Productivity vs. Quality of Software Development : An Empirical Study of the ISBSG Release 8

Chul-Mo Koo† · Dong-Jin Park† †

Abstract

Performance of software development is measured by two major criteria - roductivity and quality. Although the criteria is empirically tested in software engineering research, they often present with a limited way under consideration of a few factors or contexts for developers to focus on the either productivity facets or quality facets. Analyzing data on software development performance collected over a 13-year period from 20 countries, we investigated how major software development factors - development type, development platform, development technique, language type, DBMS, methodology, methodology acquisition, CASE,, summary of work effort, resource level, max team size, affect the performance of software development. The results suggest that productivity and quality of software development are affected by different factors and context: function points, line of code, extreme defects, major defects, or minor defects. This research provides the empirical evidence that the two performance criteria require for software developer to have careful attention to find the optimal balance between the two performance criteria.

Keywords : Software development, Productivity, Quality, ISBSG

1.Introduction

Over the past three decades, software development has become a critical concern for organizations [18] as IS has become an indispensable resource for them [26]. While the cost of hardware has continued to be decreased, the cost of developing software has steadily increased. Despite the increased costs, a large number of IS organizations struggle not only to develop and deliver software products on time and within budget but also to guarantee high quality of the softwares [6]. In fact, many studies found out that most of large software projects are not completed on time and within budget and even whey they are, they

† Assistant Professor, Chosun University
† † Professor, Kongju National University (Corresponding Author)
Received : 2010-01-10, Amended: 2010-02-05, Accepted : 2010-02-27

often do not function as intended [13]. Therefore, numerous studies have looked into the issue of software development performance. Some of these studies investigated such factors as personal skills, product size, process maturity, and usage of tools as potential drivers of quality and productivity [16][17][26]. However, most of these studies focused on either quality or productivity as the measure of performance [26] and rarely has any study investigated them extensively to understand the connections of primary software engineering factors between the two important performance criteria[10][19], but is important to understand their factors and their outcomes more various aspects such as, the variety of industries (e.g., banking, consumer goods, telecommunications, etc), the nature of the software development organization (e.g., in-house, outsourcing), the skills of software staff (e.g., broad, depth with a language, a method, a tool, an application domain), and the types of software products used (e.g., a design tool, CASE tools). Many researchers have tested the essential factors in various contexts so called a replicated study. However, if practitioners would like to know a specific situation of software development, then, they need to precede an experimental study to make sure what they want to achieve. However, a specific case is not easily applicable for well designed and contextual software engineering owing to the lack of standards.

Thus, we used a large dataset which included various factors and contexts that composes of software founding factors as well as situational information. ISBSG dataset has been collected with a report of software project results during the a long period time. This is a legitimate observational study provided by Kitchenham et al.[24]. Then, software development projects can be optimized using large samples. Thus, we set out to investigate the relationship between productivity and quality in conjunction with several important software development factors. A major contribution of this study is to examine extensively productivity and quality together to figure out how major factors and context may differently influence. We empirically test the dataset of ISBSG Release 8 collected with a report of observation.

2. Literature Review

2.1 Primary factors in software engineering

Several types of software development and various application products have been implemented in a variety of industries. Software engineering researchers have suggested a software development project that has to undergo a series of phases (SDLC: Software Development Life Cycle) which has a predetermined process [1] and identified well-known empirical factors and context since 1980s [2][34][35][36]. Kichenham et al.[24] importantly notified that there are two types of empirical software engineering research: observational studies and formal experiments techniques industrial context. Many researchers have clearly defined and identified important factors and context in software development studies in following <Table 1>.

〈Table 1〉 Software Development Factors and Context

Context	Factors	Researchers
A framework for MIS software development	• Technological sophistication; Interaction with end users; Systems specification known in advance; Project management; Language level utilized; Software productivity; Development cost; Development time; Number of end users; Importance of operational efficiency; Implementational problems	[31]
Programming perspectives	• Data administration -librarian -data dictionary/directory -online development aid • Shorthand Preprocessors -code generators -packaged Macros -user defined and packaged • Predefined logic -data retrieval/report/reporting/report generators -screen management -test data generator	[12] [38]
The software development environment	• Types of software developed • End user of software developed • Programming languages used • Total number of software projects undertaken in past two years • Size of software development staff • Total number of employee • Lines of code • Project team size • Number of job experience • Levels of effort and problems encountered in various phases of the software life cycle, etc. • The degree of use, and the effectiveness, of a wide variety of software engineering tools and techniques.	[2]
Quantitative analysis of software development	• Primary programming language by application • Hardware platforms • Personnel Experience • Effort by phase • Labor Inputs, • Effort-Time Effort Percentage by Please, Code Reuse, Capital Inputs, Software Engineering Tools	[8]
Observational studies-in situational studies of industrial practice	• Industry--where software products are being applied and used • The nature of the software development; In-house I nformation systems department; independent software supplier • The skills and experience of software staff--a software language, a method, a tool, an application domain • The software processes being used--a company-standard process, the quality assurance procedures, the configuration management process	[24]

