Quantitative Project Assessment for Quality-Oriented Software Management

Shigeru Yamada [#]

[#]Department of Social Management Engineering,
Graduate School of Engineering, Tottori University
4-101 Minami, Koyama, Tottori-shi, 680-8552 Japan

ABSTRACT
Many risks due to human factors are latent in a software development project. If suitable management can be performed to these risks and a software development process can be improved continuously, we will lead the project to improve in the productivity and quality of software product. In this paper, we analyze the process data collected from actual software development projects, and clarify the process factors which affect the quality of software product, by using multivariate analysis. Further, we also discuss a method of quantitative project evaluation based on a software reliability growth model, which helps us to give useful quantitative measures for determining project completion.

Keywords
Principal Component Analysis, Multiple Regression Analysis, Structural Equation Modeling, Logistic Regression Analysis, Software Reliability Growth Model, Geometric Poisson Model.

1. INTRODUCTION
In recent years, our society is called a highly informative society or information network society, and computer systems are playing the important role in all fields. Thus, if the dependence on computer systems increases, large-scaling, complication, and diversification of the software will be advancing increasingly. Furthermore, the tendency of high quality software development and quick-delivery by the advancement of user requirements is also becoming strong. In order to develop a software product which satisfies a customer's quality requirement within a development period, it is important to carry out the management of an effective software development project. Moreover, if the product quality is high, backtracking work will decrease and lead to shorter delivery time and cost reduction. Therefore, *Quality-Oriented Software Management*[1],[2] eventually makes the productivity of software projects improve.

Generally, a software product is produced according to the development process consisting of serial processes of requirement specification, design, coding, and testing. Many risks which affect product quality are latent in a software development project. Therefore, suitable management is performed to these risks and it is thought that improving the software development process continuously leads to improved product quality.

In this paper, we analyze the process measurement data obtained from actual software development projects with multivariate analysis, and clarify the process factors which affects the product quality (see Fig. 1). Furthermore, based on a software reliability growth model, we discuss a method of quantitative project evaluation, which helps us to give useful quantitative measures for determining project completion.

2. ANALYSIS OF THE PROCESS FACTORS AFFECTING PRODUCT QUALITY

2.1 DATA USED FOR ANALYSIS
Using the actual process data obtained from 27 software development project (as shown in Table 1), the process factors which affect software product quality are analyzed. In addition, the number of faults detected in the system testing, Y_q, is used as a objective variable for the product quality.

2.2 CORRELATION ANALYSIS
Using the process data as shown in Table 1, we perform a correlation analysis to clarify the relations among variables. From the correlation analysis, we can consider the correlations as follows:

- Y_q ***has strong correlation to*** X_{s1} ***and*** X_h.
- X_{s1} ***has strong correlation to*** X_h, X_{cdr}, ***and*** X_{ct}.
- X_{s2} ***has strong correlation to*** X_{bdr}, X_{cdr}, ***and*** X_{cdoc}.
- X_{bdoc} ***has strong correlation to*** X_{bdr}.
- X_{cdoc} ***has strong correlation to*** X_{cdr}.

