A Quantitative Analysis of the Carnegie Mellon Software Engineering Institute's Capability Maturity Model/A Latent Variable Model

By

Shaoming Chang

A Dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirement for the degree of Doctor of Philosophy, the City University of New York

2000

UMI Number: 9986311

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

9/9/00
Date

Chair of Examining Committee

9/12/00
Date

Executive Officer

Dr. Mike Anshel

Dr. Ted Brown

Dr. Leon A. Kappelman
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

ABSTRACT

A Quantitative Analysis of the Carnegie Mellon Software Engineering Institute's Capability Maturity Model/A Latent Variable Model

By

Shaoming Chang

Advisor: Professor Howard Rubin

The Capability Maturity Model developed by Carnegie Mellon University's Software Engineering Institute (SEI/CMM) was the most commonly used method for enhancing software development/maintenance processes in the 1990s. The focus of SEI/CMM's research is to build a common production function for each of the five maturity levels (I to V) by comparing the behavior of the production functions of each level and then assessing the advantages and disadvantages of each production function for advancing the organization's maturity level.

This research chose the Latent Variable Model (LVM) to fit the production function because in the world of software engineering, models must frequently be compromised or modified to reflect the sampled data. By introducing latent variables we could reproduce the most typical practical production environment. The LVM we studied is composed of the following two group of variables:

- tangible variables, that is, sampled data (Function point, Duration, Man month, Team size)
- intangible variables, that is, latent variables (Management effort, Production process).

We also study Dr. Frederick Brooks' Man month myth among the five maturity levels. To analyze the differences, we then used the COCOMO model with an additional Team

size factor. Does the Team size overhead go away at higher maturity levels or does it tag along? One of the purposes of this study was to find an answer. To conclude the Production function. the Dynamic model was used.

To My wife LEILA, and two special young friends

ERIC and FLORA

ACKNOWLEDGEMENTS

I would like to thank Dr. Howard Rubin: he brought me into the world of Software Engineering. lent me his vision of the Maturity Framework. provided all the survey data for the modeling. coached and mentored my research all the way.

Special thanks to IBM statistician Dr. Sam Huang for all his advice and model building.

Thanks to my thesis committee for their support and encouragement : Dr. Mike Anshel. Dr. Ted Brown. and especially Dr. Leon A. Kappelman.

My sincerely gratitude to my parents and also to three special helpers:

- Mrs. Heidi Battaglia from Rubin Associate who sorted out the survey data
- A young lady Flora Chang who edited my draft and final paper
- Mr. Paul Bodine who put a thorough edit to my paper

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Background: IT Hardware/Software Environment

Computer hardware performance continues to increase dramatically (5). Central processing units, random access memory, and mass storage are improving their price-performance-ratio by orders of magnitude every decade (5). Unfortunately, the same improvement is not found in the world of software engineering.

Software engineering is the discipline of developing software. In 1972 F. L. Bauer offered the following succinct definition of software engineering: "The establishment and use of sound engineering principles (methods) in order to obtain economical software that is reliable and works on real machines"(27). In recent years, software engineering has become even more critical because of the demands of business automation—for example, the ATM machine, Brokerage Trading Systems, and the growing Internet economy beginning in the 1990s. As a result, the size of the software systems being constructed worldwide has increased dramatically (5). The changing cost structure of software development is reflected in the following chart from *Fortune* magazine (September 1989) (37):

APPLICATION	LINES OF CODE	DEVELOPMENT TIME(MY)	COST ($million)
Lotus	400,000	263	7.0
1989 Lincoln	83,000	35	1.8
Bank ATM	780,000	150	13.2
Checkout/Scanner	90,000	58	3.0
Space Shuttle	25,600,000	22,096	1,200.0

In the 1990s, demand in the software market grew as a result of the emergence of the Internet, fourth-generation computer languages and application generators, and commercial off-the-shelf and reuse-driven approach to software development (10). Barry Boehm's work provides a set of projections that illustrate the growth in software (5):