2.2 Previous Study of ISBSG Dataset

We searched the use of the International Software Benchmarking Standards Group (ISBSG) Release 8 Edition repository for researching effort for software projects in previous literature. The previous study notes two questions: (1) There have not enough researched on the ISBSG dataset using the statistics method (2) There does not estimate the various software development factors and productivity and quality derived from the ISBSG dataset. We summarize the previous studies in our best knowledge in <Table 2>.

〈Table 2〉 The literature review of the ISBSG repository

Authors	Study	Journals
Jeffery et al. [22]	The use of the International Software Benchmarking Standards Group (ISBSG) repository for estimating effort for software projects in an organization not involved in ISBSG	Information and Software Technology
Lokan et al. [28]	Summary of ISBSG data	IEEE Software
Maxwell [30]	Benchmarking exercise's effectiveness using ISBSG	IEEE Software 2001
Kralj et al. [25]	Improved standard FDA(Function Point Analysis) method	Journal of Systems & Software

2.3 Software productivity and quality

Cusumano and Kemerera [8] assert that the importance of productivity and quality, and reuse of software code. Software development performance is directly closed to the line of code and its error rate (defects). Performance of software development is frequently measured and studied in two different dimensions - productivity and quality [1][4][9] [19][26]. Productivity - a quantitative aspect of performance - is assessed by various measures such as lines of code developed per programmer month [19] [23] and function points [1]. Function points are a popular productivity metric for software development. No matter what metric is used, productivity is defined as a ratio of the output to the amount of effort that was spent for the project [19]. On the other hand, software quality is "the degree to which software possesses a desired combination of attributes" [20]. The International Standard ISO-9126 Software Product Evaluation designates six software quality attributes such as functionality, reliability, efficiency, maintainability, and portability. Quality itself could be possibly understood by means of a variety of perspectives such as processes, products, services, mistakes, or errors [14][16]. Many researchers imply that software quality is closely related to the detection and correction of errors [11][27][33] and have focused on the perspective of software defects [5][7][15][29]. Humphrey indicates that detecting and correcting errors are the most common causes to delay software development procedure [18]. Table 3 summarizes previous studies on productivity and quality of software development.

〈Table 3〉 A Summary of software productivity & quality studi

Study	Variables used	Key findings
Banker and Kauffman [3]	Person days, new object percentage, application type, maturity processes	Effective software reuse is a precondition for project success
Subramanian and Zarnich [34]	Tool type, systems development method, tool experience	Rapid application development method and tool experience were associated with software productivity
Faraj and Sproull [13]	Presence of expertise, professional experience, administrative coordination, development methods, team performance	A strong relationship with team performance and input characteristics, presence of expertise, and administrative coordination
Krishnan et al.[26]	Personal capability, conformance quality, deployment of resources, development processes	Life-cycle productivity from the conformance quality and the quality product from personal capability, deployment of resources, and process
Harter et al.[16]	Process maturity, product size, complexity, ambiguity, quality, effort	The reduction in cycle time and effort due to improved quality
Harter and Slaughter [17]	Process maturity, product quality, CASE, workload, organizational inertia, infrastructure costs	Infrastructure costs come from quality improvement

3. Study Design

3.1 Hypotheses

Kitchenham et al. [24] provide the guideline of empirical research in software engineering and several factors may influence productivity and quality in software development project. We may provide a number of hypotheses and empirically analyze whether these hypotheses would be supported or rejected. The many of the hypotheses may be a replication study compared with the past study, however, we test these hypotheses using of a large volume of observational data from ISBSG including various context. The two questions addressed are here the relation of software development factors with related to productivity (function points, line of code) and quality (defect rate) and noted various contexts in software development We formulated hypotheses as the ISBSG data characteristics provided:

