World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:1, 2007

The Factors Significant to Software Development Productivity

Zhizhong Jiang, and Craig Comstock

Abstract**—**The past decade has seen enormous growth in the amount of software produced. However, given the ever increasing complexity of the software being developed and the concomitant rise in the typical project size, managers are becoming increasingly aware of the importance of issues that influence the productivity levels of the project teams involved. By analyzing the latest release of ISBSG data repository, we report on the factors found to significantly influence the productivity among which average team size and language type are the two most essential ones. Building on this we present an original model for evaluating the potential productivity during the project planning stage.

Keywords**—**ISBSG, Linear Model, Productivity, Software Engineering.

I. INTRODUCTION

SOFTWARE has become the key element in the evolution of computer-based systems and products. Over the past 50 years, software has evolved from a specialized problem solving and information analysis tool to an industry in itself [1]. The two primary problems in software development that have yet to be solved satisfactorily are making systems cost effective and of higher quality. A major obstacle to solve the problem of cost effective is the intrinsic complexity in developing software. Improving the productivity is an essential part of making system cost effective [2].

Along the progress of software development there are extensive researches in the measurement of the development productivity. Humphrey and Singpurwalla [3] use the statistical techniques of time series analysis to predict the productivity of software development with reasonable accuracy. Blackburn et al. [4] imparts a global survey of software developers on improving speed and productivity of software development. This paper charts a clear path for the appraisal of software productivity. Kitchenham et al. [5] proposes a new productivity measurement method and presents an example of its use for web applications. Whereas these studies focus mainly on the measurement of the productivity, there are unfortunately very few investigations on the elements that influence the productivity. The main reason is the unavailability of the large database for the study. It is now feasible to advance the quantitative researches on the productivity with the latest release of ISBSG data repository (Release 9, published in 2005). Covering 3024 projects ever developed, this release is conceived as one of the largest and most important project databases worldwide. In this study we focused on analyzing this database with advanced statistical methods. The resultant model discovered the major factors significant to the productivity, and provided an approach for its evaluation. The statistical results showed that average team size, language type (3GL etc.), development platform (multi platform etc.) and development techniques (event modeling etc.) are the significant factors for the productivity among which the first two factors are the dominant parts.

The paper is organized as follows: section II gives an overview of ISBSG and the early studies based on its data releases; section III introduces the latent factors significant to the productivity; section IV and V are the detailed model developing processes; section VI presents full discussions on the derived model; section VII is the conclusion of this study.

Manuscript received November 30, 2006. This research was supported by ISBSG (International Software Benchmarking Standards Group) for providing the data.

Zhizhong Jiang, was with Department of Statistics, University of Oxford, 1 South Parks Road, OX1 3TG, U.K. He is now with Manchester Business School, The University of Manchester, Booth Street West, Manchester, M15 6PB, UK (phone: +44(0)8708328157; fax: +44(0)1612756596; e-mail: zhizhong.jiang@brasenose.oxon.org, Zhizhong.Jiang@postgrad.mbs.ac.uk).

Craig Comstock, was with Harvard University. He is now with University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD UK (e-mail: craig.comstock@lmh.ox.ac.uk).

II. BACKGROUND

The ISBSG (International Software Benchmarking Standards Group) is a not-for-profit organization established in the late 1990s. Its primary task is to assemble the largest publicly accessible database of software projects available to IT professionals today. The last release (Release 9) contains 3024 projects, some dating back to the early 90's. The data kept on each project includes up to 90 metrics or descriptive pieces of information, including the size of project, number of developers, organization type, platform, number of users, programming language and database used, man-hours worked on the project by phase, and major defects that made it to production.

Several studies on the analysis of ISBSG data repository appeared in the literature. Lokan [6] describes the data repository in detail, and summarizes several findings that have emerged from analyses and researches using the repository. Oligny et al. [7] presents a non-linear relationship between project duration and effort. However, these studies are only based on the limited capacity of the early data release.

III. Data Description

Whereas the latest release of ISBSG data contains a great number of parameters recording each project developed, we only introduce the ones (including the variable PDR which is a direct measurement of the software development productivity) that purportedly have effects on the productivity.

