



Free/Libre and Open Source Software Metrics

Sponsored through Framework Programme Sixth (Call 5) by



The FLOSSMetrics Consortium consists of: Universidad Rey Juan Carlos, University of Maastrich, Wirtschaftsuniversitaet Wien, Aristotle University of Thessaloniki, Conecta s.r.l., Zea Partners and Philips Medical Systems PMS Nederland B.V.

Document Information

Version: 3.0
Date : Nov. 25, 2009
revision : 0

Owning Partner: UM

Author(s):
Kirsten Haaland
Ioannis Stamelos
Rishab A. Ghosh
Ruediger Glott
Eleni Kwnstantinou
Eleni-Maria Stea

Reviewer(s):
Sulayman Sowe
Stefan Koch

To:
Public

Purpose of distribution:

Printed on at

Status:

- Draft
- To be reviewed
- Proposal
- Final/Released

Confidentiality:

- Public - Intended for public use
- Restricted - Intended for FLOSSMETRICS consortium only
- Confidential - Intended for individual partner only

Deliverable ID:


D11.3

Title:

Cost/effort estimation study (for libre software projects)

License for distribution:

This work is licensed under a [Creative Commons Attribution-Share Alike 3.0 License](http://creativecommons.org/licenses/by-sa/3.0/)
(The license can be found in <http://creativecommons.org/licenses/by-sa/3.0/>)


	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 2 of 23
		Version: 3.0
		Date: Nov. 25. 2009
		Status : Public Confid : Final

Deliverable: D11.1

Title: Cost/effort estimation study (for libre software projects)

Executive Summary:

In this report we describe the high level study that aimed at the approximation of the substitution costs for FLOSS. Substitution cost is the monetary value of the effort necessary for implementing a FLOSS application from scratch in a software company. We describe our approach for estimating substitution cost, based on building generic estimation models for the modern software industry. For this purpose we use the latest version of ISBSG, and apply our models on a subset of projects from a Debian distribution (pilot study), and on a subset of the FLOSSMetrics Database projects (final study). This yields an approximation of the total substitution costs for these two collections of FLOSS applications. We report on various problems, limitations and issues related to the data, the model precision, and the currently available FLOSS project data. However, the availability of more elaborated information in the FLOSSMetrics Database is expected to alleviate some of these issues, and more precise calculations will become feasible, using the same methodology as outlined in this report.

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 3 of 23
		Version: 3.0
		Date: Nov. 25. 2009
		Status : Public Confid : Final

CHANGE LOG

Ver.	Date	Author	Description
0.1	09/09/2009	Kirsten Haaland	Initial proposal for structure
0.2	29/20/2009	Kirsten Haaland	Edit structure and content
1	08/10/2009	Kirsten Haaland	First version
2	10/11/2009	Ioannis Stamelos	Final version (for review)
2.1	14/10/2009	Kirsten Haaland	Review
2.2	21/10/2009	Ioannis Stamelos	Minor updates EbA
2.5	22/11/2009	Kirsten Haaland	Minor updates
3.0	25/11/2009	Stefan Koch	Review and release

APPLICABLE DOCUMENT LIST

Ref.	Title, author, source, date, status	Deliverable Identification



	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 4 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final

TABLE OF CONTENTS

1. Introduction	5
2. Methodology for substitution cost estimation	7
3. Dataset preparation	9
4. Building the Estimation Models	15
4.1 OLS Model	15
4.2 Analogy Based Model	16
5. Results	18
6. Validity threats to the study	20
6.1 Use of SLOC as a size metric	20
6.2 Use of ISBSG for model building	20
6.3 Estimation model precision	20
6.4 Debian and FM3-2 sample project composition	21
7. Conclusions and Future work	22
8. References	23

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 5 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final


1. INTRODUCTION

A wealth of successful software applications has been brought forward by Free/Libre Open Source Software (FLOSS); the new software development paradigm based on volunteer participation and open development processes. FLOSS plays an increasingly important role in the ICT sector in particular, and the world economy in general. It is therefore interesting to attempt to estimate the economic impact of FLOSS.


