Estimating Software Development Effort Based on Phases

Valentina Lenarduzzi, Sandro Morasca
Department of Theoretical and Applied Sciences
University of Insubria
Varese, Italy
valentina.lenarduzzi@gmail.com
sandro.morasca@uninsubria.it

Davide Taibi
Software Engineering Research Group
University of Kaiserslautern
Kaiserslautern, Germany
taibi@cs.uni-kl.de

Abstract—Software development effort estimation is a very important issue in software engineering and several models have been defined to this end. In this paper, we carry out an empirical study on the estimation of software development effort broken down by phase, so that estimation can be used along the software development lifecycle. More specifically, our goal is twofold. At any given point in the software development lifecycle, we estimate the effort needed for the next phase. Also, we estimate the effort for the remaining part of the software development process. Our empirical study is based on historical data from the ISBSG database. The results show a set of statistically significant correlations between: (1) the effort spent in one phase and the effort spent in the following one; (2) the effort spent in a phase and the remaining effort; (3) the cumulative effort up to the current phase and the remaining effort. However, the results also show that these estimation models come with different degrees of goodness of fit. Finally, including further information, such as the functional size, does not significantly improve estimation quality.

Keywords—software estimation, isbsg, data driven estimation

I. INTRODUCTION

Cost and effort estimation in software projects have been investigated for several years. Nonetheless, compared to other engineering fields, there are still a huge number of projects that fail in different phases due to effort prediction errors [4].

Several effort estimation models have been defined based on user experience or on previous project results but, to the best of our knowledge, no studies have tried to estimate the remaining effort after some phase based on the effort spent up to that phase. In our work, we define a data-driven model for effort estimation during all development phases.

The goal of our work is to improve existing estimation models by monitoring and estimating project costs after each development phase.

Our approach can be used to predict and monitor project effort during ongoing projects for the next development phase or for the rest of the project. Our approach will help project managers react as soon as possible and reduce project failures due to estimation errors.

We study the effort reported in the ISBSG database[5], analyzing correlations between all product development phases and between the sum of the previous phases and the following ones, and the influence of IFPUG Function Points.

Few studies investigated the effort distribution among phases such as [1][2]. In our approach, in addition to defining distribution ratios, we suggest how to calculate these ratios based on company projects.

II. THE EMPIRICAL STUDY

In this section, we first introduce our research questions. Then, we describe the dataset we used and the data processing we applied. Finally, we analyze the data and provide results.

A. *Research Questions*

The goal of this study is to explore the relationships between the efforts in product development phases so as to provide effort predictions for the following phase and the rest of the project. Thus, we investigate these first three research questions:

RQ1: Is it possible to use the effort of one phase for estimating the effort of the next development phase?

RQ2: Is it possible to use the effort of one phase to estimate the remaining project effort?

RQ3: Is it possible to use the effort spent up to a development phase to estimate the remaining project effort?

Since a variety of factors may affect effort, we also investigate whether including functional size helps obtain models that are more accurate than those obtained by using only effort data. This leads to the study's fourth research question:

RQ4: Does considering functional size, in addition to the effort for a phase, improve the effort prediction for the next phase?

B. *Data Set*

We used the International Software Benchmarking Standards Group (ISBSG) (release 11) data set. The data set allows ISBSG users to compare their projects for benchmarking and estimation purposes. It contains more than 5000 software projects collected worldwide from 1990 to 2006 from several business areas such as banking, financial, manufacturing, and others.

The data set contains several variables that can be useful for estimating the effort in different development phases. The most important variables we consider in this work are:

- *Development Type*:
  - New development projects: projects developed following the complete development lifecycle from the beginning (planning / feasibility, analysis, design, construction, and deployment)
  - Enhancement projects: changes made to existing applications where new functionality has been added, or existing functionality has been changed or deleted
  - Re-development projects: re-development of an existing application.
- *Effort per development phase:* This attribute contains the breakdown of the work effort reported via six categories:
  - Planning: preliminary investigations, overall project planning, feasibility study, and cost benefit study
  - Specifications: systems analysis, requirements, and architecture design specification
  - Design: functional, internal, and external design
  - Building: package selection, software coding and code review, package customization, unit testing, and software integration
  - Testing: system, performance, acceptance testing planning and execution
  - Deployment: release preparation for delivery, release installation for users, user documentation preparation. Note that this category is actually called "Implementation" in the ISBSG data set, but we renamed it to "Deployment" here to better clarify its meaning and differentiate it from the "Building" phase in which the software code is actually written.
- *Effort unphased*: includes all projects that specify the whole effort without making distinctions among phases.
- *Primary Programming Language:* This attribute describes the primary language used for the development. Some of the most commonly used languages by the projects are JAVA, C++, PL/1, Natural, Cobol.