Software Cost ($ billion/year)

	USA/DOD	USA	World
1980	6	45	95
1985	11	70	140
1990		125	250
1995	36		
2000	63	400	800

To quote Boehm (writing in 1987), "By 1995, a 20 percent improvement in software productivity will be worth 90 billion dollars worldwide" (5). Despite this growth, software production is still plagued by missed schedules and cost overruns. Mark Paulk cited a catastrophic example:

"An unpublished review of 17 major Department of Defense software contracts found that the average 28-month schedule was missed by 20 months. And the Government Accounting Office has repeatedly reported on costs rising by millions of dollars"(35).

An ideal software development process should be predictable (21). The concept of predictability implies that cost estimates and schedule commitments are met with reasonable consistency and that the quality of the resulting products generally meets the users' needs (21). It is generally believed that if software engineering is to exhibit predictable behavior process issues are the key (13). This leads to two key questions:

- *Which process issues are significant enough to impact productivity?*

- *Is there a model that can quantify the behaviors of software processes?*

1.2 Millennium IT Workforce Labor Shortage

As demonstrated by Professor Howard Rubin's findings in the 1995 through 1998 Worldwide Benchmark Project, the annual demand for information systems professionals in U.S. corporations is growing 25 percent per year: "One estimate of the shortage is that 200,000 jobs currently remain unfilled (1996 survey). Based on the results of the 1996 Worldwide Benchmark Project, the amount of software that could be produced if the vacancies were filled could support an additional $500 billion in corporate revenue and $10 billion in associated income. The shortage is potentially taking a half-billion dollars in revenue out of the economy today" (41).

Thus, both Barry Boehm's observation and Dr. Rubin's survey demonstrate the importance of software processing.

1.3 Common Models in the 1970s and 1980s

In the 1970s to 1980s, several models were introduced to describe and analyze software process behaviors. Three of the most important were the following:

- Code-and-Fix model(6)

-- The stages of this model are: write the code, test, fix the problem, then loop back to test.

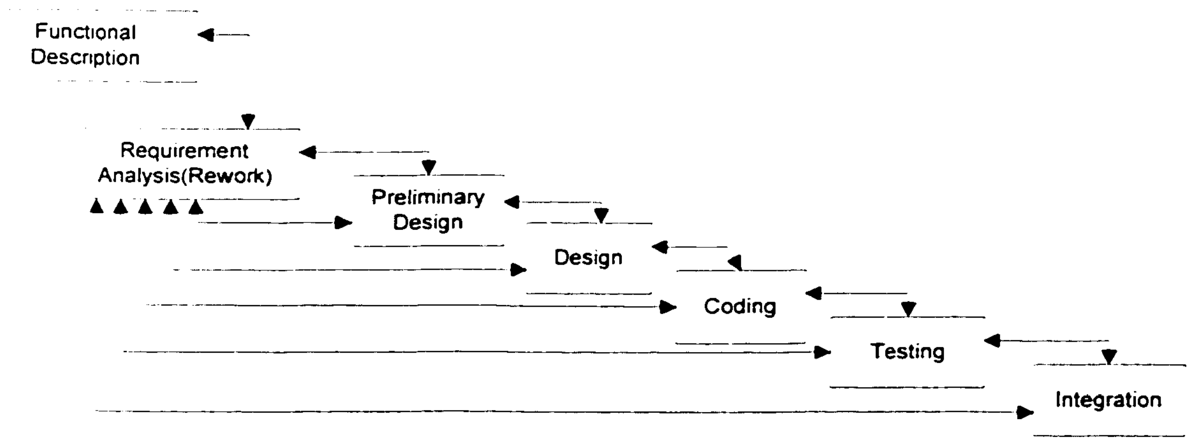
- Stagewise model(6)

-- In this model, software is developed in successive stages, namely: operational planning, developing operational specifications, developing coding specifications, coding, parameter testing, assembly testing, shakedown, system evaluation. (This model is a refinement of the code-and-fix model, in that it divides the entire process into stages that can be monitored or controlled by software engineers. This was the first model that provided information that could be analyzed.)

