from different kinds of development types. These are the results got from ISBSG based ACE estimates. The second run of ACE used only 145 projects that were new developments only. In this case the variable *development type* was excluded for the ranking algorithm. Therefore, all projects matched the *development type* of the Megatec projects.

Regression are also done twice, because the regression model are done once for all projects from resource level 1 and 2 and another time for new developments only. The following regression models are utilized:

For the ranking: $\ln(\text{effort}) = 2.228 + 0.655 \cdot \ln(\text{max team size}) + 0.748 \cdot \ln(\text{fp})$

For the selecting: $\ln(\text{effort}) = 2.393 + 0.822 \cdot \ln(\text{max team size}) + 0.684 \cdot \ln(\text{fp})$

The two regression models do not show a big difference when looking at the coefficients and the intercepts. This could be the result of the very similar descriptive statistics for both groups of data. Range, mean and standard deviation of *effort*, *team size*, *function points* and *PDR* had almost the same values as seen before.

In the table only the results obtained from the selection are presented, because the ones for the ranking were presented before (section 5.2.).

Table 22. Predicted effort when using the selected ISBSG projects (new developments only) and applying regression and ACE with the four estimation techniques

ID	Actual Effort	Regression	ARE in %	One analogue without SA	ARE in %	One analogue with SA	ARE in %	Two analogues without SA	ARE in %	Two analogues with SA	ARE in %
1	3777.00	3777.00	57	8290.00	119	7642.45	102	5957.50	58	5985.63	58
2	4388.50	4388.50	93	16502.00	276	15928.77	263	12159.00	177	12449.22	184
3	1647.25	1647.25	20	839.00	49	866.97	47	866.00	47	920.50	44
4	2069.75	2069.75	26	3939.00	90	4928.29	138	3424.50	65	3742.87	81
5	1946.80	1946.80	8	3939.00	102	3561.67	83	3754.50	93	3629.22	86
6	1500.00	1500.00	64	183.00	88	235.16	84	796.00	47	785.72	48
7	1113.50	1113.50	4	1620.00	45	1697.78	52	1109.50	0.4	1075.75	3
8	362.20	362.20	35	17.00	95	7.29	98	100.00	72	38.63	89
9	921.00	921.00	30	804.00	13	1000.00	9	1021.00	11	676.39	27
10	193.50	193.50	117	759.00	292	661.07	242	687.50	255	553.36	186
11	1218.00	1218.00	119	784.00	36	790.58	35	2707.00	122	2178.45	79
12	1317.50	1317.50	33	375.00	72	373.07	72	2157.00	64	1976.51	50
13	529.40	529.40	49	30.00	94	31.62	94	235.00	56	307.10	42
14	691.25	691.25	6	4622.00	569	4693.81	579	2402.50	248	2499.95	262
15	290.75	290.75	37	440.00	51	327.85	13	235.00	19	172.82	41
16	447.50	447.50	33	4622.00	933	4137.79	825	2402.50	437	2203.81	392
17	261.50	261.50	252	4622.00	1668	4784.32	1730	2610.50	898	2650.00	913
18	13904.75	13904.75	33	6068.00	56	10609.00	24	8419.00	39	21680.48	56
19	414.75	414.75	21	183.00	55	210.36	49	100.00	76	116.13	72

The performance of regression is very similar for the selected projects rather than for all projects from resource level 1 and 2. Even the ARE's are comparable, which means that the same projects have a high or low ARE, both when using only the selected projects and all the 225 projects as base for the estimates. Their results were discussed before.

For ACE, the same projects are always overestimated: projects 2, 5, 10, 16 and 17. But their AREs are even higher here. Additionally, project 14 has a high ARE, which is caused by the most similar ranked project. This project needed 27.5 hours per function point, but has almost the same number of *function points* as target project 14. The second analogue, however, matches much better and the ARE decreases a lot.

The reasons for the other overestimates were mentioned already. There is a discrepancy in *PDR*, which makes it difficult to estimate the target projects. This difference seems to become larger when selecting new developments only.

When looking at the results obtained from the selecting, the general tendency of the performance of the analogy-based techniques is opposite to the one obtained from the ranking of ISBSG projects.