The ISBSG dataset provide factors: function points, summary of work effort, resource level, max team size, development type, development platform, language type, primary language type, DBMS used, upper CASE used, lower CASE (with code generator) used, low CASE (with no code generator), integrated CASE used, used methodology, how methodology acquired, development techniques, total defect delivered, extreme defects, major defects, minor defects, and also suggest contexts: organization type (industry), business area type, application type, and project scope. Therefore, we provide the following hypotheses:

- H1. Development type (new development, enhahahment, re-development) may differentlment, enhahahment, re-dev(function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)

- H2. Development platform (main frame, mid range, PC) may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H3. Development techniques (data modeling,
process modeling, etc.) may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H4. Language type (3G, 4G, ApG) may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H5. Primary language (COBOL, C++, C etc.) may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H6. DBMS used may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H7. Methodology used may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H8. How methodology acquired (developed in-house, purchased, combined in-house/purchased) may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H9. Upper CASE used may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H10. Lower CASE (with code generator) may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H11. Lower CASE (with no code generator) may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H12. Integrated CASE used may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H13. Summary of work effort may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H14. Resource level may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)
- H15. The size of a project team may differently influence productivity (function points, line of code) and quality (total defects delivered, extreme defects, major defects, minor defects)

4. Research Methodology

4.1 Samples

To test the hypotheses suggested, we analyzed data collected by ISBSG (International Software Benchmarking Standards Group, R8) for over a 13 year period between March, 1989 and December, 2002. The ISBSG database tracked over 1,500 software projects classified in four different types of software systems - (1) Management Information Systems (27%);

(2)Transaction/Production Systems (39%); (3) Office Information Systems (4%); and Others (2%). The data was collected from firms operating in a wide variety industries in twenty different countries. The data provides details of software development and business in <Table 4.>

〈Table 4〉 Software development of various contexts

Organization Type	Num(%)
• Communication	436(21.5)
• Financial, Property & Business Services	152(7.5)
• Insurance	144(7.1)
• Public Administration	144(7.1)
• Manufacturing	117(5.8)
• Banking	108(5.3)
• Electricity, Gas, Water	49(2.4)
• Wholesale & Retail Trade	37(1.8)
• Ordering	23(1.1)
• Billing	22(1.1)
• Others	270(13.3)
• Missing	525(25.9)
• Total	2027(100.0)

Project Scope	Num(%)
• Planning;Specification; Build ;Test	622(30.7)
• Specification;Build;Test	476(23.5)
• Build;Test	107(5.3)
• Design;Build;Test	51(2.5)
• Others	19(0.9)
• Missing	752(37.2)
• Total	2027(100.0)

Business Area Type	Num(%)
• Telecommunications	297(14.7)
• Banking	149(7.4)
• Insurance	149(7.4)
• Financial (excluding Banking)	109(5.4)
• Manufacturing	93(4.6)
• Accounting	54(2.7)
• Engineering	45(2.2)
• Inventory	27(1.3)
• Sales & arketing	27(1.3)
• Legal	25(1.2)
• Personnel;Education	23(1.1)
• Others	159(7.8)
• Missing	870(42.9)
• Total	2027(100.0)

Application Type	Num(%)
• Transaction/Production System	497(24.5)
• Management Information System	392(19.3)
• Financial transaction process/accounting;	65(3.2)
• Office Information System	62(3.1)
• Network Management	50(2.5)
• Business system	42(2.1)
• Electronic Data Interchange	35(1.7)
• Decision Support System	29(1.4)
• Executive Information System	22(1.1)
Ordering Billing	29(1.4)
• Process Control	21(1.0)
• Others	198(9.8)
• Missing	556(27.4)
• Total	2027

4.2 Data validity

We excluded some records as they deemed unreliable for the study. ISBSG rated quality and reliability of each data piece using letter grades - A, B, C and D, where A represents the highest level of integrity and D the lowest. We eliminated 192 records categorized as C or D and used the remaining 1,835 records for further analysis <Table 5>.