Table 1: Process data in actual software development projects

Project No.	Size			Man day	Risk		PJ		System design		S/W design		Coding		Integrated testing	System testing	
	New development (kstep)	Reconstruction development (kstep)	Reuse development (kstep)	Planned man-day (Hr)	Embedded system(0,0) Enterprise system(0,1) Embedded system(V)(1,0)		leader experience (Years)	Outsourcing ratio (%)	Document pages	Number of detected problems	Document pages	Number of detected problems	Number of detected problems	Number of problems detected in static testing	Number of test-items	Number of test-items	Number of detected faults (product quality)
	$Xs1$	$Xs2$	$Xs3$	Xh	$Xbu1$	$Xbu2$	$Xexp$	$Xoff$	$Xbdoc$	$Xbdr$	$Xcdoc$	$Xcdr$	Xcr	$Xqac$	Xct	Xgt	Yq
1	88.700	12.667	73.980	15548	0	0	24	70	73	20	199	34	16	380	21366	12587	316
2	63.100	0.000	0.000	18187	0	0	25	74	0	0	298	87	285	0	0	1390	167
3	106.500	311.333	7.100	13724	0	1	16	68	1356	263	10003	249	35	5	19811	4970	53
4	0.100	14.300	62.300	6081	0	0	22	83	0	0	55	3	2	0	321	6163	27
5	4.000	1.667	77.440	3459	0	0	22	83	0	0	4	0	2	0	11	4850	6
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
22	0.000	0.760	3.000	791	1	0	24	28	50	1	417	6	0	1	751	1212	8
23	0.000	0.367	20.600	543	1	0	20	50	0	0	15	3	0	0	884	2414	0
24	5.500	1.000	37.000	682	1	0	20	67	25	0	78	12	4	0	85	283	1
25	0.000	0.333	2.600	884	1	0	24	38	46	1	589	4	0	0	1847	873	10
26	0.000	0.067	23.600	783	1	0	20	57	24	2	20	7	0	0	926	780	3
27	0.600	0.067	34.800	682	1	0	24	80	62	10	19	9	4	3	71	35	2

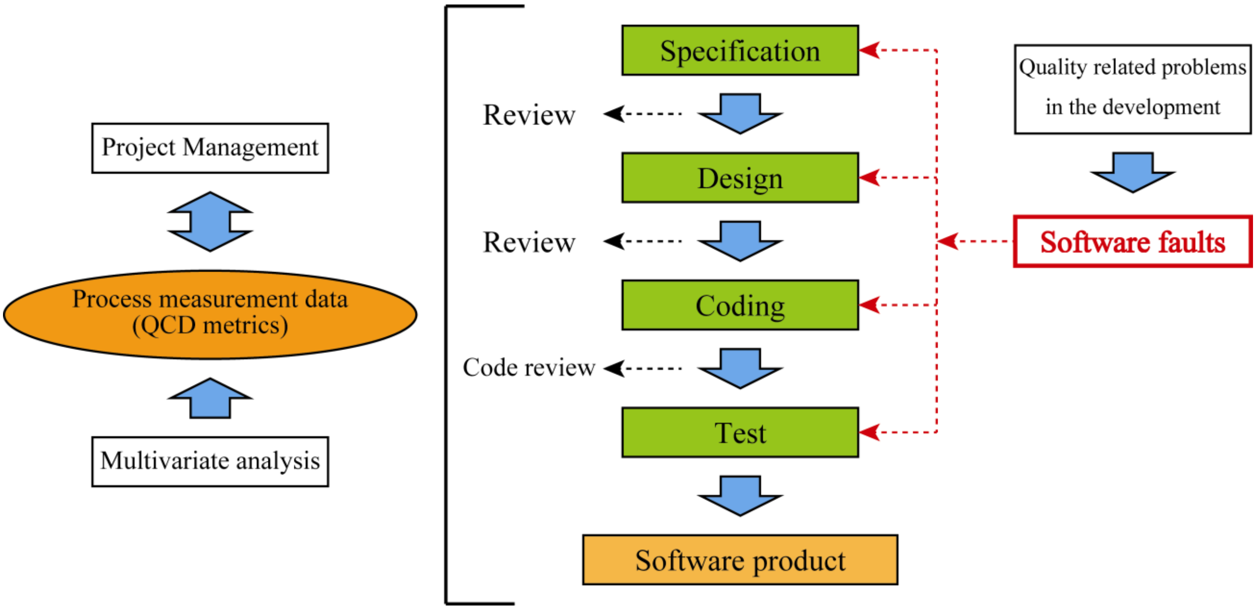

Fig. 1: Quality of software products is realized in the development process

2.3 MULTIPLE REGRESSION ANALYSIS

In consideration of a possibility that there is multicollinearity, based on the possible regression and the correlation analyses, X_{s2}, X_h, X_{bu1}, X_{bu2}, X_{bdr}, X_{cdr}, X_{cr}, and X_{ct} are selected as important factors for predicting the number of faults detected in the system testing, Y_q.