Table 23. Mean Absolute Relative Errors and PRED of the estimates based on ISBSG projects compared with selected projects that are only new developments

Analogue Selection Method	Adjustment Method	No. of Analogues	Mean ARE in %	PRED(0.25) in %
<i>Ranking</i>				
Regression	-	-	52	37
ACE 225 projects	None	1	82	16
		2	108	11
	Linear Size Adjustment	1	90	21
		2	114	0
<i>Selecting</i>				
Regression	-	-	55 (41) ⁶	26
ACE 145 projects	None	1	248 (160)	5
		2	147 (99)	16
	Linear Size Adjustment	1	239 (148)	16
		2	143 (95)	5

The selection of matching development types generally deteriorated the mean ARE for all estimation methods. When project 17 is excluded as a target project, the results for selecting (numbers in brackets) become much better. Using two analogues obviously improves the estimation accuracy compared to one analogue. This tendency conforms to the estimation results that were based on Megatec. The size adjustment shows no difference in accuracy.

When looking at the results obtained from the ranking, they conform to the results obtained from ISBSG based estimates. Nevertheless, no significant difference in accuracy can be mentioned between the two methods.

⁶ The mean ARE obtained from excluding project 17 as being a target project.

Table 24. Results of the t-tests and the standard deviations to compare the different techniques for selecting new developments only

	ACE-1 no SA	ACE-1 with SA	ACE-2 no SA	ACE-2 with SA	Standard deviation of mean ARE in %
ACE-1 no SA	-	0.296	0.048	-	412
ACE-1 with SA	0.296	-	-	0.063	417
ACE-2 no SA	0.048	-	-	0.509	211
ACE-2 with SA	-	0.063	0.509	-	209
Regression	0.099	0.143	0.490	0.593	58

The different techniques for combining selecting and ACE are not significantly different, apart from using ACE with one analogue compared with two analogues without size adjustment (p-value is less than 0.05). The values of standard deviations of mean ARE are very high, which indicates that a variety of values differ greatly from the mean. This is the case especially for ACE when using only one analogue as a base for the estimation.

There is no significant difference between selecting and ranking when applying independent samples t-tests to compare them. Therefore, it can not be concluded that *development type* is a cost factor within the ISBSG data set.

Table 25. Independent samples t-tests for the ARE of the different estimation methods

Comparison	Mean ARE in % for Ranking	v	Mean ARE in % for Selecting	Difference in Mean ARE%	p-Value (2 tailed)
Regression	52	v	55	+3	0.898
One analogue without size adjustment	82	v	248	+166	0.099
One analogue with size adjustment	90	v	239	+149	0.143
Two analogues without size adjustment	110	v	147	+37	0.490
Two analogues with size adjustment	114	v	143	+29	0.593

5.4.2. The metric *Client/Server*

The estimates are based on the projects which got answered properly (No or Yes). This means that due to the amount of missing values there are only 62 projects left. When doing the normal estimates before, dummy values are introduced where values were missed. Therefore, a project with a dummy value is more or less excluded from getting good ranks, especially for this certain variable. Now missing values for the variable *Client/Server* don't exist any more and, hence there might be a difference when doing the estimates.

When utilizing ACE *function points*, *maximum team size*, *language type* *development type* and *Client/Server* were in the variable set for the ranking. When performing the selection, the metric *Client/Server* was excluded and replaced by two different subsets,

one for projects that were from the client/server group and one for those projects that are non-client/server-software.

This is also done for the regression. One model is built for the 62 projects, one for the projects with answer YES and another one for the projects with the answer NO to the client/server-question. The following regression models were utilized:

For the ranking: $\ln(\text{effort}) = 1.713 + 0.983 * \ln(\text{max team size}) + 0.721 * \ln(\text{fp})$

For the answer YES: $\ln(\text{effort}) = 0.884 + 1.003 * \ln(\text{max team size}) + 0.824 * \ln(\text{fp})$

For the answer NO: $\ln(\text{effort}) = 1.865 + 0.997 * \ln(\text{max team size}) + 0.701 * \ln(\text{fp})$

A difference in intercepts can be seen between “YES” and “NO”. The low value (0.884) for the intercept indicates that the effort for client-server software is less than the one for not client-server software that. This indicates that within the ISBSG projects that develop client-server software required less basic effort than the projects that do not develop client-server software. This is consistent within the ISBSG data but not with the normal expectations, because client-server-software normally uses more stringent constraints and requirements. So far, no reason has been found for this.

The actual values for each estimate are not documented, because they are similar to the previous predictions and don’t need to be explained again. The table below gives an overview about the performance of the different techniques.