〈Table 5〉 Data quality rating test the hypotheses suggested, we analyzed data

Derived Count Approach	Num(%)	Data Quality Rating	Freq(%)	Cumulative Percent	Function Points	Value Adjustment Factor*
IFPUG	1827(90.1)	A	680(33.5)	33.5	Max(20,000)	1
NESMA	93(4.6)	B	1155(57.0)	90.5	Min(6)	1
Mark II	35(1.7)	C	122(6.0)	96.5	Mean(617.07)	1.02
CFFP	33(1.6)	D	70(3.5)	100.0	S.D(1391.2)	0.101
Others	39(1.92)	Total	2027(100.0)		Total Num (2011)	Total Num (1382)
Total	2027(100.0)					

*VAF: The adjustment to the function points, applied by the project submitter, that takes into account various technical and quality characteristics e.g.: data communications, end user efficiency etc. This data is not reported for some projects, (i.e. it equals 1)

We test data quality collected by the ISBSG which reported A, B, C, D with the results of software development project. ANOVA t-test of function points, line of code, total defect delivered, extreme defects, major defects, minor defects indicates significant mean differences between A, B, C, D. It is important not only to identify the qualified data group but also to measure them a significant difference. We ensure that the note of the ISBSG's clarification A,B,C,D reported by industrial practice results provides enough a legitimate data validity <Table 6>.

<Table 6> Data validity of A,B,C,D by ANOVA

Dependent variable		d.f	F	Sig.
• Function points	Between Groups Within Groups Total	3 2007 2010	15.704	.000
• Line of Code	Between Groups Within Groups Total	3 283 286	4.275	.006
• otal defects delivered	Between Groups Within Groups Total	3 462 465	3.658	.013
• Extreme defects	Between Groups Within Groups Total	3 250 253	.403	.751
• Major defects	Between Groups Within Groups Total	3 394 397	5.198	.002
• Minor defects	Between Groups Within Groups Total	3 322 325	2.632	.050

ISBSG provides many examples of software product, development process, and human resource characteristics that includes extensively across development type and technique with a language, tools et al in consistent with Kitchenham et al. [24]'s guideline. The ISBSG identifies several factors of software development but does not offer standardized measurement patterns. For example, resource level is measured with ordinal data categorized by 4 groups. Most of factors are written by qualitative measurement in the ISBSG dataset, therefore, we translated them as nominal data and analyze them as a frequency analysis in <Table 7>.

4.3 Operationalization

We excluded All variables of ISBSG used are reported directly by observing or measuring some aspects of software development in a particular software project. Thus, we omitted the procedures of data reliability analysis. In order to test our hypotheses, we determine if factors were statistically different for productivity and for quality. A one way analysis of variance was used to test for an overall difference between how factors are differed in productivity and quality, while a multiple analysis of regression test showed that there is affected in productivity and quality in <Table 8>.

<Table 7> Demographic of software development factors

Development type	Freq (%)	DBMS Use	Num (%)	CASE Used	
New Development	838(41.3)	Yes	1473(72.7)	Upper CASE	
Enhancement	1132(55.8)	No(Missing)	554(27.3)	Yes/ No	149(7.4)/774(38.2)
Re-development	55(2.7)			Lower CASE	
		Total	2027(100.0)	Yes/ No	101(5.0)/568(28.0)
Total	2028(100.0)			Lower CASE (No code gen)	
				Yes/ No	64(3.2)/538(26.5)
				Integrated	
				Yes/ No	48(2.4)/572(28.2)
Development	Freq (%)	Primary	Num (%)	Used Methodology	
Platform		Language		Yes	1277(63.0)
MF	844(41.6)	Type		No	212(10.5)
MR	252(12.4)			Missing	538(26.6)
PC	322(15.9)	Cobol	313(15.4)		
Missing	609((30.0)	C	147(7.3)		
Total	2027(100.0)	C++	114(5.6)		
Language Type	Num (%)	Oracle	98(4.8)	How Methodology Acquired	
2GL		Visual B	93(4.6)	Developed in House	818(40.4)
3GL	17(0.8)	SQL	91(4.5)	Combined in	133(6.6)
4GL	1024(50.5)	Natural	81(4.0)	House/Purchased	84(4.1)
5GL	504(24.9)	PL/I	78(3.8)	Purchased	992(48.9)
ApG	1(0.0)	Cobol II	72(3.6)	Missing	
Missing	90(4.5)	Cobol V2	45(2.2)		
	391(19.3)	Other 4G	39(1.9)		
		Access	37(1.8)		
		Telon	31(1.5)	Developed Technique	
		Powerbui	29(1.4)	Data Modeling	333(16.4)
		Java	26(1.3)	Business Area Modelling;	
		Cool.Gen	22(1.1)	Data Modelling/Event	106(5.2)
		Others	375(18.5)	Modelling/Prototyping	
		Missing	336(16.6)	Object Oriented Analysis;	45(2.2)
				JAD (Joint Application	
				Development)/Regression	
				/Testing	
				Standards: ISO 9000, SEI	
				CMM Level 2	42(2.1)
				JAD (Joint Application	
				Development);Regression	38(1.9)
				Testing	
				Process Modelling	
				RAD (Rapid Application	34(1.7)
				Development)	
				Prototyping	28(1.4)
				Others	27(1.3)
				Missing	117(5.8)
					1257(62.0)
Total	2027(100.0)	Total	2027(100.0)	Total	2027(100.0)