In this paper we report our efforts on providing an informed estimate ([1]) of the substitution costs for a large portion of a Free/Libre Open Source Software code base. Substitution costs refer to how much it would have cost to build the same software from scratch entirely within a single firm in a proprietary software development model. Calculating the substitution costs for a given set of FLOSS applications is one way of assigning a Euro value to the production and effort represented by these applications.

We use the latest version of ISBSG dataset [2] to build the estimation models. We need to remember that our study is conducted in the context of the FLOSSMetrics project [4]. The objective of FLOSSMetrics is “to construct, publish and analyze a large scale database with information and metrics about FLOSS software development coming from several thousands of software projects, using existing methodologies, and tools already developed”. Since the database was still under construction when the study started (February 2008, Maastricht), we initially used an already available dataset of Debian 3.1 to draw preliminary conclusions about the FLOSS substitution costs. We call this first attempt as “pilot study”. We then applied the same approach on a subset of the FLOSSMetrics Database in November 2009, as close as possible to the end of the project. We call this second run the “final study”.

We wish to independently generate as many models as possible, in order to deal with the inherent prediction uncertainty involved in building a cross-organizational model ([3]). In this document we report results from two well-known estimation methods, namely Ordinary Least Squares Regression (OLS) and Estimation by Analogy (EbA). We produce two models with each method and compare their results.

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 6 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final

The rest of the report is organized as follows: Section 2 develops and describes methodology for substitution cost estimation, i.e. the prerequisites that an estimation model should fulfil in order to be used for the calculation of FLOSS substitution cost. Section 3 describes the preparation of ISBSG, DEBIAN and FLOSSMetrics subsets, while Section 4 provides a brief description of model generation and reports results from the two studies. Finally Section 5 contains preliminary conclusions and future work.

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 7 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final


2. METHODOLOGY FOR SUBSTITUTION COST ESTIMATION

Our aim is to calculate substitution costs for a set of FLOSS projects. We will do that by estimating the substitution cost for each project separately and will proceed by producing an estimate for the entire project set. Estimation for each project separately is possible through the application of a parametric estimation model, i.e. a model that will take into account project parameters such as size, application domain, elapsed time etc. Such an approach has been already taken [5] applying COCOMO, 1981 model [1] on Debian 3.1.

Our aim is to develop a cross-organizational estimation model that is representative of modern software industry. Such model would be more suitable for our purposes than COCOMO II [6] or similar models. The motivation of our approach is that it is reasonable to presume that a large code base of FLOSS would not be substituted by a single company or even a limited number of companies, residing in one or two countries. On the other hand, it is known that different companies may have disparate productivity levels. It is evident that a universal cost estimation model is needed. We emphasize that such model is about closed source development of projects that would be equivalent to specific FLOSS project counterparts. It is not our purpose to build any estimation models for production of software by FLOSS communities.

We hereby provide a comprehensive list of the characteristics, which such model should have according to our current understanding of the problem:


1. The model should be representative of the modern, global software industry. The model should capture productivity levels that would support our requirement for estimating costs of a typical, average productivity software company. We will use the ISBSG dataset for generating estimation models.
2. The model should be based on some measure of physical size. Functional sizing (e.g. Function point analysis) is not practised in FLOSS, and it would be infeasible to measure the functional size of thousands of FLOSS projects, given that no automation tools are

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 8 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final

available for such task at the moment. We chose to use Source Lines of Code as a measure of physical size because they are readily available for FLOSS projects. SLOC have been used also in the study that can be found in [5]. However limitations of SLOCs are already known, so we need to take any possible precaution for their use in the model. In particular, we should keep in mind that SLOC measurements may contain a lot of noise.