A multiple regression analysis is applied to the process data of the 27 projects as shown in Table 1. Then, using X_{s2}, X_h, X_{bu1}, X_{bu2}, X_{bdr}, X_{cdr}, X_{cr}, and X_{ct}, we obtain the estimated multiple regression equation for predicting the number of faults, $\hat{Y}_q$, given by Eq. (1) as well as the normalized multiple regression expression for Eq. (1), $\hat{Y}_q^{\,N}$, given by Eq. (2):

$$\hat{Y}_q = -0.525 \times X_{s2} + 0.003 \times X_h - 9.606 \times X_{bu1} - 9.689 \times X_{bu2} + 0.261 \times X_{bdr} - 0.642 \times X_{cdr} + 0.594 \times X_{cr} + 0.013 \times X_{ct} + 7.227, \quad (1)$$

$$\hat{Y}_q = -0.503 \times X_{s2} + 0.199 \times X_h - 0.070 \times X_{bu1} - 0.047 \times X_{bu2} + 0.241 \times X_{bdr} - 0.476 \times X_{cdr} + 0.491 \times X_{cr} + 1.068 \times X_{ct}. \quad (2)$$

In order to check the goodness-of-fit adequacy of our model, the coefficient of multiple determination (R^2) is calculated as 0.986. Furthermore, the squared multiple correlation coefficient, called the contribution ratio, adjusted for degrees of freedom (adjusted R'^2) is given by 0.980, and Eq. (2) is shown to be significant at 1% level. The result of multiple regression analysis is summarized in Tables 2 and 3.

Table 2: Analysis of variance

Source of variation	DF	SSq	MSq	F-value
Due to regression	8	110668.499	13833.562	162.510**
Error	18	1532.241	85.125	
Total	26	112200.741		

Table 3: Estimated parameters

Factor	Coefficient	SE	t-value	Standard coefficient
Intercept	7.227	4.003	1.805	
X_{s2}	-0.525	0.086	-6.109	-0.503
X_h	0.003	0.001	2.101	0.199
X_{bu1}	-9.606	4.975	-1.931	-0.070
X_{bu2}	-9.689	8.746	-1.108	-0.047
X_{bdr}	0.261	0.062	4.178	0.241
X_{cdr}	-0.642	0.112	-5.752	-0.476
X_{cr}	0.594	0.080	7.467	0.491
X_{ct}	0.013	0.001	14.982	1.069

From Table 2, it is found that the accuracy of these multiple regression equations is high. Then, we can predict the number of faults by using Eq. (2). From Eq. (2), the order of the degree affecting the objective variable Y_q is $X_{ct} > X_{s2} > X_{cr} > X_{cdr} > X_{bdr} > X_h > X_{bu1} > X_{bu2}$.

3. INTENSIVE PROCESS FACTORS

3.1 PRINCIPAL COMPONENT ANALYSIS

In order to clarify the relationships among variables and classify the process characteristics, principal component analysis is carried out. It is found that the precision of analysis is high from Table 4. And the factor loading values are obtained as shown in Table 5. From Table 5, let us newly define the first and second principal components as follows:

- *The first principal component is defined as the measure for the quality related factors.*

- *The second principal component is defined as the measure for the review related factors.*

Table 4: Characteristic values

Component	Eigenvalue	Contribution ratio	cumulative contribution ratio
1	4.22	0.469	0.469
2	2.06	0.229	0.698