<Table 8> ANOVA results by softwbye development factors

	Development type	Development platform	Development Techniques	Language type	Primary Language	Used DBMS	Used Methodology	How Methodology acquired	Upper CASE	Lower CASE Generator	Lower CASE(no code gen)	Integ CASE
FP	14.52***	3.53**	2.02	.834	1.72	18.82***	16.74***	2.75*	8.12***	.46	.154	4.82**
LOC	6.88***	4.91***	1.88	1.88	2.72**	5.34**	.20	13.21***	7.33***	.08	.01	.02
TDD	.86	7.75***	.87	.295	1.60	3.94**	31.96***	.64	2.00	2.30	1.07	.06
ExtrD	1.59	6.64***	.56	1.39	3.66***	3.18*	27.74***	.26	2.92*	2.67	.89	.41
MarjD	.96	2.83*	3.09**	.13	.79	6.11**	36.22***	.78	.97	.03	1.48	.30
MinD	.75	5.87***	1.26	.30	1.08	1.42	5.28**	2.04	.57	.85	.47	2.38

FP : Function Points, LOC : Line of Code, TDD: Total Defects Delivered, ExtrD: Extreme Defects, MarjD: Major Defects, MinD: Minor Defects

<Table 9> presents the independent variables summary of work effort, resource level, max team size, and dependent variables : productivity and quality. Summary of work effort presents the total effort in hours recorded against the project by the development organization [16]. The resource level [el] is classified into four categories as follows : (1) Level 1 involves only project development team including project staff, project management, project administration, (2) Level 2 include development team support (e.g., database administration, data administration, quality assurance, data security, standards support, audit & control, technical support) in addition to the project development team, (3) Level 3 adds computer operations involvement (e.g., software support, hardware support, information center support, computer operators, network administration), and (4) Level 4 incorporate end users or clients (e.g., user liaisons, user training time, application users and/ or clients) on top of all other personnel involved. In case of team size is measured ratio data. The maximum number of people that worked at any time on the project (peak team size) provides team size [32][37][39].

$Productivity(FP,\ LOC) = a + \beta 1 * Summary\ Work\ Effort + \beta 3 * Resource\ Level + \beta 3 * Max\ Team\ Size + \varepsilon$

$Quality\ (TDD,\ ExtrD,\ MarjD,\ MinD) = a + \beta 1 * Summary\ Work\ Effort + \beta 3 * Resource\ Level + \beta 3 * Max\ Team\ Size + \varepsilon$

<Table 9> Multiple regression by software development factor

Dependent variables	R, R2, Adjusted R2 (%)	S.W.E β (t-value)	Resource Level β (t-value)	M.T.S β (t-value)
FP	64.2, 41.2, 41	.668(24.62)***	-.075(-3.04)***	-.038(-1.42)
LOC	52, 27, 25	.593(6.20)***	-.011(-.13)	-.220(-2.33)**
TDD	26.6, 7.1, 6.1	.278(4.32)***	-.137(-2.38)**	-.065(-1.02)
ExtrD	10.1, 1, -1	-.011(-.11)	.081(.95)	-.078(-.79)
MarjD	26.4, 7, 5.9	.233(3.46)***	-.198(-3.33)***	-..054(-.818)
MinD	37.9, 14.4, 12.8	.498(4.97)***	-.014(-.185)	-.220(-2.24)**

4.4 Analysis and results

We specify the relationship between software development factors and dependent variables. ANOVA and multiple regression statistics employed in this study are shown (See, Table 8, 9).