3. The model should be based on closed source projects similar to those found in FLOSS. In particular, it should be based on projects that are similar to the FLOSS projects for which it will be used to produce substitution costs. Project attributes that define project identity, and potentially affect productivity and therefore cost, are the project application type, development platform, language type, size etc. The methodology should build generic estimation models that take into account these attributes, then, while calculating substitution costs, carefully chosen values to characterize FLOSS projects should be used.
4. The estimation model should produce interval estimates to account for data uncertainty. Given that the model will be calibrated on multi-organizational project data and will be applied on hundreds or thousands of projects developed by communities of volunteer programmers, a range of possible cost values should be produced. Ideally, the estimate for each FLOSS project and for the entire set of FLOSS projects should be a range of values along with a probability distribution.
5. Because of the inherent uncertainty in this estimation context, we should build more than one model. This is a typical precaution in software cost estimation. The precision of the models should be first assessed and then they should applied to the FLOSS projects under study to produce a variety of estimates. Such estimates should then be compared in order to (a) produce the final estimate and (b) assess the overall precision of the calculated substitution cost.

Having defined the prerequisites of the estimation models we now precede with defining the steps for generating the models.

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 9 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final


3. DATASET PREPARATION

We base our study on three datasets, namely the ISBSG dataset, the Debian 3.1 dataset and the FM3_Nov_2009 subset. The ISBSG dataset Revision 10 (called ISBSG R10 in the rest of the paper) consists of only closed source projects, and represents the modern software industry. ISBSG is a non-profit organization that collects data on software projects from all over the world. We believe that ISBSG is the most reliable source for calibrating and building a cross-organizational software cost estimation model [3]. Our training set is a subset of ISBSG R10, which contains 443 projects measured with SLOC. Additional ISBSG categorical project attributes that are meaningful for FLOSS projects as well are language type, resource level, and development type and platform (see www.isbsg.org).

Our first concern was to validate SLOC as size metric for our data. ISBSG organization provides guidelines for using the data in ISBSG R10. Because they have not validated SLOC values they advise not to use SLOC counts. We overcame this issue by (a) inspecting SLOC counts and performing a sanity check, leading to the adjustment of some evidently wrong numbers, (b) analyzing the correlation with FP, which is the major size metric in ISBSG, and (c) referring to other papers that have already used SLOC related information from ISBSG and reached meaningful results (e.g. [7], [8]).

After some data cleaning (projects with low, i.e. C or D, ISBSG data quality rating have been excluded) and outlier analysis based on productivity measured in SLOC/manhour, we produced a dataset of 395 projects (ISBSG-SLOC-all subset). Three observations were identified as duplicates, two as being outliers further than two standard deviations away from the SLOC/manhour measure, thus the majority of observations, 44 in total, were excluded due to low quality rating.

In order to investigate the validity and quality of SLOC as size metric in ISBSG R10, we built regression models with SLOC as independent and Summary Work Effort as dependent variable in

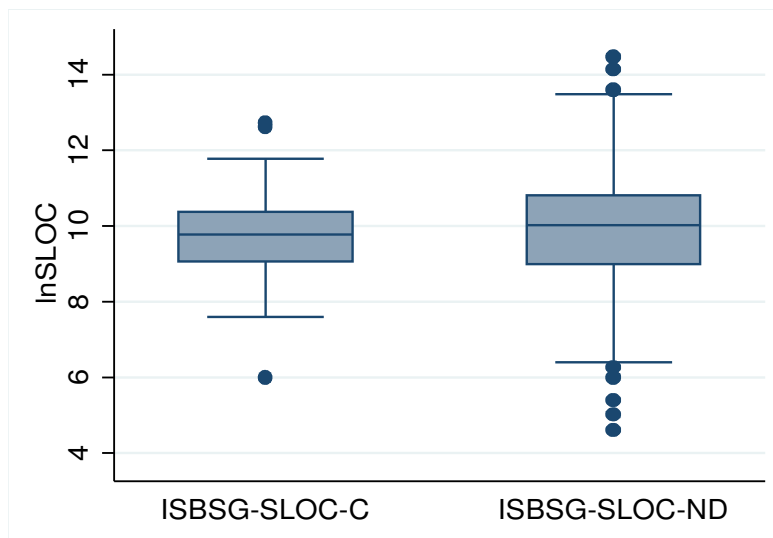
	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 10 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final