Table 5: Factor loading values

	Component1	Component2
X_{s2}	0.828	0.506
X_h	0.780	-0.589
X_{bu1}	-0.351	0.273
X_{bu2}	0.541	0.489
X_{bdr}	0.734	0.466
X_{cdr}	0.908	0.174
X_{cr}	0.383	-0.641
X_{ct}	0.831	-0.007
Y_q	0.560	-0.696

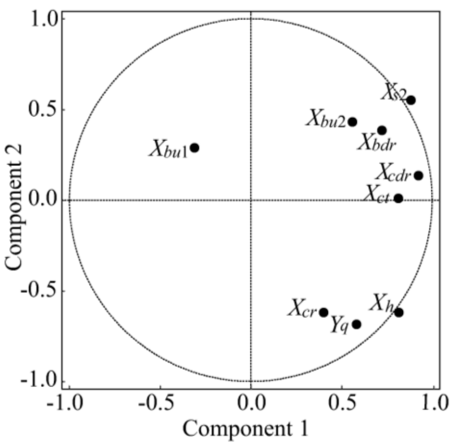

Fig. 2: Scatter plot of the factor loading values

Fig. 2 is a scatter plot of the factor loading values in Table 5. From Fig. 2, we can consider the correlation as follows:

- *A planned man-day (X_h) has positive correlation to the number of faults detected in system testing.*

- *The number of problems detected in source code review (X_{cr}) has positive correlation to the number of faults detected in system testing.*

The principal component scores are obtained as shown in Table 6. From Table 6, since project No.1 and No.3 have the large value in the first principal component, the product quality is considered to be bad. Moreover, although a reconstruction development size is small as for Project No.2, since a planned man-day (X_h) is large and the number of problems detected in source code review (X_{cr}) is also large, the product quality is considered to be bad.

Table 6: Principal component scores

No.	Component1	Component2
1	1.7254	-2.3580
2	1.3358	-3.3348
3	4.0723	2.1771
4	-0.1941	-0.3685
5	-0.4002	-0.1606
6	-0.3701	-0.2291
7	-0.1293	-0.4411
8	-0.2628	-0.3345
9	-0.0544	-0.4812
10	0.0359	-0.4470
11	0.0775	0.7138
12	-0.0550	0.3022
13	-0.4775	0.0597
14	-0.5367	0.0625
15	-0.4033	0.0270
16	-0.0212	0.2524
17	-0.3190	-0.0297
18	0.3579	1.1845
19	-0.4739	0.3747
20	-0.6983	0.3386
21	0.5239	0.8540
22	-0.6207	0.2845
23	-0.6587	0.3299
24	-0.6321	0.3112
25	-0.5835	0.2643
26	-0.6196	0.3136
27	-0.6182	0.3344

3.2 APPLICATION OF STRUCTURAL EQUATION MODELING

In order to clarify the causal relationships among the process factors which have affected product quality, the path-diagram is created and the analysis by structural equation modeling is conducted.

Fig.3 shows the result of analyzing the created path-diagram by structural equation modeling. The provided chi-square test and fix index are shown in Tables 7 and 8, respectively. From Tables 7 and 8, p-value is 0.990, and GFI 0.979 (AGFI is 0.904). Then, it can be said that the goodness of fit to the data of structural equation modeling is high. Therefore, we confirm that the process factors that affect the product quality, (X_{s2}, X_h, X_{bu1}, X_{bu2}, X_{bdr}, X_{cdr}, X_{cr}, X_{ct}), are important in managing the product quality.