- *Normalized Productivity Delivery Rate (PDR)*

PDR is the parameter which directly reflects the productivity level. Practically it is calculated from *Normalized Work Effort* divided by *Adjusted Function Points*. In the database *Normalized Work Effort* is recorded as the normalized total hours spent on the software development, and *Adjusted Function Points* is the gauge for the project size. This is an inverse measure of the productivity in that the larger PDR, the smaller is the productivity.

Function point analysis (FPA) provides a standardized methodology for measuring project size. Abran and Robillard [8] introduce an empirical study of FPA measurement processes. Since the projects in the ISBSG data repository apply different criterions (IFPUG, NESMA, MARK II etc.) for the calculation of function point, the functional size is adjusted by an adjustment factor, and the resultant adjusted size is reported in *Adjusted Function Points* (AFP).

- *Average Team Size*

It is the average number of people that worked on the project through the entire development process. This factor presumably impacts the development productivity in that considerably different number of developers is believed to result in dissimilar productivity levels.

- *Language Type*

It defines the language type used for the project (2GL, 3GL, 4GL, or ApG). 2GL (second-generation languages) are machine dependent assembly languages; 3GL (FORTRAN, C etc.) are high-level programming languages; 4GL (SQL etc.) is more advanced than traditional high-level programming languages; ApG (Application Generator) is the program that allows programmers to build an application without writing the extensive code. In practice all 4GL languages are designed to reduce programming efforts. Thus language type would be another latent factor significant to the productivity.

- *Development Type*

Describes whether the software development was a new development, enhancement or re-development.

- *Development Platform*

Defines the primary development platform. Each project is classified as PC, Mid Range, Main Frame or Multi platform.

- *Development Techniques*[1]

Techniques used during software development (e.g. Waterfall, Prototyping, Data Modeling etc). A large number of projects make use of various combined techniques.

- *Case Tool Used*

Indicates if the project used any case tool (yes or no).

- *How Methodology Acquired*

Describes whether the development methodology was Traditional, Purchased, Developed In-house, or a combination of Purchased and Developed.

Three points needs to be mentioned here:

1) Since particular programming languages (e.g. Java, C++, VB) belongs to one of the language types (e.g. 3GL, 4GL), we did not take into account the factor *Primary Programming Language* which contains numerous programming languages. Otherwise redundancy is introduced into the study.
2) The ISBSG data Release 9 contains one parameter *Total Defects Delivered* which records the total number of defects reported in the first month of use of the software. A majority of the entries for this parameter have missing values. We tested whether software defects affect the productivity but found there is no correlation between them. The exclusion of this factor could save lots of degrees of freedom for the later regression analysis.
3) It is conceivable that senior software developers are more proficient and productive than junior developers. ISBSG data repository does not report this and assumes the developers are all well-qualified practitioners.

Since Release 9 has very sparse records before the year 1994, our analysis starts from 1994.

IV. The Significant Factors

Examining the proposed factors in section III, we realize that *Average Team Size* is the only continuous variable and all the other factors are categorical variables. So the underlying model for the analysis deals with two continuous variables (*Average Team Size* and the dependent variable *PDR*) and six categorical variables (*Language Type*, *Development Type*, *Development Platform*, *Development Techniques*, *Case Tool Used*, and *How Methodology Acquired*)[2].

We first examine the relationship between PDR and TeamSize using scatter plot which can explore the possible relationship between two variables. It is important to point out here that the original data of PDR and TeamSize are extremely skewed. We take the natural log transformation (with base e)

[1] There are 28 different development techniques used in the 3024 projects. The main ones are Waterfall (524), Data Modelling (383), Process Modelling (254), JAD (Joint Application Development) (188), Prototyping (179), Regression Testing (155), Object Oriented Analysis & Design (98), Business Area Modelling (93), RAD (Rapid Application Development) (91), Event Modelling (77).

[2] To make it easy to interpret, we abbreviate these variables as: TeamSize for *Average Team Size*, LangType for *Language Type*, DevType for *Development Type*, Platform for *Development Platform*, CaseTool for *Case Tool Used*, Method for *How Methodology Acquired* (development methodology).

to make the data look normally distributed. The Fig. 1 below is the scatter plot for the two variables log(PDR) and log(TeamSize). It demonstrates that the relationship between them is close to linear. Accordingly we apply linear model to investigate these two variables and the other six categorical variables discussed before.