- Waterfall model(6)

-- This model is an improvement of the stagewise model. In the stagewise model, the process goes from the first stage then to the second stage and so on until they reach the last stage (system evaluation). The process could then go back to stage 1 and start all over. In the Waterfall model, however, evaluation or feedback could occur between any stage or phase.

George Hansen offers a chart (17)



Clearly, the waterfall model provides more flexibility for making any change or adjustment to the process as well as better control.

As these models emerged, several quantitative software estimation models were also introduced, such as ESTIMACS, COCOMO, and SLIM (4). They all take a quantitative approach to estimating resource allocation, and they all model the productivity of the software processes. Putnam's SLIM (Software Life Cycle Model) provides a good example of how these models work:

S_line_of_code = F (Life_cycle_in_man_year, Develop_time_in_year, Technology_const)

Putnam's model provides a production function, *(delivered_line_of_code)*, which is a function of the following three process behaviors: *life_cycle_in_man_year*, *development_time_in_year*, and *Technology_constant*. The SLIM model introduced good control and prediction into the software development process.

1.4 COCOMO and COCOMO II

In 1981, the COnstructive COst Model (abbreviated COCOMO) provided a method for gaining a comprehensive estimation of software cost (3). Based on *Source_line_of_code* combined with the Waterfall model, the COCOMO model provided a decent scheme for predicting the **Man month** and **Schedule** if there is a reasonable prediction of *Total_line_of_code*.

As technology evolves, new software development practices have emerged, including nonsequential and rapid-development process models; reuse-driven approaches involving commercial off-the-shelf packages, reengineering, and applications composition and generation capabilities; object-oriented approaches supported by distributed middleware; and software process maturity initiatives.(10) Because the COCOMO model has experienced difficulties in modeling these new practices, COCOMO 2.0 has developed in 1995 (11). (The name "COCOMO II" replaced COCOMO 2.0 in 1997.)

The COCOMO II model has three submodels, Application Composition, Early Design, and Post-Architecture. COCOMO II can use both *Source_line_of_code* or *Function_point*. The three submodels operate in the following ways:

- The Application Composition submodel uses involves using prototyping to resolve potential high-risk issues such as user interfaces, software/system interaction, performance, or technology maturity.(11)
- The Early Design submodel involves explosive exploration of alternative software/system architectures and concepts of operation. The corresponding

COCOMO II capability involves the use of function points and a small number of additional cost drivers. (11)

- The Post-Architecture model involves the actual development and maintenance of a software product. This submodel has a set of seventeen effort multipliers and a set of five scale factors (10). It estimates for the entire life cycle of the software production.

The evolution from COCOMO to COCOMO II represents the progress being made in changing, replacing, or adding the cost drivers (effort multiplier) and scaling factors. The COCOMO II is suitable for the current as well as upcoming software development environments (08).

COCOMO/COCOMO II diagnoses the software processes from Micro Perspectives. Just as microeconomics analyzes economic behavior on a local or detailed level, but COCOMO/COCOMO II provides a very effective way of estimating the cost, required effort, and schedule of a software development process in a case by case bases.

In the late 1980s, a revolutionary framework for categorizing software processes was developed. This framework, known as the SEI Maturity Framework, groups software processes according to certain common characteristics.

1.5 The SEI Maturity Framework

In 1987, the Software Engineering Institute (SEI) of Carnegie Mellon University developed a software process maturity framework (21) and maturity questionnaire (20) in response to a request by the U.S. government to provide a method for assessing the capability of their software contractors. This project was sponsored by U.S. Department of Defense and was motivated by the following facts (21):

- The cost of U.S. software cost was growing by 12 percent each year
- Schedule overruns were common
- Software often adversely affects the effectiveness of weapon systems

Watts Humphrey evolved the maturity framework at the SEI from his previous work at IBM in 1986. He refined the concept--his vision--of maturity levels and established the foundation for its current use throughout software industry (35).