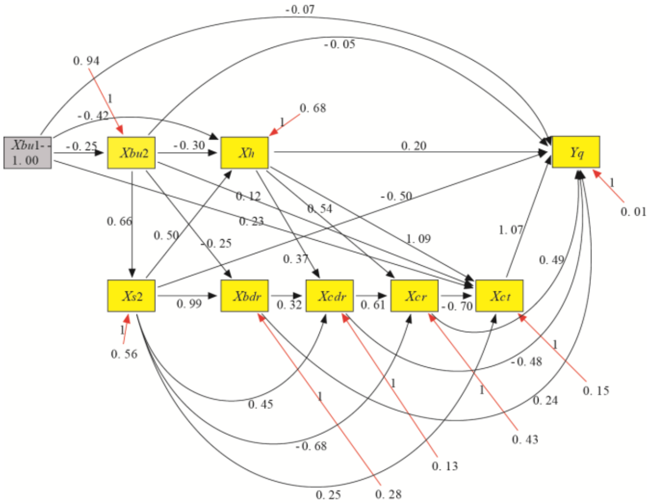

Fig. 3: The path-diagram by structural equation modeling

Table 7: Chi-square test

	Test statistic	degree of freedom	p-value
INDEPENDENCE MODEL CHI-SQUARE	297.560	36	
MODEL CHI-SQUARE	2.582	10	0.990
MINIMIZED MODEL FUNCTION VALUE	0.099		

Table 8: Goodness-of-fit index

	Abbreviated name	Abbreviated name
BENTLER-BONETT NORMED FIT INDEX	NFI	0.991
COMPARATIVE FIT INDEX	CFI	1.000
LISREL AGFI FIT INDEX	AGFI	0.904
LISREL GFI FIT INDEX	GFI	0.979
ROOT MEAN-SQUARE RESIDUAL	RMR	0.033
STANDARDIZED RMR	SRMR	0.033
ROOT MEAN-SQUARE ERROR OF APPROXIMATION	RMSEA	0.000

4. DISCRIMINANT ANALYSIS BY LOGISTIC REGRESSION EQUATION

Using the process data as shown in Table 1, we apply the logistic regression analysis and derive a regression equation to determine whether faults will be detected in the system testing or not. The derived regression equation is useful for judging whether or not the final product quality is good, i.e., the project is successful. The objective variable for logistic regression analysis, z, is defined as follows:

- z=1: The faults in the system testing will be detected.
- z=0: The faults in the system testing will not be detected.

Then, from Table 9, the goodness-of-fit of the derived logistic regression equation is well.

Table 9: Goodness-of-fit test by Hosmer-Lemeshow

Chi-square statistic	1.511
Flexibility	3
p-value	0.680

Grouping by the predictive probability	~ 0.55	~ 0.93	~ 0.93	~ 0.97	~ 0.99
The number of the applicable samples	4	4	4	5	5
Observation level (Mean)	0.250	1.000	1.000	1.000	1.000
Fitted value (Mean)	0.148	0.881	0.931	0.954	0.981
Chi level	0.578	0.734	0.544	0.493	0.314

From Table 10, we obtain the estimated logistic regression equation given by Eq. (3) for determining whether or not faults will be detected in the system testing.

Table 10: Estimated parameters

Factor	Coefficient	SE	p-value (the upper side)	Tolerance
Intercept	-6.309	1.963	0.001	
X_{exp}	0.405	0.094	0.000	0.901
X_{qac}	0.004	0.002	0.040	0.901

$$z \equiv \ln \frac{p(x)}{1-p(x)} = 0.405 \times X_{exp} + 0.004 \times X_{qac} - 6.309, \qquad (3)$$

$p(x)$: The probability of faults detection by using X_{exp} and X_{qac} .

From Table 11, the distinction hit ratio is 100%. Therefore, the derived regression equation in Eq. (3) can judge whether or not the software project has a quality process with high accuracy.

Table 11: Judgement table

	Predicted $z=0$	Predicted $z=1$	Number of projects
Observed z is 0	3	0	3
Observed z is 1	0	19	19
Number of projects	3	19	22

5. QUANTITATIVE PROJECT EVALUATION

We discuss quantitative project assessment based on the process data as shown in Table 1. A project progress growth curve is assumed to be the relationship between the number of process progress phases and the cumulative number of problems detected during the process progress. Then, we apply Moranda geometric Poisson model[3],[4], which is a software reliability growth model(SRGM), to the process data on X_{bdr}, X_{cdr}, X_{cr} and X_{qac} as shown in Table 1.

