International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:3, 2007

# The Variation of Software Development Productivity 1995-2005

Zhizhong Jiang, Peter Naudé, and Craig Comstock

***Abstract*****—**Software development has experienced remarkable progress in the past decade. However, due to the rising complexity and magnitude of the project the development productivity has not been consistently improved. By analyzing the latest ISBSG data repository with 4106 projects, we discovered that software development productivity has actually undergone irregular variations between the years 1995 and 2005. Considering the factors significant to the productivity, we found its variations are primarily caused by the variations of average team size and the unbalanced uses of the less productive language 3GL.

***Keywords*****—**Productivity, Programming Languages, Software Engineering, Team Size.

## I. Introduction

THE past decade has seen increasing complexities and costs of software development. How to improve development productivity has been an ongoing concern for the project managers. Early studies appear to have recognized the significance of improving the productivity. Howes [1] proposes a methodology to manage software projects for maximum productivity, while Loesh [2] provides an approach to define standard design templates for the software functions that encourages repeatability, thereby improving productivity. Blackburn et al. [3] impart a global survey of software practitioners on improving speed and productivity of software development. Based on the Analytic Hierarchy Process, Finnie et al. [4] describe a model that allows prioritization of the driving forces to achieve higher software development productivity. Jiang and Comstock [5] identify the factors affecting productivity levels and present an original model for its evaluation.

Unfortunately, despite the attention that has been given to it, the productivity of software development has not improved consistently. This needs to be seen in the light of the impressive improvements in hardware speed and network capacity [6]. Gross measures presented in the literature indicate that software productivity has been declining more rapidly than any other industry [7].

With the availability of the large ISBSG database recording numerous projects developed worldwide, we can explore the trend of software development productivity over time. Whereas many scholars still argue whether the productivity has really declined [8], our analysis reveals that the productivity has experienced irregular variations of decline and rise in the past decade, and there is no sign of imminent improvement. Further, we find that average team size also varies over time and there has been unstable use of the less productive language 3GL. As the major determinants of software development productivity, these two factors explain most of the variation of the productivity.

The paper is organized as follows: section II briefly introduces the data and its provider ISBSG; sections III and IV illustrate the variations in software development productivity and average team size over the years; section V shows the irregular use of the third-generation language (3GL) in software development; section VI discusses the factors leading to the productivity variation; and finally section VII presents the conclusion to the study.

## II. Background

Established in 1997 the International Software Benchmarking Standards Group (ISBSG) is a nonprofit organization, whose mission is to help improve the management of IT resources through the provision and exploitation of public repositories of software engineering knowledge. The ISBSG has established and maintained two repositories of software metrics: software development and software maintenance.

The latest release of ISBSG data repository (Release 10) contains information on 4106 projects. The data kept on each project includes up to 90 metrics or descriptive pieces of information. Lokan [9] describes the early data repository in detail, and summarizes several findings that have emerged from analyses and studies using the repository. For our purpose of this study, we mainly focus on six metrics as *Normalized Productivity Delivery Rate* (an inverse measure of the productivity), *Average Team Size* (average number of developers for the whole project), *Development Language* (specific language type used, e.g. 3GL, 4GL), *Development Platform* (e.g., Multi platform, Main Frame), *Implementation Date* (actual date of implementation of the software), and *Development Techniques* (e.g. waterfall, prototyping).

Manuscript received March 5, 2007. This research was supported by the ISBSG (International Software Benchmarking Standards Group) for providing the data.

Zhizhong Jiang was with Department of Statistics, University of Oxford. He is now with University of Manchester, Booth Street West, Manchester, M15 6PB, UK (phone:+44(0)8708328157; fax: +44(0)1612756596; e-mail: Zhizhong.Jiang@postgrad.mbs.ac.uk, zhizhong.jiang@brasenose.oxon.org).

Peter Naudé was with University of Bath, UK. He is now a Professor in Manchester Business School, University of Manchester, Booth Street West, Manchester, M15 6PB, UK (e-mail: pete.naude@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).

It is necessary to point out that the ISBSG data repository contains one parameter—*Data Quality Rating*, which indicates the reliability of the data recorded. It has four grades *A*, *B, C*, and *D*. While the data with quality ratings *A*, *B* and *C* are assessed as being acceptable, little credibility can be given to any data with rating *D*. Therefore, to obtain meaningful results we excluded those cases with quality rating *D*.