ISBSG, after transforming them on a logarithmic scale (variables LNSLOC and LNEFFORT respectively, while LNFP stands for the logarithmic transformation of variable Functional Size in ISBSG). A linear regression model with LNSLOC as independent and LNEFFORT as dependent variable produced an adj-R2 equal to ~0.57, showing that there is a good correlation between the two variables. Another linear regression model, with LNSLOC as independent and LNFP as dependent variable produced an adj-R2 equal to ~0.6, showing that there is a good correlation between the project functional and physical size as well. Interestingly, a regression model with LNFP as independent and LNEFFORT as dependent variable produced an adj-R2 equal to ~0.50, showing significant, but weaker than SLOC, correlation between Function Points and project effort.

Based on the results of the correlation analysis and the use of SLOC in ISBSG by other researchers, we decided to proceed with our study, using SLOC as a proxy for the physical size of both closed and open source projects. The ISBSG-SLOC-all 395 project dataset consists of quite heterogeneous projects, written in various languages. It is difficult to generate an accurate effort estimation model under these conditions, so we devised two further subsets. One subset, namely ISBSG-SLOC-ND, consists of only 176 “New Development” projects written in various languages. Another subset, namely ISBSG-SLOC-C, consists of 48 ISBSG-SLOC-all projects all written primarily in the C language.

Figure 1 presents a boxplot of the distribution of the two datasets. From the figure it is evident that there are some outliers, especially in the ISBSG-SLOC-ND dataset, however they play a significant and important role building our estimation models, and have therefore been included.


Figure 1. Boxplot showing distribution of calibration datasets.



Debian 3.1 consists of 6173 projects in total. We chose an old version of Debian because the majority (57%) of the projects in Debian 3.1 are written in C language, which helps reducing the uncertainty for our estimates. We further developed a subset of 5157 projects that were of comparable size to those found in the two ISBSG training datasets.

Working with the FLOSSMetrics Database, we managed to produce SLOC measurements for 457 projects in total. We also tried to collect qualitative information, by retrieving tags for FLOSSMetrics projects that originated from Sourceforge. This dataset is called FM3-1. We calculated the lines of code for several open source projects as follows:


1. Using a script we connected to the FLOSSMetrics database server and got all the database names. Since, each database keeps information for a different project we had to check them one by one in order to collect the desired information.

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 12 of 23
		Version: 3.0
		Date: Nov. 25. 2009
		Status : Public Confid : Final

2. The main part of the line counter program was checking each database in a loop, collecting the following information: 1- name of the project, 2 - CMS URI. The problem was that all the databases have not got exactly the same structure and naming conventions. Hence, it was difficult to locate the appropriate fields although methods such as regular expressions were used.
3. In each iteration, the program used the CMS URI to download and measure the code and collect some tags that indicate the project category. For the code measurement the SlocCount tool was used. For the tags, the process was more complicated. We had to use a part of each project name as keyword, and search on SourceForge for a page with this keyword in the URI. Then we parsed the page using regular expressions looking for a specific pattern similar to all pages that contained tags. If the pattern was found, we collected the tags, otherwise we continued the iteration.
4. All the information collected was written in a webpage for subsequent use. The webpage is <http://swserv2.csd.auth.gr/loc/svncnt/svnloc>

During the implementation we faced several difficulties:

1. Some of the downloaded projects were very large and the downloading/measurement process took a long time (even days).
2. If a project measurement failed, a project was not found or the server crashed the program had to continue from the last project measured.
3. During the execution of the program, a lot of information in the server, the databases and the project CMSs changed. We had to take this into account in order to avoid execution problems.
4. All the databases did not have exactly the same structure and naming conventions.