Table 26. Mean Absolute Relative Errors and PRED of the Estimates based on ISBSG projects comparing ranking and selecting for variable *Client/Server*

Analogue Selection Method	Adjustment Method	No. of Analogues	Mean ARE in %	PRED(0.25) in %
<i>Ranking</i>				
Regression	-	-	44	32
ACE 62 projects	None	1	61	26
		2	52	42
	Linear Size Adjustment	1	106	26
		2	88	32
<i>Selecting</i>				
Regression	-	-	50	26
ACE 62 projects	None	1	56	32
		2	52	47
	Linear Size Adjustment	1	103	26
		2	90	26

No significant difference can be observed between ranking and selecting. When using ACE the selected analogues are often the same and in both cases the value of ARE deteriorates when using size adjustment, which is an indication of a nonlinear relationship between size and effort.

Statistical tests can confirm this. The only significant difference can be found between using size adjustment or not when using two analogues both for ranking and selecting. There is no difference between ranking and selecting. This could indicate that the variable *Client/Server* is not a cost factor for the ISBSG projects when estimating the Megatec projects.

5.4.3. The metric *Development Platform*

The estimates were based on the projects with an answer to the question about the *development platform*. This means that due to the amount of missing values and the exclusion of projects developed on mainframes (Megatec projects are developed on PCs or midrange) 79 projects are left. A difference in predictions could be expected because there are only matching projects concerning the development platform.

For the ranking: $\ln(\text{effort}) = 1.624 + 0.911 * \ln(\text{max team size}) + 0.767 * \ln(\text{fp})$

For the answer MR: $\ln(\text{effort}) = 2.561 + 0.561 * \ln(\text{max team size}) + 0.710 * \ln(\text{fp})$

For the answer PC: $\ln(\text{effort}) = 1.444 + 1.228 * \ln(\text{max team size}) + 0.713 * \ln(\text{fp})$

Again there is a big difference in intercepts to realize between software developed on a PC or midrange machine. Basically, when looking at the ISBSG data it was found, that software developed on a PC needed less effort than software developed on a midrange machine. The difference in mean effort between both groups is 1700 hours. This seems to be the reason for the difference in intercepts between the two regression models.

Table 27. Mean Absolute Relative Errors of the Estimates for ISBSG projects comparing ranking and selecting of the variable *development platform*

Analogue Selection Method	Adjustment Method	No. of Analogues	Mean ARE in %	PRED(0.25) in %
<i>Ranking</i>				
Regression	-	-	48	37
ACE 79 projects	None	1	93	21
		2	122	32
	Linear Size Adjustment	1	128	5
		2	141	32
<i>Selecting</i>				
Regression	-	-	54	37
ACE 79 projects	None	1	102	21
		2	115	32
	Linear Size Adjustment	1	132	21
		2	125	21

Regression outperforms ACE again for both ranking and selecting. There are no significant differences between selecting and ranking found out. Size adjustment deteriorates the values of mean ARE.

When looking at these results, it can not be concluded that *development platform* is a cost factor for the ISBSG projects, either.

5.5. Further results

Additionally, we explored whether the prediction accuracy improves when using the same scale of *function points* for ISBSG projects as we have got for the Megatec data. The *function points* ranges from 39 to 3290. A subset of ISBSG projects was created that shows the same range of *function points*. This subset was used as base for the estimation as well, but this step did not lead to a significant change in accuracy, either; the accuracy improved by 1% in mean ARE for the regression at least.

This also reinforces the fact that the main difference between the projects of Megatec and ISBSG is their required *effort* and, hence, also in the *PDR*. The range of *function points* also differs, but on average the two data sets show comparable numbers. When using ACE, it is searched for projects with similar features. It is likely to find similar projects in a large data set. The problem with the ISBSG data is that the measured *effort* of the completed software projects is very high and is used to predict the *effort* of the target project. Thus, the effort predictions are often too high and not accurate enough.

6. Summary and conclusions

This section summarizes the major findings of this study and discusses possible areas for further research.

The current study compared one parametric with a non-parametric estimation method using a large, multi-organizational and a smaller, company-specific data set. These data sets were base for estimating certain target projects. The choice of metrics of the data sets used for the estimations was based on their availability in both data sets and their comparability; this means the data for a certain metric had to be collected in the same way, although it is obtained from different organizations. Therefore, the number of metrics included in the actual research process is relatively small, but it ensures the comparability of the results. The estimation results were compared based on the prediction accuracy.