## III. Variations in the Productivity

In the ISBSG data repository the development productivity for a particular project is represented by one parameter—*Normalized Productivity Delivery Rate* (PDR). It is defined as *Normalized Work Effort* (normalized total working hours for the development) divided by *Adjusted Function Points* (project size). Clearly PDR is an inverse measure of the productivity in that the larger value of PDR actually signifies lower productivity.

Since the yearly PDR data are highly skewed, natural logarithmic transformation (with base e) is applied to make the data normally distributed. The average PDR is obtained by calculating the mean of log(PDR) and converting it back to the PDR with exponential transformation. For some years the data are not normally distributed even with the log transformation. In these cases the median of the original PDR data is taken as the average PDR for that particular year. The annual averages of PDR between 1995 and 2005 are shown in Table I below. The column *Number of Observations* gives the number of projects recorded for the related year. The movement of the average PDR is plotted in Fig. 1.

TABLE I
The Average of PDR 1995-2005

| Year | Average PDR | Number of Observations |
|---|---|---|
| 1995 | 9.0 | 123 |
| 1996 | 10.4 | 102 |
| 1997 | 7.4 | 132 |
| 1998 | 7.7 | 264 |
| 1999 | 9.0 | 419 |
| 2000 | 9.3 | 544 |
| 2001 | 8.5 | 235 |
| 2002 | 10.5 | 335 |
| 2003 | 8.0 | 182 |
| 2004 | 8.2 | 257 |
| 2005 | 13.6 | 231 |

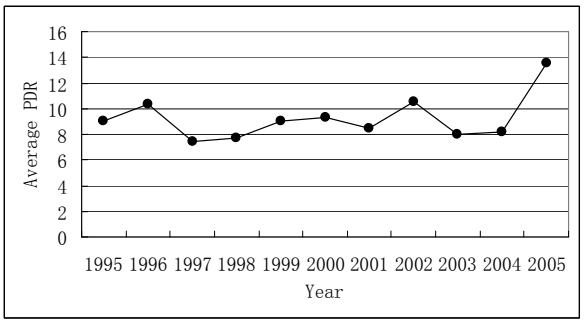


Fig. 1 The trend of annual average of PDR 1995-2005

The averages drawn in Fig. 1 demonstrate that the annual average PDR is relatively unstable going through a series of changes. PDR rises in 1996, and then it suddenly drops in 1997 where the lowest PDR (7.4) occurs. Thereafter it continues to increase until year 2000. After that it varies considerably, dropping in 2001 but reaching its highest level to date in 2002. This is followed by two low years, before it reaches the peak (13.6) in 2005.

Recalling that PDR is an inverse measure of software development productivity, we can infer that the productivity has experienced irregular variations of rise and decline over the past years. The year 1997 saw the highest productivity levels. However, and rather alarmingly, the lowest level of productivity arises in the latest year 2005 when, to deliver one function point, it actually needs 13.6 man hours. We therefore conclude that software development productivity has not been improving over time, and seek to explain this phenomenon.

## IV. Variation of Average Team Size

To explain the irregular variations in software development productivity we can explore the factors which influence the productivity. Research by Jiang and Comstock [5] identified that average team size and development language (e.g. 3GL, 4GL) are the two most significant factors influencing the productivity. As a result we turn to the study of these two factors.

In the data repository the variable *Average Team Size* has a very skewed distribution for the years 1995-2005. Therefore logarithmic transformation is taken, and like PDR the average value is obtained either from the mean or the median of the data. The annual average team size is displayed in Table II. Since *Average Team Size* has very sparse data in the database particularly for 2004 and 2005 (13 and 12 cases respectively), the data for these two years are merged with the new label 2004(5).

TABLE II
THE AVERAGE TEAM SIZE 1995-2005

| Year | Average Team Size | Number of Observations |
|---|---|---|
| 1995 | 3.0 | 63 |
| 1996 | 3.4 | 26 |
| 1997 | 3.0 | 61 |
| 1998 | 4.0 | 119 |
| 1999 | 5.0 | 219 |
| 2000 | 6.0 | 174 |
| 2001 | 6.3 | 61 |
| 2002 | 7.0 | 38 |
| 2003 | 4.6 | 38 |
| 2004(5) | 6.2 | 25 |

The data are further displayed in Fig. 2 which shows the extent to which the annual average team size has varied over time. Average team size was very low in 1995 and 1997 (3.0). However, it constantly rises until the year 2002 where the maximum value emerges (7.0). After one remarkable drop in 2003, average team size goes up again in 2004 and 2005. Clearly the overall trend in average team size is upwards. This confirms well with the view that due to the growing intricacy of software development, projects requires more and more developers to work together.