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 13 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final


- Tags were not available for each project since some of the projects were not in SourceForge repositories.

The advantage of this technique despite the long execution time is that the measurements are precise and only consistent information is collected. Additionally, it is fault tolerant, since it reconsiders in case of problems.

However, in this dataset the size of several projects were outside the range of the training set, and another subset was created, FM3-2; which excludes the smallest projects, and more importantly, excludes the largest 21 projects which were clearly outside the range of ISBSG training set size range. In addition, Sourceforge tags were not used at all in the study because (1) many projects had no tags because they did not reside on SourceForge, and (2) because it was hard to match SourceForge tag information with ISBSG fields. The descriptive statistics are in Table 1.

Table 1. Descriptive statistics for the datasets

	ISBSG-SLOC-ND	ISBSG-SLOC-C	DEBIAN-1	FM3-1	FM3-2
N	176	48	5157	457	433
Total SLOC	12467849	1852572	101886869	2.06e+08	1.08e+08
Max	1900000	334800	333936	1.36e+07	1913199
Min	100	400	399	1	100
Mean	70840.05	38595.25	19757	451584.5	250436.1
Median	22638	17716.5	5012	106283	97460
Stdev	198819.6	64345.11	39671	1211858	365343.6

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 14 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final


Overall the ISBSG and Debian/ FLOSSMetrics datasets' similarities and differences are:

Similarities:

- both are of “cross-company” nature, OSS projects are developed by a multitude of developers
- both are representative of world-wide software development (projects implemented by developers / companies from all over the world)

Differences:

- closed source vs. open source, different developer motivation, team sizes, release rate, development process, project deliverables (documentation, ...)
- differences in descriptive statistics

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 15 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final

4. BUILDING THE ESTIMATION MODELS

This section provides details on estimation models we have built. We employed the most widespread methods for effort estimation, namely regression models (OLS) and analogy based estimation (EbA), have been applied. The initial idea was to explore more methods, like categorical regression models, analogy combined with regression, and machine learning methods. However, the absence of extensive qualitative information for FLOSS projects did not allow such endeavour.

4.1 OLS MODEL


OLS models have been produced with the use of the statistical software package STATA®. ISBSG categorical variables have been handled through dummy variables. The adj-R2 was 0,78 for ISBSG-SLOC-ND and 0,66 for ISBSG-SLOC-C. Table 2 reports precision figures (MMRE and PRED25) for the two models.

Table 2. Summary results for OLS estimation model

ISBSG-SLOC-ND		ISBSG-SLOC-C	
MMRE	PRED25	MMRE	PRED25
58,26%	28,41%	55,47%	27,08%

MMRE stands for Mean Magnitude of Relative Error, and is a common measure of precision in software engineering. MRE for a single prediction is given by the formula in fig.1 (y_A is the actual project effort and y_E is the estimated effort).

$$MRE = \frac{|y_A - y_E|}{y_A} \quad (1)$$

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 16 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final

PRED25 refers to the amount of projects that are estimated with less than 25% error. Typically in software cost estimation, accuracy values of MMRE less than 25% and PRED25 more than 75% are desirable. However, it is also admitted that estimation results depend heavily on data quality and characteristics.


4.2 ANALOGY BASED MODEL

The EbA model was produced with the use of BRACE ([9]), a tool that supports the calibration of the method on a given dataset through the use of a genetic algorithm and provides interval estimations for project portfolios using bootstrapping ([10]). Calibration of EbA refers to the empirical selection of the best combination of distance metric (e.g. Euclidean, Manhattan distance), number of analogies (one or more), statistic for calculation of estimate (mean or median) and size adjustment (yes or no). We calibrated EbA on both ISBSG_SLOC_all_ND and ISBSG_SLOC_C. Precision results for the EbA models are reported on Table 3.