Today, organizations can use the maturity questionnaire that was jointly developed by SEI and Mitre Corp. to assess an their maturity level (35). In the original model, there were 101 item questions requiring yes or no answers. Each question was associated with one maturity level. There were also special questions that were designated as being key to each specific maturity level. To qualify for a certain level, 90 percent of the key questions and 80 percent of all the questions for that level had to receive yes answers.

The maturity levels themselves are hierarchical. Level 2 must be attained before Level 3. Levels 2 and 3 must be attained before Level 4, and so forth. The framework has a total of five levels of process maturity, defined as follows (21):

- Level I. Initial --Ad hoc: the organization has no control at all.
- Level II. Repeatable --The organization has achieved a stable process with a Repeatable level of statistical control by initiating rigorous project management of commitments. cost. schedule. and changes.
- Level III. Defined --The organization has defined the process to ensure consistent implementation and to provide a basis for improved understanding of the process.
- Level IV. Managed --The organization has initiated comprehensive process measurements beyond those of cost and schedule performance.
- Level V. Optimizing --The organization now has a foundation for continued improvement and optimization of the process.

Based on this framework. two newer frameworks have evolved to help the software organizations advance to higher maturity levels. They are CMM (Capability Maturity Model) and PM-CMM (People Management Capability Maturity Model).

Between 1990 to 1993. the ESPRIT (European Software Program for Research in Information Technologies) Project developed the BOOTSTRAP model. This model is based on SEI's CMM. the ISO 9000 series of standards. and the European Space Agency's generic process model (42). BOOTSTRAP has a different process assessment and improvement framework than the CMM model(42):

- It provides detailed capability profiles and maturity levels.
- It gives profiles and the levels of organizations and projects separately.

- It allows benchmarking.
- It provides ISO 9000 Gap analysis.
- It allows SEI maturity levels to be calculated.
- It includes immediate feedback from the organization assessed.

1.5.1 CMM (Capability Maturity Model)

The CMM defines Key Process Area (KPAs) to help organizations enact the changes needed to advance to a higher level (34). The KPAs to advance from Level I to II are as follows:

--Software configuration management

--Software quality assurance

--Software subcontract management

--Software project tracking and oversight

--Software project planning

--Requirements management

The KPAs to advance from Level II to III are the following:

--Peer review

--Intergroup coordination

--Software product engineering

--Integrated software management

--Training program

--Organization process definition

--Organization process focus

The KPAs to advance from Level III to IV are:

--Software quality management

--Quantitative process management

And the KPAs to advance from Level IV to V are:

--Process change management

--Technology change management

--Defect prevention

1.5.2 PM-CMM (People Management Capability Maturity Model) (15)

The PM-CMM consists of a five-level framework that establishes successive foundations for continuously improving employee talent (15):

- Initial Level (Level I)

The organization does not provide a consistent environment for managing talent, managers are using people skills inconsistently, and there is no basis for improving. The KPAs for advancing to the next higher level are:

--Work environment and participatory culture

--Management-guided practices for setting up consistent productivity behaviors

--Standardized practices for compensation and reward that are also communicated to all employees

--Training and career development

--Performance management. that is. simply applying a sound metric

--Developing staffing and people management value–including recruiting. training. and retraining

- Repeatable Level (Level II)

A repeatable capability can institutionalize the effective performance of basic people management activities. To advance to higher levels. people management needs to formulate strategic. operational. and tactical unit plans for developing talent. The KPAs are as follows:

--Team building

--Competence-based practices

--Competence development

--Knowledge and skill analysis

--People management planning

- Defined Level (Level III)

At this level. the organization can develop a common organizational culture. To advance to a higher level. the organization needs to have good metrics. The KPAs are as follows:

--Organization performance alignment

--Organization competence management

--Quantitative people management

- Measured Level (Level IV)

This level of people management capability gives the organization an important predictor of the trends in its business capability. To advance to higher level. the KPAs are:

--People management innovation

--People management improvement

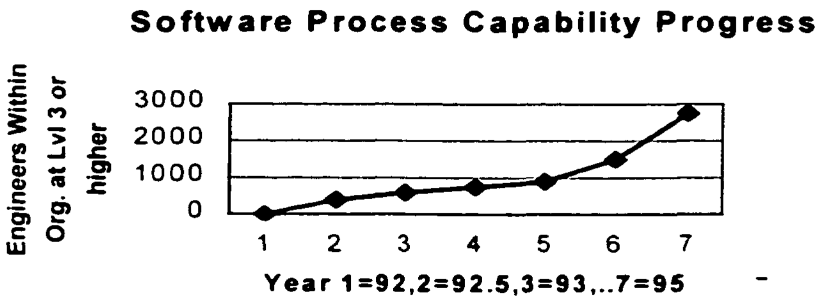
- Optimizing Level (Level V)

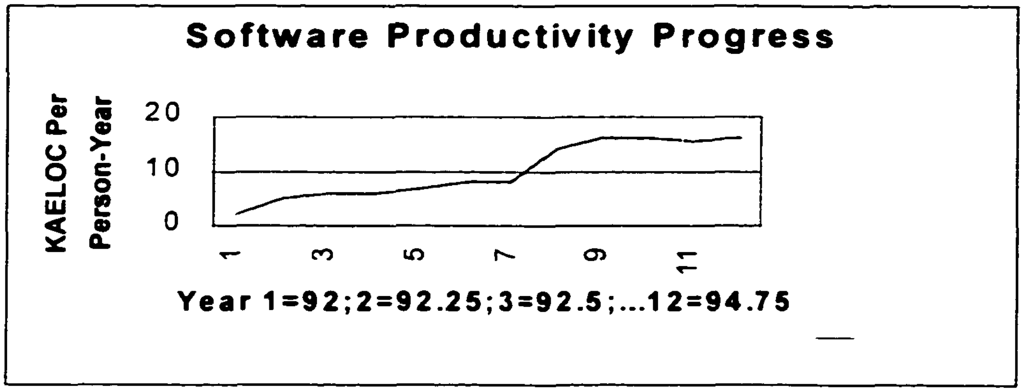
The organization has the means for identifying opportunities to strengthen its people management proactively. Its people management system is creating a culture of performance excellence.

1.6 The SEI's Maturity Framework Today

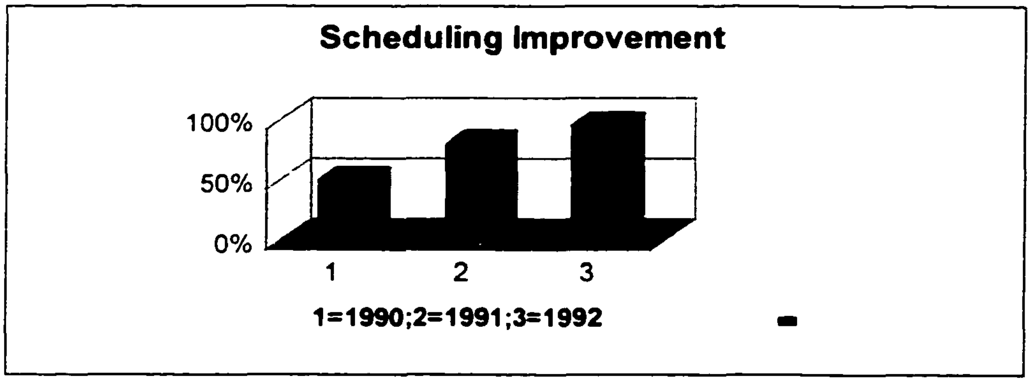
The SEI's maturity frameworks have been widely adopted in the software engineering world (39). The CMM maturity framework is generally viewed as the best way to characterize the behaviors of any software organization and to map out the capability/productivity of that organization. As a result, the SEI's maturity framework has become the most widely accepted assessment standard. It is considered the best practice available to software organizations for improving their productivity/quality by advancing to higher maturity levels (39).