We discuss project progress modeling based on the Moranda geometric Poisson model because an analytic treatment of it is relatively easy. Then, we choose the number of process progress phases as the alternative unit of testing-time by assuming that the observed data for testing-time are discrete in an SRGM as shown in Fig.4.

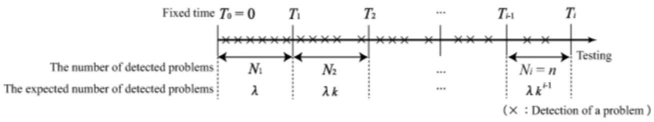

Fig. 4: Moranda geometric Poisson model

In order to describe a problem-detection phenomenon during progress phase i ($i = 1,2,\cdots$), let N_i denote a random variable representing the number of problems detected during i^{th} project progress interval (T_{i-1}, T_i]($T_0 = 0$; $i = 1,2,\cdots$). Then, the problem-detection phenomenon can be described as follows:

$$\Pr\{N_i = n\} = \frac{\{\lambda k^{i-1}\}^n}{n!} \times \exp[-\lambda k^{i-1}] \qquad (4)$$
$$(\lambda > 0, 0 < k < 1; n = 0,1,2,\cdots),$$

where $\Pr\{A\}$ means the probability of event A, and

$\lambda =$ the average number of problems detected in the first interval($0, T_1$],

$k =$ the decrease ratio of the number of problems detected by review activities.

From Eq.(4), setting $T_i = i(i = 1,2,\cdots)$, we obtain the following quantitative project evaluation measures, that is, the expected cumulative number of problems detected up to n^{th} process progress phase, $E(n)$, and the expected total number of problems latent in the software project, $E(\infty)$, are given as Eqs.(5) and (6), respectively:

$$E(n) = \sum_{i=1}^{n} \lambda k^{i-1} = \frac{\lambda(1-k^n)}{(1-k)}, \qquad (5)$$

$$E(\infty) = \lim_{n\to\infty} E(n) = \frac{\lambda}{1-k}. \qquad (6)$$

Project evaluation measures play an important role in quantitative evaluation of the process progress. The expected number of remaining problems, $r(n)$, represents the number of problems latent in the software project at the end of n^{th} process progress phase, and is formulated as

$$r(n) \equiv E(\infty) - E(n), \qquad (7)$$

and the instantaneous MTBP which means the mean time between problem-detections is formulated as

$$MTBP(n) = \frac{1}{\lambda k^{n-1}}. \qquad (8)$$

We present numerical examples by using the Moranda geometric Poisson model for Project No.3. Fig.5 shows the estimated cumulative number of detected problems, $E(n)$, and the actual measured values during process progress interval($0, n$] where the estimated parameters are given as $\hat{\lambda} = 327.87$ and $\hat{k} = 0.425$ by using a method of maximum-likelihood. Fig.6 shows the estimated expected number of remaining problems, $r(n)$. From Fig.6, it is found that there are 19 problems remaining at the end of static testing phase ($n = 4$).

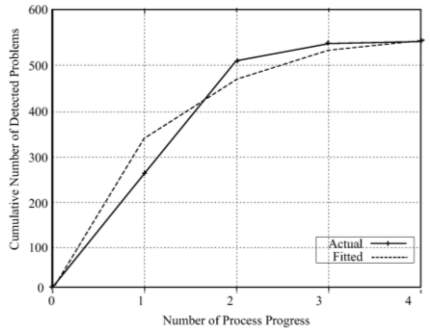

Fig. 5: The estimated cumulative number of detected problems, $E(n)$

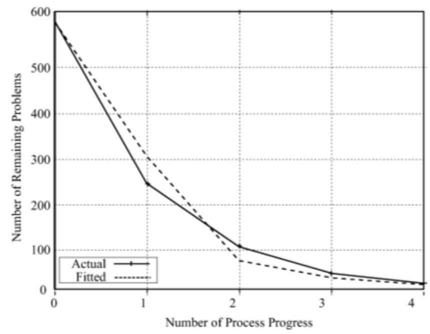

Fig. 6: The estimated expected number of remaining problems, $r(n)$

Further, the estimated instantaneous MTBP is obtained as shown in Fig.7. From Fig.7, it is found that the process progress is going well because the MTBP is growing.