We do not consider "Effort unphased" since it does not provide any information on the phases, whose effort is the main focus of our work. The ISBSG data set contains 5052 projects, 1975 of which are new developments; 2869 are enhancement projects, and the nature of 213 is not specified.

C. *Data Processing*

In this step, the ISBSG data are preprocessed to obtain the data set for effort estimation. The selection is carried out in two steps and only projects containing effort values greater than or equal to zero for each phase are considered (Fig 1).

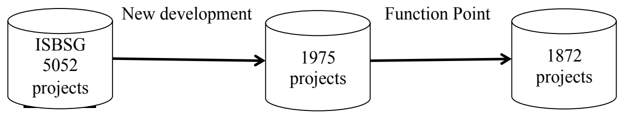

Fig. 1. Data Processing

Step 1: We selected new development projects and obtained 1975 projects. In Table I, we show the descriptive statistics of the retrieved projects, where effort is reported in terms of Person-Months (PM), as reported in the ISBSG database.

TABLE I. EFFORT DESCRIPTIVE STATISTICS PER PHASE

Phases	Descriptive statistics			
	#Projects	***Mean (PM)***	***Std.dev (PM)***	***Median (PM)***
Planning	394	687.46	1775.65	160.00
Specification	627	1102.76	2945.44	242.00
Design	374	1094.10	2743.63	330.00
Building	779	3121.34	5994.17	963,00
Testing	722	1314.44	2872.82	419.50
Deployment	482	661.10	2773.01	105.00

Step 2: We selected the new development projects containing information on functional size. We identified 1872 projects with different count approaches. IFPUG and FiSMA are the most common count approaches used in the database (see Tables II and III).

TABLE II. NUMBER OF PROJECTS WITH VALID DATA FOR EACH PHASE PER FUNCTIONAL COUNT APPROACH

	IFPUG	FISMA	COSMIC	NESMA	Other approaches
#Projects	1310	278	187	42	153

TABLE III. # PROJECTS PER PHASE VS #PROJECT FOR NEXT PHASE PER FUNCTIONAL COUNT APPROACH

	Plann. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
IFPUG	211	50	60	330	164
FiSMA	5	98	120	121	104
COSMIC	23	88	106	124	98
NESMA	10	2	2	9	5

D. *Effort Data Analysis*

In this section, we analyze the data retrieved in the data processing step.

We first apply Ordinary Least Squares regression (OLS) to investigate the existence of a correlation between a phase and the next one, a phase and the sum of the next phases, and the sum of previous phases and the sum of subsequent ones. In this analysis, we consider the project obtained in data processing – step 1.

To improve estimation quality, we also investigate the possible influence of functional size, considering the project selected in data processing – step 2. We only select projects with IFPUG as the functional count approach since only few projects of those containing information on effort per phase use a different count approach.

During the building of the OLS models, we analyze the data set and remove the outliers to prevent them from unduly influencing the OLS regression lines obtained. Specifically, we identify the outlier values that range more than 3 times standard deviation from the mean [14]. In the building of our models, we use a 0.05 statistical significance threshold, as customary in empirical software engineering studies. All of the results we show are statistically significant and the vast majority of them are actually associated with p-values smaller than 0.001. The analysis have been validated by means of a 10-fold cross-validation.

As shown in Fig. 2, the effort spent for a phase can be used to predict the effort for the next one, with a high goodness of fit seen only when predicting effort for the design phase based on the specification phase. Note that, for each pair of adjacent phases, In addition to R^2, which is the accuracy indicator typically used in OLS, we analyzed the values of MdMRE and MMRE obtained on the data set based on the model built, and the average values of MdMRE and MMRE found in the 10-fold cross validation. The use of several different indicators allow us to obtain a more complete picture of the accuracy of each model.