Now let's take a look at some software organizations that adopted the SEI framework and benefitted from advancing to higher maturity levels. The following figures that show, respectively, the improvement in Motorola's engineers from 1992 through 1995 by advancing to Level III (33) and the increase in software productivity at Motorola when it practiced SEI/CMM in the next page.





The following figure shows Schlumberger's improvement in scheduling from 1992 to 1995 (46). Through its Software Improvement Program Schlumberger's adherence to schedule improved from 50 percent in 1990 to 80 percent in 1991 and to 95 percent in 1992. They only practiced the CMM model.



2 MODELING

Scientists trying to understand behavior rarely have the luxury of the simple bivariate experiment in which a single independent variable is manipulated and the consequences are observed for a single dependent variable. The more common method for dealing with a model is to use multiple variables. some of which are observed and some unobserved. or latent.

2.1 Introduction to Latent Variable Model

The Latent Variable Model has two types of variables. In the research the following variables are used to build the production model:

- Observed variables (Man_month. Function_point. Team_size. Duration)
- Unobserved variables / latent variables (*Management_Effort. Production_Process*)

As Figure 1 shows. these two types of variables are connected by arrows. which represent the causal relation.

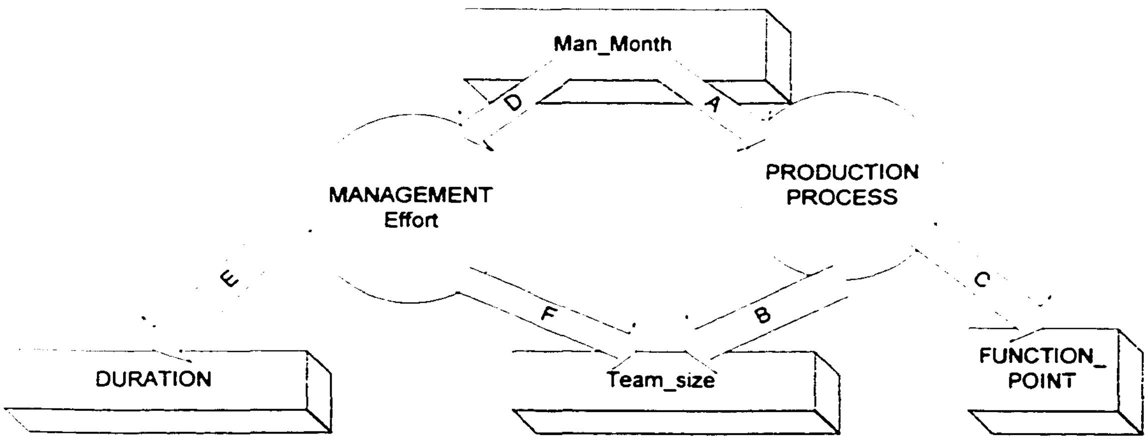


Figure 1 : Latent Variable Model/Sample

Two models were studied in this research:

- Model without latent variable--Modified COCOMO Model
- Model with latent variables--Integrated Production Model

The following two sections will discuss each of these models.

2.2 Modified COCOMO Model

As shown in Figure 2, the COCOMO model used three variables (*Man_Month, Function_point, Duration*) (4). Brooks's law suggested a strong relation between the **Team size** and the Scheduling (13). We therefore we built this modified COCOMO model with an additional variable (*Team_size*) and discussed the consequences and influences of this variable (if any).