The main research question of the study aimed to investigate whether there is any difference in accuracy between using multi-company data, which is the ISBSG data, and data from one company (Megatec) as a base in the estimation process. The accuracy was measured in mean ARE between estimated and actual effort, and was analyzed with the help of t-tests. The estimates were obtained from OLS regression and the tool ACE. The results show:

Estimates based on Megatec are significantly more accurate than those based on ISBSG or in other words using multi-company data for cost estimates is not as accurate as when using one-company data. This does not confirm the findings of [Briand et al. 98]. Reasons for this can be the methodologies of data collection, the difference in population and distribution between the two data sets, and the heterogeneity of the ISBSG data:

Cost estimation experts collected the Megatec data, whereas the ISBSG data was handed in as a survey filled out by a project member. Furthermore, the ISBSG data is a collection

from many different organizations, thus, the accuracy of answering the survey might differ from one to another organization.

The population or distribution of the two data sets was also compared. It was found that there are, for example, differences in how the *effort* was recorded (different levels), differences in *development type*, *language type* and in the *project delivery rate*. It was possible to create subsets of the ISBSG data that better fit the conditions of the Megatec data concerning these categorical metrics. However, the difference in the mean *project delivery rate*, which is mainly related to *effort*, couldn't be adapted by creating subsets. Thus, there was a big difference in required *effort* between the target and the source projects. When using analogy to predict effort, the estimated effort often was much higher than the actual measured one. Therefore, when performing cost estimation, it should be assured that target and source projects show similar features, at least for the *PDR*. Especially when using analogy the *project delivery rate* has to be comparable. An adjustment of the predicted effort by the difference in mean *PDR* between Megatec and ISBSG data, which is about 60%, improved the estimation accuracy of the ISBSG based estimates significantly in comparison to using no adjustment or linear size adjustment.

It was also found that the ISBSG data is not as homogenous as the Megatec data. The ISBSG data is much more skewed and doesn't show a general behavior, which means the statistics that were performed show different trends. Further studies of some metrics of the ISBSG repository also indicate that the projects stored in the ISBSG repository are have got a wide range, thus, it is not a homogenous data set. This heterogeneity is probably reasoned by the variety of business area types and organizations that are participated in the ISBSG repository. Therefore, a certain business area type should be investigated separately to address their different requirements. Furthermore, an organization key should be introduced to facilitate more reliable and accurate cost estimation by ensuring that the data is collected in a more consistent way.

This study also found that regression generally outperforms ACE when using the ISBSG repository. Although previous studies [Shepperd and Schofield 97, Walkerden and Jeffery 98] showed promising results for analogy, the results of the current study can not conclude this for the ISBSG data. The prediction accuracy is much less when using ACE than for regression, which is probably also reasoned by the strong heterogeneity of the ISBSG data. A combination of several estimation techniques could be a more accurate opportunity for the ISBSG data.

Walkerden and Jeffery [98] recommended applying linear size adjustment for ACE. It does improve the results for Megatec based estimates, but deteriorates the ISBSG based estimates. Therefore, linear size adjustment is not recommended for the ISBSG repository.

Another conclusion we can draw from these results is, that a cost model that performs well in one environment doesn't have to be as good in a different environment. When using analogy, Megatec based estimates were much more accurate than ISBSG based estimates. When using ACE, the ranking algorithm didn't seem to be suitable enough for the ISBSG data, because results obtained from ACE show a wide range in ARE. In order to obtain better results the cost model has to be adapted to the ISBSG data. The ranking-algorithm could be improved by using more ratio scaled variables and also by using weights for the different variables according to their influence on *effort*.

Analogy performs much better for Megatec than for ISBSG based estimates. Analogy might be more sensitive to the homogeneity of the data, thus, it is outperformed by regression when using ISBSG data. This confirms the results of Briand et al. [98] concerning analogy, which say that analogy doesn't seem to be as robust when using data external to the organization for which the model is built.

The estimation results based on the Megatec data confirm the findings of Walkerden and Jeffery [98]; using ACE with size adjustment performs better than regression. Furthermore, it was found that the use of two analogues improves the prediction accuracy for the Megatec based estimates.