The results indicate that productivity (function points and line of code) is significantly differed depending on development type (new development, enhancement, re-development), development platform (main frame, mid range, PC). Line of code is significantly differed relying on what primary languages are. Whether DBMS used or not also influences productivity, however, used methodology only influences function points. Depending on the acquisition of methodology (developed in-house, purchased, combined) affect productivity as well. Only upper CASE tool statistically influences productivity, but ICASE only is a significant determinant to function points. Summary of work effort significantly influence function points and line of code. The more people widely and vertically involved in software project (resource level), more function points produces, instead, the larger team size, the less line of code produces.

In terms of software quality, total defects delivered, extreme defects, major defects, and minor defects are significantly different whether development platforms of main frame, mid range, or PC are. Development technique (Data modeling, process modeling, etc.,) is only related to major defects. Extreme defects arise from primary language. Whether DBMS used or not may affect extreme defects and major defects. Used of methodology is statistically significant toward defects factors. Finally, upper CASE tool is related to extreme defects. In the regression analysis, summary of work effort influence total defect delivered, major and minor defects. Resource level is shown a negative effect (reverse code) toward total defects delivered and major defects. In case of resources level 4includes end users and client, therefore, defects rate may less appear.

5. Discussion and Conclusions

The goal of this study is to uncover the performance of software development measured by two major criteria - productivity and quality. Analyzing data on software development performance collected over a 13-year period from 20 countries, we investigated how major factors of software development are significantly different toward productivity and quality : (1) development type, (2) development platform, (3) development technique, (4) language type, (5) primary language, (6) used DBMS, (7) used methodology, (8) methodology acquisition, (9) upper CASE, (10) lower CASE (with code gen), (11) lower CASE (no code gen), (12) integrated CASE, and (13) summary of work effort, (14) resource level, (15) max team size affect toward productivity and quality. The results of this study provide that,

- first, development type as a context variable is an very important factor toward productivity both function points and line of code, however is not related to quality aspect.
- second, development platform is significantly critical both productivity as well as quality.
- third, DBMS and methodology adopted is important factors not only productivity but also quality
- fourth, the way of methodology acquisition is not statistically related to quality.
- fifth, upper CASE out of other CASE tools is important for productivity.
- finally, the more involvement of user resource level, the better qualitative performance, but conversely result in a negative productive performance.

Our findings suggest that major factors and a context variable influence differently regarding the software performance of productivity and quality. We find out that the effects of the independent variables result in software productivity depending on function points or line of code versus quality including extreme, major, and minor defects in software project. First, we confirms previous researches by way of replicable study, however, we measured ter, wlationship between extensive independent factors and dependent factors. Secondly, development platform, used DBMS, and used methodology out of all factors suggest that tery have *madesearcheion significantly to* ter,productivity versus ter,udy, howeveat is, pxtctieioner should take carefully consi-deraeion that tery do enable project members to detect errors and gdyrantep bee productivity and quality in software development, if tery apply those methods and tools adequately. Finally, our finding of resource levelonre developmenviouoftware suppware computer operations, and end users) decreases the error and dercheion in software developmentweveis result suggests that the important of cooperation ioong other groups and end user involvement, which means that managers who are responsible a project need to communicate between team members, coordinate their roles, and encourage them to cooperate. Such a team supporting for a project make less errors and ultimately deliver the software product on time.

Our study provides insight into the causes and effects of productivity versus quality for software development. This research provides the empirical evidence that the two performance criteria require for software developer to have careful attention to find the optimal balance between the two performance criteria. Although this research improves our understanding of the causes and effects of productivity versus quality for software development, we took sample out of the ISBSG Release 8 dataset, which means, we need to carefully interpret the findings depending on considering the limitation. Future studies in productivity and quality may include other context situations.

References

[1] Ahituv, N and Neumann, S.(1984), "A Flexible Approach to Information System Development," MIS Quarterly, pp.69-78.

[2] Banker, R.D., Davis, G. B., and Slaughter, S. A.(1998), "Software Development Practices, Software Complexity, and Software Maintenance Performance: A Field Study," Management Science, 44(4), April, pp.433-450.

[3] Banker, R. D. and Kauffman, R. J.(1991), "Reuse and Productivity in Integrated Computer-Aided Software Engineering: An Empirical Study," MIS Quarterly, September, pp.375-401.

[4] Banker, R. D. and Kemerer, C. F.(1989) "Scale Economies in New Software Development," IEEE Transactions on Software Engineering, 15(10), October, pp. 1199-1205.