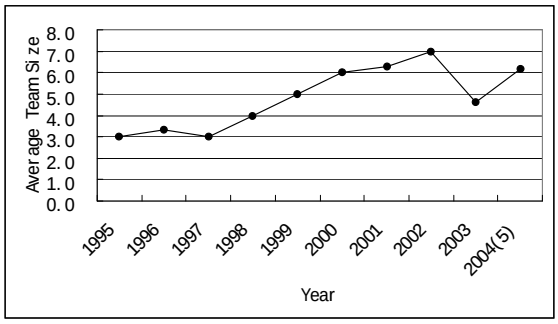


Fig. 2 The variation of average team size 1995-2005

## V. THE VARIABLE USE OF DIFFERENT LANGUAGE TYPES

The second key factor essential to the productivity is the development language. There are four language types existent in the data repository: 2GL, 3GL, 4GL, and ApG (Application Generator). Most of the projects have utilized 3GL or 4GL as the development language. Although 4GL has proved to be much more productive than 3GL (see [5] and [10]), it has not been greatly used. Table III below gives the breakdown of the different language types used across the years.

TABLE III
THE BREAKDOWN OF FOUR LANGUAGE TYPES 1995-2005

| **Year** | **2GL** | **3GL** | **4GL** | **ApG** | **Total** |
|---|---|---|---|---|---|
| 1995 | 0 | 51 | 50 | 5 | 106 |
| 1996 | 0 | 48 | 18 | 8 | 74 |
| 1997 | 0 | 48 | 39 | 11 | 98 |
| 1998 | 1 | 117 | 68 | 10 | 196 |
| 1999 | 1 | 241 | 117 | 5 | 364 |
| 2000 | 3 | 329 | 187 | 14 | 533 |
| 2001 | 2 | 150 | 82 | 4 | 238 |
| 2002 | 0 | 262 | 84 | 1 | 347 |
| 2003 | 0 | 141 | 65 | 2 | 208 |
| 2004 | 0 | 139 | 162 | 0 | 301 |
| 2005 | 1 | 184 | 80 | 13 | 278 |

Table III displays the two development languages 2GL and ApG are rarely used. Besides, 4GL and ApG have been shown to be both equally more productive than 3GL [5]. Therefore, to see the effect of language type on the productivity variation, we can just examine the frequency of use of 3GL over the years. Fig. 3 below represents the percentage of 3GL used as the development language between 1995 and 2005. This indicates that there has been variable use of the language 3GL over the years. Year 2002 saw the most extensive use of 3GL (75.5%). For most of the years over 60% the projects applied 3GL as the development language. Even in 1995, 1997 and 2004 there were still some 50% of the projects using 3GL (48.1%, 49.0% and 46.2% respectively). Therefore, we can conclude that 3GL has been the most prevalent development language in the past. Given its broad use in 2005 (66.2%) 3GL still remains popular today.

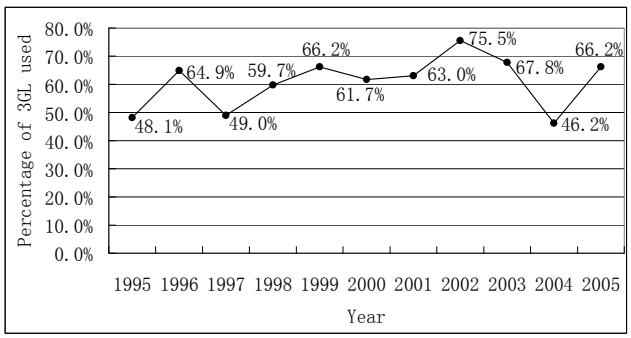


Fig. 3 The percentage of 3GL used 1995-2005

## VI. THE EXPLANATION OF THE PRODUCTIVITY VARIATION

Given that average team size and development language have been identified as the two key factors that influence the productivity level, the variations of the productivity are most likely influenced by these two factors. We examine this by

incorporating the preceding three figures in Fig. 4 below[1]. As we mentioned before, the variable *Average Team Size* has very sparse data for 2004 and 2005, and the resultant average is produced by fusing the data in these two years. Similarly, to give systematic comparisons, we acquired the average of PDR and 3GL respectively for these two years.