The investigation of the ISBSG data for potential cost factors yielded that the main cost factors are *maximum team size* and *function points*. The examination of some of the categorical metrics stored in the ISBSG repository did not lead to any conclusions about their potential of being a cost factor. We created subsets of the ISBSG data in order to see whether this would improve the estimation accuracy. These subsets only included software projects that fit the development type or the scale of function points of the Megatec projects, but the accuracy of the prediction didn't improve significantly.

Furthermore, it would be wrong to conclude that a bigger data set yields more accurate estimation results. If poor data is used, the cost estimates won't be much accurate. Therefore, it is more important to collect the relevant data carefully and with the knowledge in mind of how to use the data in cost models and in which business area the data is used. Therefore, it is also recommended to include more relevant metrics in the data collection as experience measures, an organization key, and application platform, for example. They could improve especially the analogy-based estimates.

References

- [Albrecht and Gaffney 83] A.J. Albrecht, J.E. Gaffney, "Software function, source lines of code and development effort prediction: a software science validation", IEEE Transactions on Software Engineering, 9 (6), 639-648, 1983
- [ANGEL] http://dec.bournemouth.ac.uk/dec_ind/decind22/web/AngelPage.html
- [Ariane] www.rvs.uni-bielefeld.de/publications/Incidents/
- [Boehm 81] B.W. Boehm, "Software Engineering Economics", Prentice Hall, Englewood Cliffs, 1981
- [Breiman et al. 84] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, "Classification and Regression Trees", Wadsworth & Books/Cole Advanced Books & Software, 1984
- [Briand et al. 98] L.C. Briand, K. El Emam, D. Surmann, I. Wiczorek, K. Maxwell, "An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques", ISERN-Report-98-27.
- [Brooks 75] F. P. Brooks, "The Mythical Man Month", Addison-Wesley. Reading, 1975
- [CBR] <http://www.cbr-web.org/>
- [Conte et al. 86] S.D. Conte, H.E. Dunsmore, V.Y. Shen, "Software Engineering Metrics and Models", Benjamin/Cummings Publishing Company Inc., 1986
- [Fenton and Pfleeger 96] N. Fenton, S. Pfleeger, "Software Measurement: A rigorous and practical approach", International Thomson Computer Press, 1996.
- [Heemstra 92] F.J. Heemstra, "Software cost estimation", Information Software Technology, 34 (10), 627-639, 1992
- [IFPUG] <http://www.ifpug.org>
- [ISBSG] Software Project Estimation, "A Workbook for Macro-Estimation of Software Development Effort and duration. ISBSG", 1999 or <<http://www.isbsg.org.au/>>
- [Jeffery and Stathis 96] R. Jeffery, J. Stathis, "Function Point Sizing: Structure, Validity and Applicability", Empirical Software Engineering, 1, 11-30, 1996

- [Ladkin 98] P. Ladkin, "Computer-related incidents with commercial aircrafts", University of Bielefeld, www.rvs.uni-bielefeld.de/publications/Incidents/
- [Lederer and Prasad 93] A. L. Lederer, J. Prasad, "Information systems software cost estimating: A current assessment", *Journal of Information Technology*, 8, 22-33, 1993
- [Lokan 99] C. J. Lokan, "An empirical study of inter-item relationships in function points", to be published, 1999
- [Mukhopadhyay et al.92] T. Mukhopadhyay, S. Vincinanza, M. J. Prietula, "Estimating the feasibility of a case-based reasoning model for software effort estimation", *MIS Quarterly* 16(2), 1992
- [Norusis 98] M. J. Norusis, "SPSS 8.0 Guide to Data Analysis", Prentice Hall, 1998
- [Shepperd and Schofield 95] M. J. Shepperd, C. Schofield, "Software Support for Cost Estimation by Analogy", *Proceedings of the ESCOM 6*, Rolduc, 1995.
- [Shepperd et al. 96] M. J. Shepperd, C. Schofield, and B. Kitchenham, "Effort estimation using analogy", *Proceedings of the 18th International Conference on Software Engineering*, Berlin, 1996.
- [Shepperd and Schofield 97] M. J. Shepperd, C. Schofield, "Estimating Software Project Effort Using Analogies", *IEEE Transactions on Software Engineering*, Vol.23, No.12, November 1997
- [Walkerden and Jeffery 98] F. Walkerden, R. Jefferey, "An empirical study of analogy-based software estimation", *CAESAR-Report-98-8*.