[5] Basli, V.R., and Perricone, B. T.(1983), "Software Errors and Complexity: An Empricial Investigation,"Communications of the ACM. 27, pp.42-51.

[6] Blackburn, J.D., Scudder, D.G., and Wassenhove, L. N.(1996) "Improving Speed and Productivity of Software Development. Working Paper, INSEAD, Fontainebleau, France.

[7] Crosby, P. B.(1979), Quality is Free. McGraw-Hill, New York.

[8] Cusumano, M. A. and Kemerer, C. F.(1990), "A Quantitative Analysis of U.S. and Japanese Practice and Performance in Software Development," Management Science, 36(11), pp.1384-1406.

[9] Davis, G. B.(1988), "Commentary on Information Systems : Productivity

Gains from ComputerAided Software Engineering," 2(2), Accounting Horizons, June pp.90-93.

[10] Dromey, R. G.(1995), "A Model for Software Product Quality," IEEE Transactions on Software Enginneering, 21(2), February pp.146-162.

[11] Dyer, M., and Kouchakdjian, A.(1990), "Correctness Verification: Alternative to StructuralSoftware Testing," Information Software Technology, 32(1), pp.53-59.

[12] Ewers, J. and Vessey, I.,(1981), "The Systems Development Dilemma - A Programming Perspective," MIS Quarterly, 5(2), pp. 33-45.

[13] Faraj, S., and Sproull, L.(2000), "Coordinating Expertise in Software Development Teams," Management Science 46(12), December pp.1554-1568.

[14] Fox, C. J., and. Frakes, W. B.(1997), "The Quality Approach : Is It Delivering?" Introduction to a special section of Communications of the ACM 40 (6), June 24-29.

[15] Gyrna, F.(1988), Quality Costs. Quality Control Handbook, 4th ed. McGraw-Hill, New York.

[16] Harter, D.E., Krishnan, M.S., and Slaughter, S.A.(2000), "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development," Management Science, 46(4), April pp.451-466.

[17] Harter, D. E., and Slaughter, S. A.(2003), "Quality Improvement and Infrastructure Activity Costs in Software Development: A Longitudinal Analysis," Management Science 49(6), June pp.784-800.

[18] Humphrey, W.S(1995), A Discipline for Software Engineering. Addison-Wesley, Reading, MA.

[19] Humphrey, W.S. and Singpurwalla, N.D.(1991), "Predicting (Individual) Software Productivity," IEEE Transactions on Software Engineering, 17(2), February pp.196-207.

[20] IEEE, Inc. IEEE(1993), Standard for a Software Quality Metrics Methodology. IEEE Computer Society. New York:

[21] Ives, B. and Olson, M. H.(1984), "User Involvement and MIS Success: A Review of Research," Management Science, 30(5), pp.586-603.

[22] Jeffery, R., Ruhe, M. and Wieczorek, I.,(2000), "A Comparative Study of Two Software Development Cost Modeling Techniques Using Multi-Organizational and Company -Specific Data," Information and Software Technology, 42(14), pp.1009-1016.

[23] Jones, T.C.(1986) Programming Productivity, McGraw-Hill, New York, NY

[24] Kitchenham, B. A., Pfleeger, S. L, Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K. and Rosenberg, J.(2002), "Preliminary Guidelines for Empirical Research in Software Engineering," IEEE Transactions on Software Engineering, 28(8), pp.721-734.

[25] Kralj, T., Rozman, I., Hericko, M. and Zivkovic, A.(2005), "Improved standard FPA method resolving problems with upper boundaries in the rating complexity process," Journal of Systems & Software 77(2), pp. 81-90.

[26] Krishnan, M. S., Kriebel, C. H., Kekre, Sunder., and Mukhopadhyay, T.(2000), "An Empirical Analysis of Productivity and Quality in Software Products," Management Science 46(6), June pp.745-759.

[27] Laitenberger, O., and DeBaud, J. "An Encompassing Life Cycle Centric Survey of Software Inspection," Journal of Systems Software, 50(1), pp.5-31.

[28] Lokan, C., Wright, T., Hill, P. R. and Stringer, M.(2001), "Organizational Benchmarking Using the ISBSG Data Repository," IEEE Software, 18(5), pp.26-32.