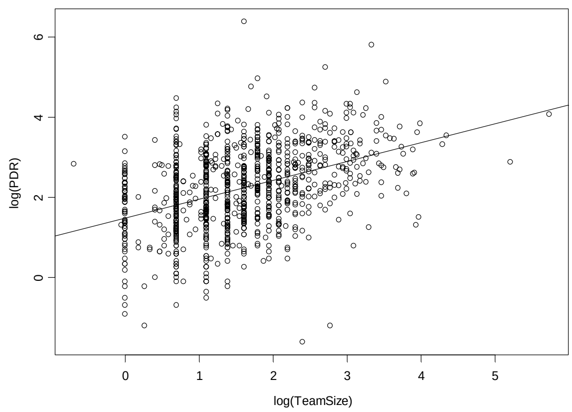

Fig. 1 The scatter plot of log(TeamSize) and log(PDR)

While the current ISBSG data repository has 3024 projects, the database contains plenty of missing values. This study accounted for the effect of missing data while making statistical analysis. Another challenging work is how to handle the various combined uses of different development techniques. We manage it by separating each of the main development techniques as one single factor and take into account all of the second-order interactions[3] between them. In S-plus[4] we run multiple linear regression with the core data and obtained the final explanatory variables significant to the dependent variable log(PDR). In other words these explanatory variables are significant factors for the productivity. S-plus produces the regression results, and the resultant model is as follows:

$$\log(PDR) = 2.8400 + 0.3659\times\log(TeamSize) - 0.6872\times I(3GL) - 1.2962\times I(4GL) - 1.3225\times I(ApG) - 0.1627\times I(MR) - 0.4189\times I(Multi) - 0.3201\times I(PC) - 0.4280\times I(OO) - 0.2812\times I(Event) + 0.7513\times I(OO{:}Event) - 0.2588\times I(Business) - 0.0805\times I(Regression) + 1.0506\times I(Business{:}Regression)$$

Though it looks complex, it is easy to understand with some interpretations:

1) TeamSize stands for average team size for the development; 3GL indicates using *3GL* as the language type, similarly for 4GL and ApG; MR means the project uses *Mid Range* as the primary development platform, Multi and PC represent using *Multi* platform and *PC* platform for the project respectively; The remained terms are related to the development techniques: OO is *Object Oriented Analysis & Design*, Event is *Event Modeling*, Regression is *Regression Testing*, and Business is *Business Area Modeling*.
2) log( ) is the natural log with base e; The value of the mapping function $I(\cdot)$ is 1 if the relevant technique in the parentheses is used, otherwise it is 0 (that is, $I(A) = 1$ if A is used, otherwise $I(A) = 0$); The operator **:** defines the interaction between two (or more) variables. Accordingly there is no interaction if there is only one particular development technique used. I(OO: Event) is 1 if and only if both OO and Event Modeling are used.
3) The default development platform is *Main Frame*, and the default language type is *2GL*. In other words, if the project uses the default platform and the default language type, then the above expression is reduced to:

$$\log(PDR) = 2.8400 + 0.3659\times\log(TeamSize) - 0.4280\times I(OO) - 0.2812\times I(Event) + 0.7513\times I(OO{:}Event) - 0.2588\times I(Business) - 0.0805\times I(Regression) + 1.0506\times I(Business{:}Regression)$$

Other development methods are adjusted by their related coefficients.

4) The significance levels for the variables are based on the reported p-values[5]. Those with p-values smaller than 5% is deemed as statistically significant [9].

Project planners can apply the above model to estimate the productivities beforehand. For example, a certain project is designed to have an average of 8 team members, uses 4GL, Multi platform, and the combined development techniques of Business Area Modeling and Regression Testing, then

$$\log(PDR) = 2.8400 + 0.3659\times\log(8) - 1.2962\times 1 - 0.4189\times 1 - 0.2588\times 1 - 0.0805\times 1 + 1.0506\times 1 = 2.60, \quad PDR = 13.4$$

So the estimated Normalized Productivity Delivery Rate is 13.4, which means to deliver one function point it needs 13.4 person hours.