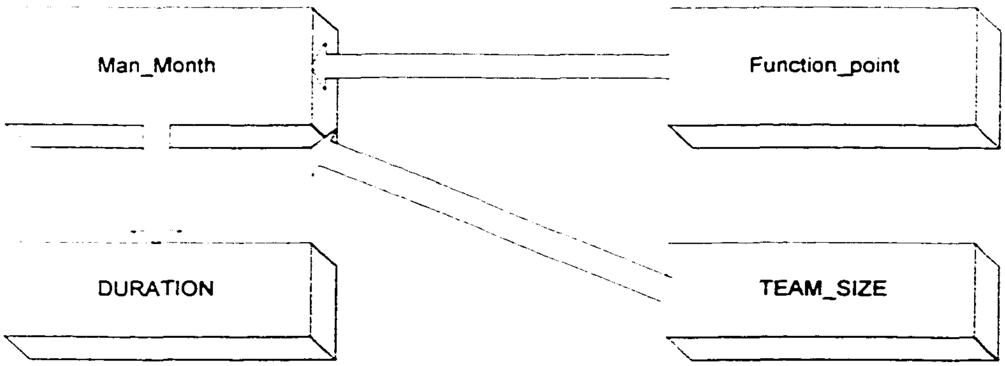


Figure 2 : Modified COCOMO Model

2.3 Integrated Production Model

In search of the role management/production played in the software process mystery, two latent variables (*Management_effort, Production_process*) and four observed variables (*Function_point, Man_month, Duration, Team_size*) were selected to build the model. As a result, three models (potential combinations—see Figures 3, 4, and 5) were tested/analyzed and discussed later in this section.