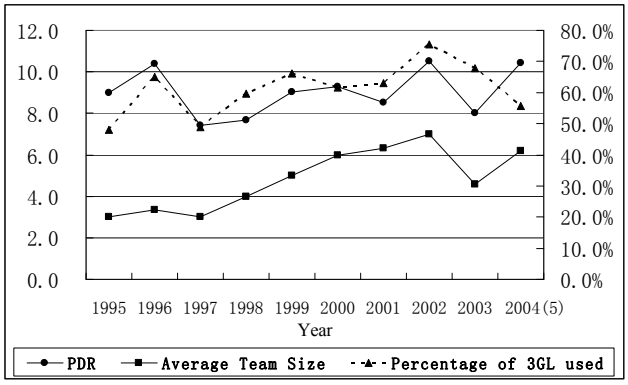


Fig. 4 The movements of PDR, average team size and use of 3GL 1995-2005

Surprisingly the three curves in Fig. 4 demonstrate rather similar patterns. Starting from 1995 they rise simultaneously in 1996, then decline together in 1997 when the low values occurred. In year 2002 all three curves reach the peak, and then they drop down together in 2003. We can generalize their patterns of change year by year in Table IV.

TABLE IV
THE MOVEMENTS OF PDR, AVERAGE TEAM SIZE AND USE OF 3GL 1995-2005

| Period | PDR | Average Team Size | Use Percentage of 3GL |
|---|---|---|---|
| 1995-1996 | ↑ | ↑ | ↑ |
| 1996-1997 | ↓ | ↓ | ↓ |
| 1997-1998 | ↑ | ↑ | ↑ |
| 1998-1999 | ↑ | ↑ | ↑ |
| 1999-2000 | ↑ | ↑ | ↓ |
| 2000-2001 | ↓ | ↑ | ↑ |
| 2001-2002 | ↑ | ↑ | ↑ |
| 2002-2003 | ↓ | ↓ | ↓ |
| 2003-2004(5) | ↑ | ↑ | ↓ |

[1] In Fig. 4 *Average Team Size* and PDR share the left y-axis and 3GL is represented by the right y-axis.

Table IV exhibits that for most of the periods PDR, average team size and use percentage of 3GL have the same patterns of change (increasing or decreasing simultaneously). This is consistent with the model developed by Jiang and Comstock [5]. In their model average team size and PDR have a positive relationship, so that the growth of average team size will lead to larger PDR (lower productivity). Besides, the increasing use of 3GL can only result in lower productivity (larger PDR).

There are two periods of time (1999-2000, 2003-2004(5)) when the three curves have incongruous trends. For both periods, PDR and average team size increase while the use percentage of 3GL declines. As both average team size and use percentage of 3GL significantly affect productivity, when they move in opposite directions, the productivity is decided in great part by the one with greater variation. In this case the productivity is also likely to be modulated by other important factors such as the use of different development platforms and various development techniques, although these factors are not the determinants of the productivity [5].

Finally, we can identify one period (2000-2001) in which PDR declines while both average team size and use percentage of 3GL slightly increase. This indicates that the productivity has actually risen though it is expected to slightly go down.

To explain this we can inspect other factors that influence productivity levels. There are four main factors essential to the productivity–*Average Team Size, Development Language , Development Platform* and *Development Techniques*, among which the first two are the most critical ones [5]. We now make comparisons between 2000 and 2001 focusing on *Development Platform* and *Development Techniques*.

For the four development platforms, platforms PC and Multi are proved to be most productive while platforms Mid Range and Main Frame are least productive [5]. Table V illustrates the usage of the four platforms for the two years 2000 and 2001. In 2000 there are nearly two thirds (316/494) of the projects employed Main Frame and Mid Range, but in 2001 there were just 54% (134/246) of the projects using them. Therefore, compared to 2001 the wide-scale uses of the less productive development platforms in 2000 restrained the improvement of the productivity.