V. Model Checking

Usually it is required to check the goodness of fit of the fitted model. The effective and practical way is to examine the conformability of the fitted (or predicted) values to the observed (or recorded) values. Fig. 2 displays the diagnostic

3 Second-order interaction is the interrelation between two variables.

4 S-plus is an advanced programming language for statistical analysis and graphics.

5 The p-values for the final significant factors.

Type III Sum of Squares

	Df	Mean Sq	F Value	Pr(F)
log(TeamSize)	1	60.05693	84.68682	0.0000000
LangType	3	16.89406	23.82247	0.0000000
Platform	3	2.64894	3.73529	0.0111654
Regression	1	4.11929	5.80865	0.0162655
Business	1	1.50692	2.12491	0.1454755
OO	1	0.06963	0.09819	0.7541315
Event	1	0.21505	0.30324	0.5820732
Business:Regression	1	6.13919	8.65692	0.0033916
OO:Event	1	3.62298	5.10880	0.0241839
Residuals	566	0.70916		

plot of the observed values (y-axis) against the fitted values (x-axis). From the plot we can see that the fitted values conform well to the observed values.

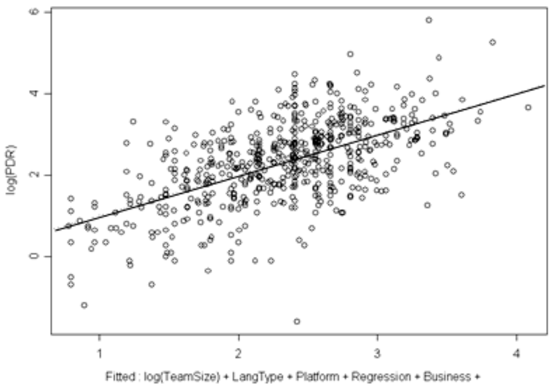

Fig. 2 Diagnostic plot of the observed values against the fitted values

Furthermore in linear model ($y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + \varepsilon_i$, i=1, 2, …,n), it is assumed that the residuals ε_i are normally distributed with zero mean. In our study we applied Q-Q plot to check the normality of the residuals. The approximately straight line proves that the residuals are normally distributed. Thus the normal assumption is validated.

Finally, we checked the diagnostic plot of the residuals against the fitted values. Fig. 3 below shows the points evenly scatter along the zero residual line, and there exists no trace of heteroscedasticity.

Therefore the linear model we applied is feasible. We now turn to the discussions based on the model we derived.

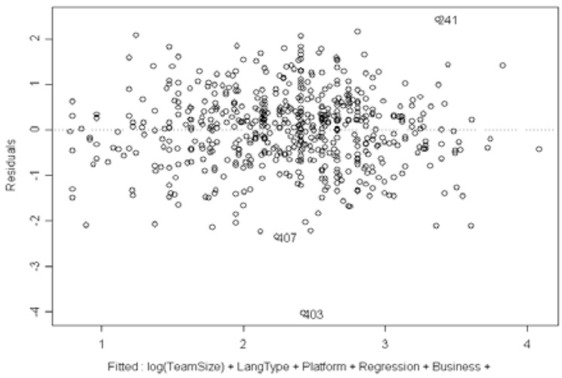

Fig. 3 Diagnostic plot of the residuals against the fitted values

VI. Discussions on the Productivity Evaluation Model

In section III we mentioned *Normalized Productivity Delivery Rate* (PDR) is defined as *Normalized Work Effort* divided by *Adjusted Function Points*. PDR is an inverse measure of the productivity in that the smaller PDR, the higher is the productivity. Looking at the productivity evaluation model and the reported p-values for the variables, we can generalize:

1 *Average Team Size*, *Language Type*, *Development Platform*, and *Development Techniques* are significant factors for the productivity. Among these factors *Average Team Size* and *Language Type* are the two most essential ones (their p-values are less than 0.01% which are extremely significant). The use of *Case Tool*, *Development Type*, *Development Methodology*, and other unmentioned techniques have no considerable effects on the productivity.

2 The increase of *Average Team Size* will lead to lower productivity.

According to the model log(PDR) and log(TeamSize) have a positive linear relationship. So the bigger average team size will bring on larger value of log(PDR) and hence lower productivity.