Table 3. Summary results for EbA estimation models

ISBSG_SLOC_all_ND		ISBSG_SLOC_C	
MMRE	PRED25	MMRE	PRED25
65,57%	18,83%	49,90%	31,25%

As one can see, both the OLS and the EbA model produce results that are not satisfactory from the accuracy point of view. On the other hand, in both cases residuals were randomly distributed and no bias was observed. Therefore, because such models will be applied on a large number of FLOSS projects, we anticipate that errors will counterbalance each other. As a consequence,

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	<p>Page : 17 of 23</p> <hr/> <p>Version: 3.0 Date: Nov. 25. 2009</p> <hr/> <p>Status : Public Confid : Final</p>
---	--	--

although the prediction of a single project effort will not be reliable, we assume that prediction of the overall effort for a large code base will be close to the actual effort needed to implement it.

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 18 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final

5. RESULTS

Based on these models and the assumptions discussed above, we produced rough estimates of the substitution cost for DEBIAN-1 and FM3-2. We assumed that all projects would be substituted / developed using a 3GL type language and that they would be developed from scratch (corresponding to 'new development' in ISBSG data). OLS produced only point estimates for the portfolio of DEBIAN-1 projects (no interval estimates have been produced since the intervals proposed by OLS were too large to produce meaningful values). These are 437,51 Million Euros for ISBSG-SLOC-ND and 412,68 Million Euros for ISBSG-SLOC-C. The results for EbA are shown in Table 4.


Table 4. Substitution costs for DEBIAN-1 5157 software applications (numbers in million Euro)

	Lower bound (95% confidence)	Most probable value	Upper bound (95% confidence)
ISBSG-SLOC-ND	264,67	360,55	462,66
ISBSG-SLOC-C	251,30	328,00	426,61

Results for OLS models on FM3 datasets are provided in Table 5.

Table 5. Substitution costs based on OLS estimations for FM3-1 and FM3-2 datasets (numbers in million Euro)


	ISBSG-SLOC-ND	ISBSG-SLOC-C
FM3-1	361.46	294.54
FM3-2	245.60	207.27

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 19 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final

**Table 6. Substitution costs based on EbA estimations for FM3-1 and FM3-2 datasets
(numbers in million Euro)**

	ISBSG- SLOC- ND_min	ISBSG- SLOC-ND	ISBSG- SLOC- ND_max	ISBSG- SLOC- C_min	ISBSG- SLOC-C	ISBSG- SLOC- C_max
FM3-1	159,8	231,4	322,6	477,6	503,7	781
FM3-2	153	213,4	313,2	92,4	104,6	144,8

We used a conservative estimate of the salary of a Software Engineer/Developer/Programmer with only one year experience, further taking the average of the salary between Europe and the United States. This amounted to 38.814 Euro per year. We also assumed the average working time per year to be 1600 hours.

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 20 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final

6. VALIDITY THREATS TO THE STUDY

There are a number of validity threats to our study. In this section, we discuss the most significant of them.

6.1 USE OF SLOC AS A SIZE METRIC


SLOC is a physical size metric that has been criticized by many researchers. It is considered that it can not be used to represent accurately effort spent for implementing software because it can not be predicted safely at the beginning of the project, it depends heavily on the programming language used, the programming style of each individual programmer, etc. In our study SLOCs are not predicted, they are just counted at a specific project time slot. Other concerns for SLOCs are still valid. We presume that significant part of the prediction error is due to noise produced by the use of SLOC.

6.2 USE OF ISBSG FOR MODEL BUILDING

ISBSG projects used for calibrating the estimation models may be different than the kind of projects found in Debian and FLOSSMetrics. This fact may also be the cause for part of the prediction error.


6.3 ESTIMATION MODEL PRECISION

Due to various assumptions the precision of the two models is not high. As mentioned above, we consider that only the cumulative effort (total substitution cost) can be used for any useful calculations.