[29] Lyu, M.R.(1996), Handbook of Software Reliability Engineering. McGraw-Hill, New York
[30] Maxwell, K D.,(2001), "Collecting Data for Comparability: Benchmarking Software Development Productivity," IEEE Software, 18(5), 22-25.
[31] Moore, J.(1979), "A Framework for MIS Software Development Projects," MIS Quarterly, 3(1), pp. 29-38.
[32] Putnam, D. "Team Size Can Be the Key to a Successful Project," http://www.qsm.com/process_01.html
[33] Slaughter, S. A., Harter, D. E. Krishnan, M.S.(1998), "Evaluating the Cost of Software Quality," Communication of ACM, 41(8), pp.67-73.
[34] Subramanian, G and Zarnich, G. E.(1996), "An Examination of Some Software Development Effort and Productivity Determinants in ICASE Tool Projects," Journal of Management Information Systems, 12(4), Spring pp.143-160.
[35] Symons, C.(1998), " Function Point Analysis: Difficulties and Improvements," IEEE Transactions on Software Engineering, 14(1), pp. 2-11.
[36] Thayer, R. H,Pyster, A. B.and Wood, R. C.(1981), " Major Issues in Software Engineering Project Management," IEEE Transactions on Software Engineering, 7(4),pp.333-342
[37] Walz, D.B., Elam, J.J. Curtis, B.(1993), "Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration," Communications of ACM. 36,pp.63-77.
[38] Warren, H. and Bahram, A.(86/87) "The Role of Programming Language in Estimating Software Development Costs," Journal of Management Information Systems, Winter 3(3), pp. 101-110.
[39] Zultner, R.E.(1993), "TQM for Technical Teams," Communications of the ACM, 36(10), October pp.78-91.

Chul-Mo Koo

1997 Ajou Univ.
(B.S. in English Literature)
1999 Ajou Univ. (M.S. in MIS)
2003 Sogang Univ.
(Ph.D. in MIS)
2004-5 University of Minnesota MISRC Post Doc.
2004-2006 Marshall University, Assistant Prof.
2007.1-12 Inha Univ. BK Research Prof.
2008.1-2 Yonsei Univ. BK Research Associate
2008.3-Present Assistant Professor B at School of Business, Chosun Univ.
Research Papers: International Journal of Electronic Commerce, International Journal of Information Management, Industrial Management & Data Systems, Information Systems Frontiers, Journal of Internet Commerce, APJIS
Research Areas: Green IT & IS, Social Network Communications

Dong-Jin Park

1983 Ajou Univ.
(B.S. in Industrial Engineering)
1988 Hankuk Univ. of Foreign Studies (M.S. in MIS)
1994 Ajou Univ. (Ph.D. in MIS)
1998-Present Professor of Industrial and Systems Engineering Dept., Kongju National Univ.
1988-1990 Senior Researcher, Korea Productivity Center
1995-1998 Assistant Professor, Namseoul Univ.
Research Areas: Metadata, Scientific Data Management, Production Information System, ERP, MES

ISBSG 8을 이용한 소프트웨어 개발의 생산성과 품질에 관한 실험적 연구

구철모 · 박동진

요 약

본 연구는 소프트웨어 개발과 성과에 대한 연구를 조사하였다. 소프트웨어의 생산성과 품질성과 크게 두 측면으로 조사되었다. 소프트웨어에 대한 기존 연구를 통하여 많은 연구들이 진행되어 왔지만 생산성과 품질에 대한 전체적인 영향요인을 포괄적으로 다룬 연구는 매우 적은편이다. 본 연구는 20개국에서 13년 동안 수행된 프로젝트의 성과를 대상으로 영향요인을 조사하였다. 영향요인은 다음과 같다. 소프트웨어 개발 형태, 개발 플랫폼, 개발 기술, 개발 언어, DBMS, 방법론, 방법론 획득 방식, CASE 툴, 총 개발투입 시간, 인적자원 참여 수준, 최대 팀 사이즈가 조사되었다. 본 연구결과 펑션 포인트, 코드 라인 수, 소프트웨어 결함(치명적, 중요, 작은)각 품질과 생산성을 대표하는 변수로 측정되었고 제시된 영향요인과 관련이 있음이 조사되었다. 따라서 소프트웨어 개발에 있어 성과와 품질에 영향을 미치는 요인들이 각각 틀리며 이와 같은 이유 때문에 개발에 참여한 소프트웨어 개발자들은 두 가지 측면에 요인을 균형있게 고려해야 할 것이다.

키워드 : 소프트웨어 개발, 소프트웨어 생산성, 소프트웨어 품질, ISBSG