As shown in Figure 3, there is a good correlation between the effort of each phase and the sum of the efforts of the following ones. The result shows good improvement of the prediction compared to the one obtained considering only a phase and the next one.

Along the same lines, we investigate whether the sum of the effort of the phases up to a specified phase could be used as a predictor for the remaining effort in the project. The results described in Figure 4 and show a fairly large improvement of the correlations compared to the correlations obtained considering a phase and the next one and a phase and the sum of the next phases.

To improve the quality of the effort estimation based on the effort variable, we consider IFPUG Function Points in addition to the effort for a phase. The results show that Function Points significantly help improve the prediction only when analyzing the design phase based on the specification phase (Pearson = 0.682 vs 0.94) and the deployment phase based on the testing phase (0.361 vs 0.61). Figure 5 shows the results of the correlations per phase considering IFPUG function points.

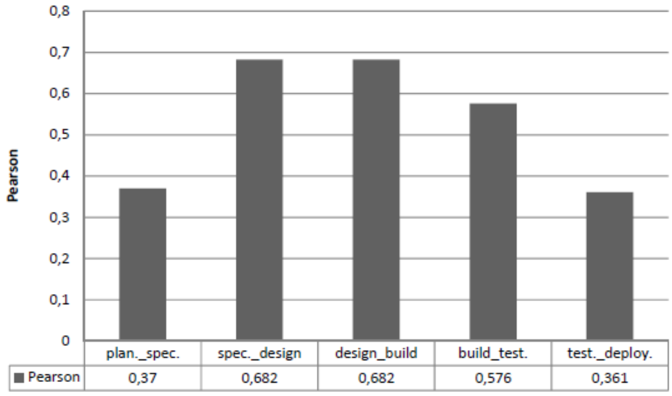

Fig. 2. Pearson correlation between the effort of one phase and that for the next phase

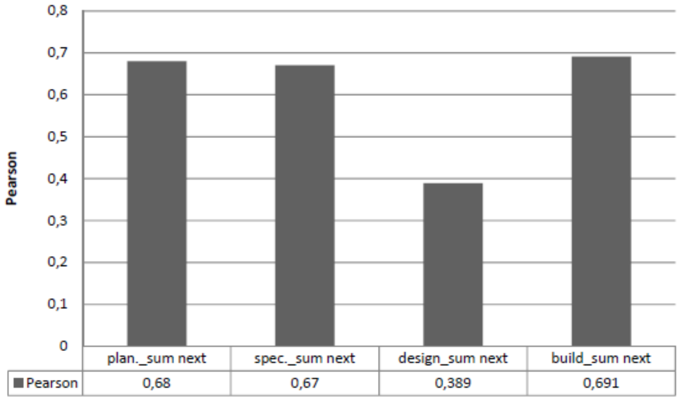

Fig. 3. Pearson correlation between the effort of the current phase and the effort of the sum of the remaining phases

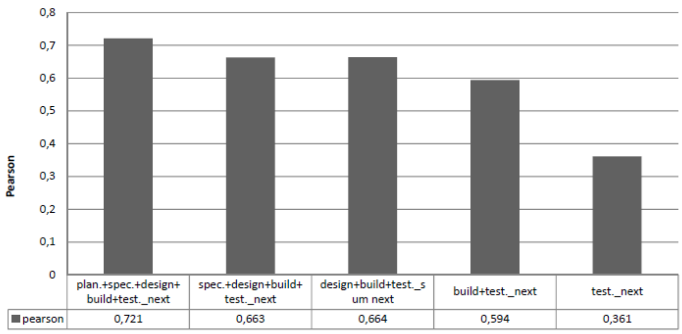

Fig. 4. Pearson correlation between the effort of the sum of previous phases and the effort of the sum of subsequent phases l

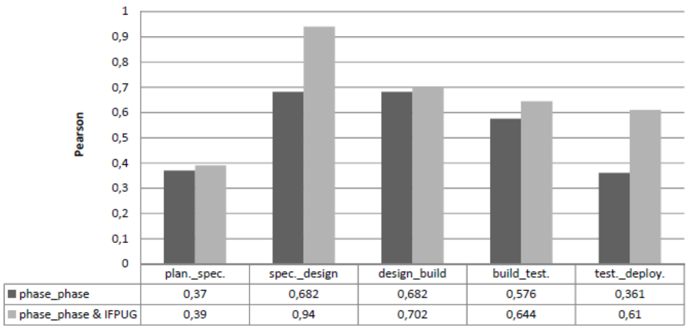

Fig. 5. Previous phase vs next phase and IFPUG Function Point correlation coefficient (Pearson)

III. DISCUSSION

Based on the results obtained in the data analysis, we can answer our research questions.