3 Different development methods have varied significances to the productivity. To improve the productivity (control other factors):

- 3.1 ApG and 4GL are superior languages than 3GL and 2GL;
- 3.2 Multi platform is slightly better than PC followed by Mid Range and Main Frame;
- 3.3 The single use of Event Modeling or Object Oriented Analysis & Design can produce higher productivity. However, their combined use can only neutralize this effect;
- 3.4 The combined use of Business Area Modelling and Regression Testing is adverse to the improvement of the productivity.

We notice the more negative of the coefficients of the mapping function $I(\ )$, the smaller the value of log(PDR), and hence the more productive of their corresponding development approaches. This is compared with the default development techniques. As we discussed in section IV the default development methods used in the model are *Main Frame* (for development platform) and *2GL* (for language type). Accordingly, different development methods are compared by their matching coefficients of $I(\cdot)$ with the default methods acted as the benchmarks.

We first examine the four language types. The related coefficients of $I(\cdot)$ for 2GL, 3GL, 4GL and ApG are 0, −0.6872, −1.2962 and −1.3225 respectively. So 4GL and ApG are more capable of reducing the value of log(PDR) than 2GL and 3GL. This means 4GL and ApG are more productive than 2GL and 3GL. This complies with the rule that in principle 4GLs are designed to reduce programming efforts, the time it takes to develop software.

By the same reasoning we could see Multi platform (−0.4189) is slightly more productive than platform PC (−0.3201) which is better than Mid Range (−0.1627) with Main Frame as the default platform (0).

As for the development techniques, the single use of Event Modelling or Object Oriented Analysis & Design can reduce log(PDR) by −0.2812 and −0.4280 respectively. However, the result will be −0.4280−0.2812+0.7513=0.0421 if they are used together. This greatly increases the value of log(PDR) and thus it loses the effect of boosting the productivity. Similarly, the conjoint use of Business Area Modelling and Regression Testing will substantially enhance the value of log(PDR) (−0.2588−0.0805+1.0506=0.7113). Therefore project developers should avoid using them together for the purpose of higher productivity.

VII. Conclusion

This study worked on the latest release of ISBSG data repository which is deemed as the most complete database in software development globally. By running regression analysis in S-plus this research found the factors significant to software development productivity. They are the two most critical factors average team size, language type as well as the other two factors development platform and development techniques. A productivity evaluation formula is presented to estimate the productivity during project planning stage. The model reveals that ApG and 4GL are more productive than 3GL. Other discussions are also given on the development platforms and development techniques.

References

[1] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. London: McGraw-Hill, 2001, pp. 15.

[2] S. T. Albin, *The Art of Software Architecture: Design Methods and Techniques*. New York: Wiley, 2003, pp. 6.

[3] W. S. Humphrey and N. D. Singpurwalla, N.D. 1991. "Predicting (individual) software productivity," *IEEE Trans. Software Engineering*, vol. 17, pp. 196 – 207, Feb.1991.

[4] J. D. Blackburn, G. D. Scudder, and L. N. Van Wassenhove, "Improving speed and productivity of software development: a global survey of software developers," *IEEE Trans. Software Engineering*, vol. 22, pp. 875 – 885, Dec. 1996.

[5] B. Kitchenham and E. Mendes, "Software productivity measurement using multiple size measures," *IEEE Trans. Software Engineering*, vol. 30, pp. 1023 – 1035, Dec. 2004.

[6] C. Lokan, "Statistical analysis of ISBSG data and FPA analysis," *Proceedings of 1999 Australian Conference on Software Metrics*, Nov. 1999.

[7] S. Oligny, P. Bourque, A. Abran and B. Fournier, "Exploring the relation between effort and duration in software engineering projects," *World Computer Congress*, pp.175-178, Aug. 2000.

[8] A. Abran and P. N. Robillard, "Function Points Analysis: An Empirical Study of Its Measurement Processes," *IEEE Trans. Software Engineering*, vol. 22, pp. 895-910, Dec. 1996.

[9] M. J. Crawley, *Statistical Computing: An Introduction to Data Analysis using S-Plus*. Chichester: John Wiley & Sons Ltd, 2002.