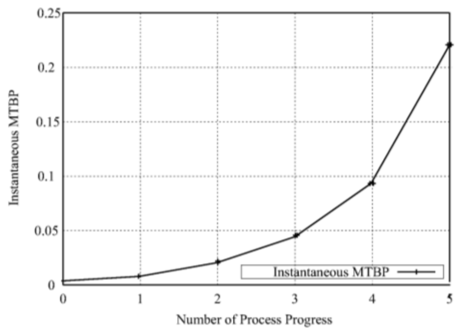

Fig. 7: The estimated instantaneous MTBP, $MTBP(n)$

6. CONCLUDING REMARKS

In this paper, we have quantitatively analysed the significant factors in the software development project by applying multivariate analysis to actual process data. As a result of multiple regression analyses, the factors which affect the product quality have been derived. Further, we have found that it is important to detect the problems in the early stage in the software development and to carry out risk management to reduce project risks due to human factors.

As a result of principal analysis, it is turned out that the project with a large reconstruction development size and a large planned man-day has a possibility that the number of problems may increase and the quality may decrease. Furthermore, the project where the reconstruction development size is small, and the planned man-day is large, and the number of problems detected in source code review large, is considered to increase the number of faults detected in the system testing.

And as a result of structural equation modeling, the flow of the process of a software development project can be modeled and it is found that the management of the upper process is important in terms of product quality. Furthermore, it is also found that a planned man-day is another factor which affects product quality.

Based on a result of logistic regression analysis, we have derived the regression equation that was effective for distinction of the product quality. We have also estimated the number of problems in the process progress, i.e., assessd project quality, by using the SRGM.

From the analysis results above, in order to lead a software development project to success, it is important to carry out *Quality-Oriented Software Management* [5],[6],[7].

In the future, we need to derive a highly accurate quality prediction model, and find the factors which influence management measures of product quality in order to lead a software project to become more successful.

7. ACKOWLEDGMENTS

This work was supported in part by the Grant-in-Aid for Scientific Research (C) (Grant No.25350445) from Ministry of Education, Culture, Sports, Science, and Technology of Japan.

8. REFERENCES

[1] S. Yamada and M. Takahashi: *Introduction to Software Management Model*, Kyoritu-Shuppan, Tokyo(1993).

[2] S. Yamada and T. Fukushima: *Quality-Oriented Software Management*, Morikita-Shuppan, Tokyo(2007).

[3] S. Yamada: *Elements of Software Reliability - Modeling Approach*, Kyoritsu-Shuppan, Tokyo(2011).

[4] P.B. Moranda： Prediction of software reliability during debugging, *Proc. 1975 Annual Reliability and Maintainability Symp.*, pp. 327--332, 1975.

[5] S. Yamada and A. Kawahara: Statistical analysis of process monitoring data for software process improvement, *Intern. J. Reliability, Quality and Safety Engineering*, Vol. 16, No. 5, pp.435--451(2009).

[6] S. Yamada, T. Yamashita, and A. Fukuta: Product quality prediction based on software process data with development-period estimation, *Intern. J. Systems Assurance Engineering and Management*, Vol. 1, No. 1, pp.72--76(2010).

[7] S. Yamada and J. Yamakawa: Statistical analysis based on software process monitoring data for effective project management, *J. Global Perspective on Engineering Management*, Vol.1, No.2, pp.44--50(2012).