TABLE V
THE BREAKDOWN OF FOUR DEVELOPMENT PLATFORMS 2000-2001

| Year | Main Frame | Mid Range | PC | Multi | Total |
|---|---|---|---|---|---|
| 2000 | 215 | 101 | 134 | 44 | 494 |
| 2001 | 99 | 35 | 47 | 65 | 246 |

We now examine the second factor *Development Techniques*. For the many development techniques recorded in the ISBSG data repository, there are only four techniques significant to the productivity, these being Business Area Modeling with Regression Testing, Event Modeling with Object Oriented Analysis & Design (OO) (see [5]). Table VI lists the number of uses of these four techniques in 2000 and 2001. Compared to the large number of cases in these two years, the use frequency for most of these development techniques is rather insignificant. Furthermore, although Regression Testing is applied much more in 2000 than in 2001, its usage alone proves to be negligible in determining

productivity [5]. Whereas the solo use of OO or Event Modeling is useful for higher productivity, their interaction neutralizes this effect [5]. Thus, in 2000 the cases that effectively employ OO or Event Modeling for better productivity are reduced to 30 (43-13) and 11 (24-13) respectively.

Therefore, compared with 2000 the improvement of the productivity in 2001 is mainly due to the broader uses of the more productive development platforms. However, we hold that the accumulative effects of other insignificant factors on productivity improvement between 2000 and 2001 are likely.

We now conclude the variations of the productivity in the past are largely caused by the changes of average team size and the varying uses of different language types.

TABLE VI
THE BREAKDOWN OF THE DEVELOPMENT TECHNIQUES SIGNIFICANT TO THE PRODUCTIVITY 2000-2001

| | **Year 2000** | **Year 2001** |
|---|---|---|
| **Number of Cases** | 368 | 220 |
| | | |
| Business | 22 | 10 |
| Regression | 61 | 5 |
| Business : Regression | 2 | 2 |
| OO | 43 | 11 |
| Event | 24 | 7 |
| OO : Event | 13 | 2 |

## VII. CONCLUSION

This study utilized the latest release of the ISBSG data repository. The analysis shows that software development productivity has not improved over time. This is in parallel with the mounting intricacy of software development. We found the productivity experienced irregular variations between 1995 and 2005, and there is no trace of its ongoing improvement. In view of the factors affecting productivity, we found average team size and the uses of different language types have also varied in the past. We identify that the variations in the productivity are mainly caused by the changes of average team size and the irregular uses of different language types for the development.

## REFERENCES

[1] N. R. Howes, "Managing software development projects for maximum productivity," *IEEE Transactions on Software Engineering*, vol. SE10, 1984.
[2] Loesh, R.E. (1985), "Improving productivity through standard design templates," Data Processing, 27 (9), 57-59.
[3] J. D. Blackburn, G. D. Scudder, and L. N. V. Wassenhove, "Improving speed and productivity of software development: a global survey of software developers," *IEEE Transactions on Software Engineering*, vol. 22, pp. 875-885, 1996.
[4] G. R. Finnie, G. E. Wittig, and D. Petkov, "Prioritizing software development productivity factors using the analytic hierarchy process," *The Journal of Systems and Software*, vol. 22, pp. 129-139, 1993.
[5] Z. Jiang and C. Comstock, "The Factors Significant to Software Development Productivity," presented at 19th International Conference on Computer, Information, and Systems Science, and Engineering, Thailand, 2007.
[6] I. R. Chiang and V. S. Mookerjee, "Improving software team productivity," *Communications of the ACM*, vol. 47, pp. 89-93, 2004.
[7] D. Anselmo and H. Ledgard, "Measuring productivity in the software industry," *Communications of the ACM*, vol. 46, pp. 121-125, 2003.
[8] R. Groth, "Is the software industry's productivity declining?," *IEEE Software*, vol. 21, pp. 92-94, 2004.
[9] C. J. Lokan, "Statistical analysis of ISBSG data and FPA analysis," presented at Proceedings of 1999 Australian Conference on Software Metrics, Australia, 1999.
[10] R. Klepper and D. Bock, "Third and fourth generation language productivity differences," *Communications of the ACM*, vol. 38, pp. 69-79, 1995.

**Zhizhong Jiang** is a PhD student at Manchester Business School, University of Manchester, United Kingdom. He received his B.E.(first-class honors) from Harbin Institute of Technology (China) and MSc in Applied Statistics from University of Oxford.

**Peter Naudé** is a Professor of Marketing at Manchester Business School, University of Manchester, United Kingdom. He publishes widely in business studies and information systems.

**Craig Comstock** is a PhD student at the University of Oxford. He received his B.E. from Harvard University (Cum Laude), MSc in Software Engineering from University of Oxford,, and MBA (with highest honors) from University of Chicago.