As for the RQ1, the analysis shows that it is possible to estimate the effort of the next development phase based on the effort of the previous one with an acceptable estimation error.

Analyzing RQ2, the results show that by considering the effort spent in one phase, it is also possible to estimate the remaining effort for the whole project with a similar error to that obtained when estimating only the next phase

As for RQ3, results show that it is possible to estimate the remaining project effort based on the sum of the effort spent for the previous phases. The analysis shows that the effort for the following phases can be predicted with a discrete value with varying degrees of goodness of fit (R^2 min = 0.250, R^2 max = 0.683). The Pearson coefficient increases dramatically compared to the one obtained using only one phase to estimate the remaining effort in RQ2. Running a 10-fold cross-validation, MdMRE and MdMRE shows that the goodness of fit of the models obtained may actually be somewhat questionable.

Moreover, the prediction of the remaining effort is much more accurate if we consider the total effort spent for all previous phases and that accuracy decreases if we consider a smaller number of phases.

For example, predicting the effort for the deployment phase is much more accurate if we consider the effort based on the sum of all previous phases ($R^2 = 0.515$), whereas it decreases to 0.437 when considering four previous phases, to 0.48 when considering three phases, and to 0.351 and 0.131 when considering only the previous phase.

For the last research question (RQ4), adding IFPUG function points analysis as a correlation variable does not significantly increase the goodness of fit.

Even though the level of goodness of fit is inadequate in some cases, we would like to recall that all results are statistically significant, so the effort spent up to each phase can be at least used as a predictor of the remaining effort.

Therefore, we have obtained actual models with a high goodness of fit that can be used in practice. Also, the effort values that appear as the independent variable in models with lower goodness of fit can be used as predictors.

IV. THREATS TO VALIDITY

As for internal validity, we tried to remove threats as much as possible by filtering data and removing all of those data that did not appear to be complete in the values available for phase effort.

As for external validity, the sample is somewhat heterogeneous, so the results we obtained may not be entirely applicable for specific subsets of projects, e.g., projects that use the same programming language or projects belonging to the same application domain.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an estimation model for estimating the effort needed for the next phase or the remaining effort needed to finish the project, based on historical project data.

We designed and executed an empirical study based on historical data from the ISBSG database. We studied the effort reported in 1975 projects, analyzing further correlations between all product development phases and between the sum of the previous phases and the following ones.

The results show a set of statistically significant correlations between: (1) the effort spent in one phase and the effort spent in the following one with a p-value equal to zero and an R^2 ranging from 0.134 to 0.463.

Considering only the correlation between the effort spent in one phase and the remaining effort, we still got an acceptable result for R^2 ranging from 0.25 to 0.891.

Moreover, the correlation between the cumulative effort spent up to a phase and the effort spent in the next phase increased the goodness of fit of the analysis, improving R^2 (R^2 min = 0.147, R^2 max = 0.447).

Future work will include the analysis of more variables, including programming languages and the clustering of projects, in order to define a more accurate model for specific ranges of variables.

ACKNOWLEDGMENT

This work was partly supported by the project “Metodi, tecniche e strumenti per lo sviluppo di sistemi software di qualità”, funded by the Università degli Studi dell’Insubria and by the FP7 Collaborative Project S-CASE (grant agreement no, 610717), funded by the European Commission.

REFERENCES

[1] B.Boehm, “Software Engineering Economic”s. Englewood Cliffs, NJ:Prentice-Hall, 1981. ISBN 0-13-822122-7

[2] 0Z. Jiang, P. Naudé, and B. Jiang, “The Effects of Software Size on Development Effort and Software Quality,” International Journal of Computer and Information Science and Engineering, vol. 1, no. 4, 2007, pp. 230-234.

[3] L. Mieritz, “Survey Shows Why Projects Fail”. Gartner report.Retrieved from Gartner Database, 2012

[4] The International Software Benchmarking Standards Group (ISBSG). http://www.isgsg.org. Last seen may 2014