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 21 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final


6.4 DEBIAN AND FM3-2 SAMPLE PROJECT COMPOSITION

The projects used in our study are only a portion of the Debian release and the FLOSSMetrics Database. Very large projects and very small projects are not taken into account. Such arrangement may have produced a subset of projects biased towards specific software application types and may have affected the kind of projects that participate in our calculation of substitution cost, reducing the generalizability of our results.

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 22 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final

7. CONCLUSIONS AND FUTURE WORK

This study proved to be an interesting cost estimation exercise. Thousands of projects were considered either for model building and calibration or estimation. We had to produce scripts for managing data and improve the usability of the BRACE tool, which was never operated on so many projects. We believe that these rough estimates demonstrate the feasibility of our approach for the calculation of the substitution cost of FLOSS. For the Debian dataset, point estimates are relatively close to each other and provide a safe order of magnitude for our target cost estimate. In addition, both OLS point estimates fall within EbA intervals. In general, the two methods disagree on their FM3 dataset results, with EbA providing quite disparate estimates for FM3-1 and FM3-2, when ISBSG_SLOC_C is used as estimation base. This may be caused due to the small number of projects (48) in that dataset. However, both methods provide coherent estimates for FM3-2, when based on ISBSG_SLOC_ND. Because ISBSG R10 data originate from many different companies, scattered around the world, our models are of relatively low precision. For the moment, we base our estimates mainly on physical lines of code, and a couple of ISBSG project attributes, namely development type (fixed to “new development”) and language (code written with a variety of programming languages, but with C as primary language). However as more data describing FLOSS projects become available through the FLOSSMetrics project, we intend to build estimation models that will take a variety of project attributes into account. In addition, we will attempt to produce more precise models based on data analysis and the application of other estimation methods; in particular we believe the approach can be enhanced by using machine learning approaches for cost estimation such as demonstrated in [11]. We plan to apply our models on successive versions of the FLOSSMetrics database, providing an informed estimate of the substitution cost of FLOSS by the modern software industry.

	<p>Cost/effort estimation study (for libre software projects)</p> <p>Deliverable ID: D11.3</p>	Page : 23 of 23
		Version: 3.0 Date: Nov. 25. 2009
		Status : Public Confid : Final

8. REFERENCES

- [1] Boehm, B. 1981 Software Engineering Economics, Prentice Hall.
- [2] ISBSG official web site, www.isbsg.org
- [3] Mendes, E., Lokan, C. 2008. Replicating studies on cross- vs single-company effort models using the ISBSG database, EMSE, 13, 1 (Feb 2008) 3-37.
- [4] FLOSSMETRICS project official web site, <http://flossmetrics.org>
- [5] FLOSSIMPACT report, <http://www.flossimpact.eu/>
- [6] B. Boehm, B. Clark, E. Horowitz, R. Madachy, R. Shelby, C. Westland, "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0," Annals of Software Engineering, (1995).
- [7] Aggarwal, K., Singh, Y., Chandra, P., Puri, M, 2005. Bayesian Regularization in a Neural Network Model to Estimate Lines of Code Using Function Points', Journal of Computer Sciences 1, 4 (2005) 505-509.
- [8] Moses, J., Farrow, M., Parrington, N., Smith, P, 2006. A Productivity Benchmarking Case Study using Bayesian credible intervals, Software Quality Journal, 14 (2006) 37-52
- [9] Stamelos, I., Angelis, L., Sakellaris, E. 2001. BRACE: BootstRap based Analogy Cost Estimation, in Proceedings of 12th ESCOM (2001) 17-23
- [10] Stamelos, I., Angelis, L. Managing Uncertainty in Project Portfolio Cost Estimation, Information & Software Technology, Elsevier, 43, 13 (2001)
- [11] Bibi, S., Stamelos, I., Angelis, L., "Combining Probabilistic Models for Explanatory Productivity Estimation", Information and Software Technology, Elsevier, 50(7-8), pp. 656-669 (2008)