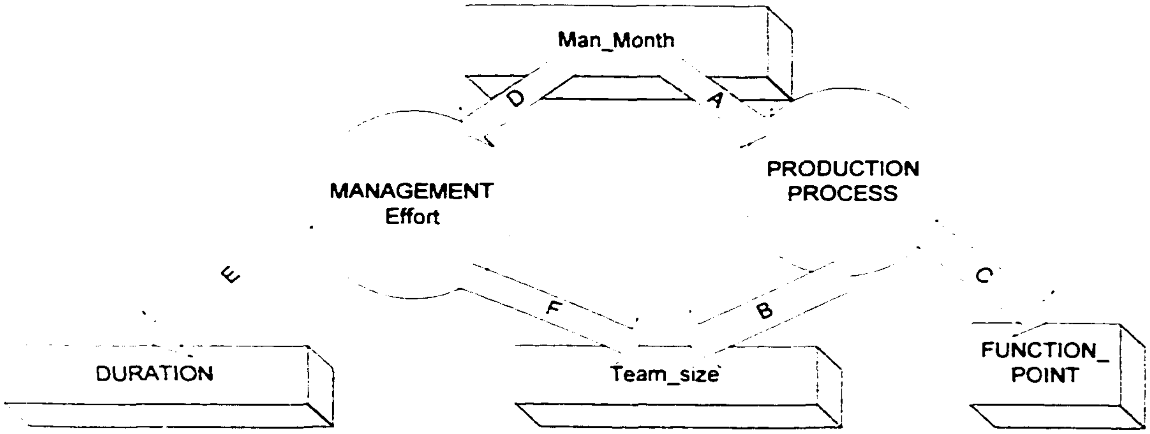


Figure 3 : Test Model I

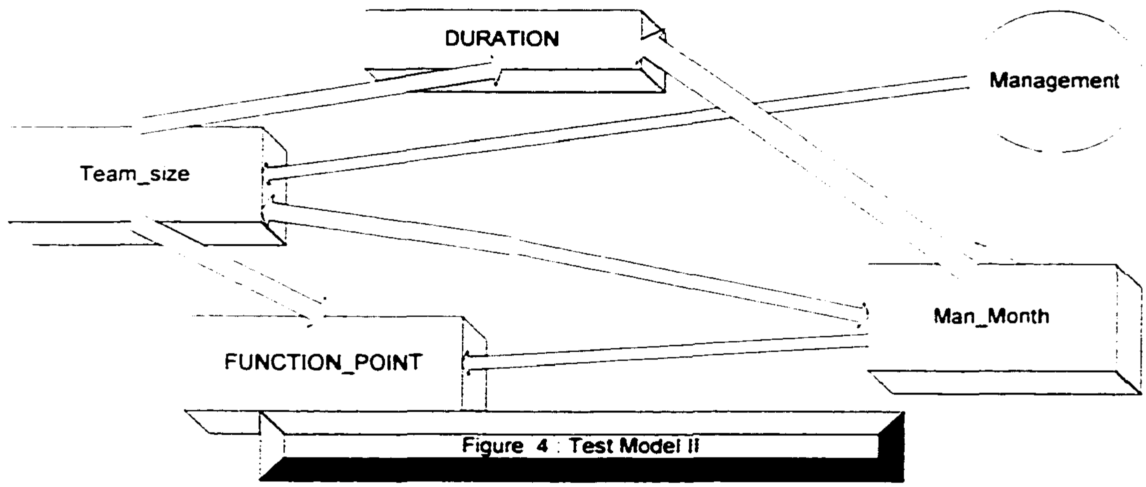


Figure 4 : Test Model II

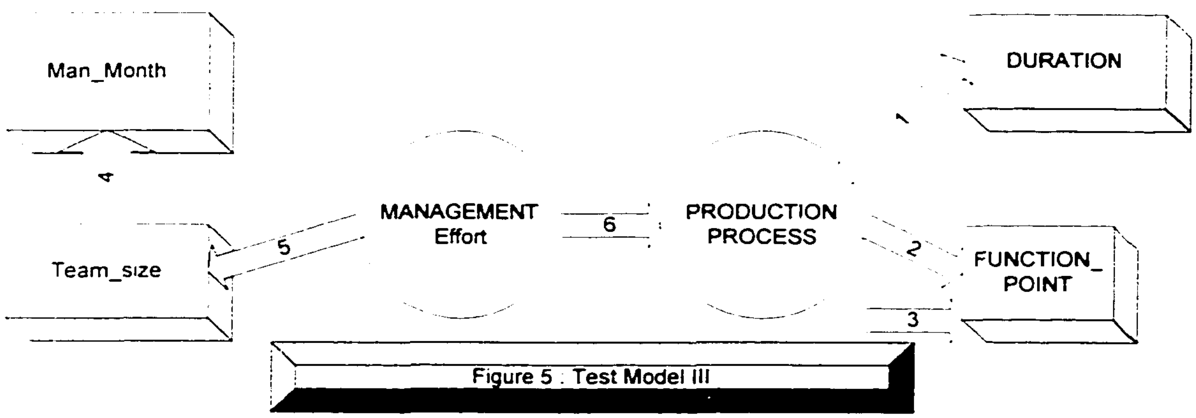


Figure 5 : Test Model III

This research used the structure equation approach (SEA, see Section 3.2) from four observed variables. The covariance of the observed variables are

Q = 4 + 3 + 2 +1 = 10 and Degree of Freedom = Q – C, where C is elaborated in column 6 of the following table:

Tested Model	Latent Variable	Observed Variable	Covariance	Implied Equation	Model Variables **C**	Goodness of Fit
I	2	4	10	6	10 = 6 + (Variables) 3 + (Inter dependence) 1 (constant)	Degree of Freedom 0 - F Test meaningless no room for error
II	1	4	10	8	13 = 8 + (Variables) 5 (Inter dependence)	Equation > Variables (Over Deterministic) No Goodness of Fit
III	2	4	10	5	8 = 5 + (Variables) 2 + (Inter dependence) 1 (cluster)	10 > 8, DoF = 2, Reasonable Choice

Analysis of Models :

In Model I there are too many unknown variables that need to be resolved. If using the structure equation approach, each arrow will form an equation, so six equations will be formed. In Figure 2, all three relations, AD, EF, and BC, form an **interdependence correlation**. This will reduce another three degrees of freedom, along with one constant, leaving a total of ten variables that need to be estimated. So this will not be a good fit because there is no room for error.

In Model II, eight equations to fit five variables is overdeterministic and would be a bad choice. Model III had a decent value for "goodness of fit," so it was chosen for this study.

3 FITTING THE LATENT VARIABLE MODEL

The Latent Variable Model is usually represented as a Path Model. like the model shown in Figure 6.

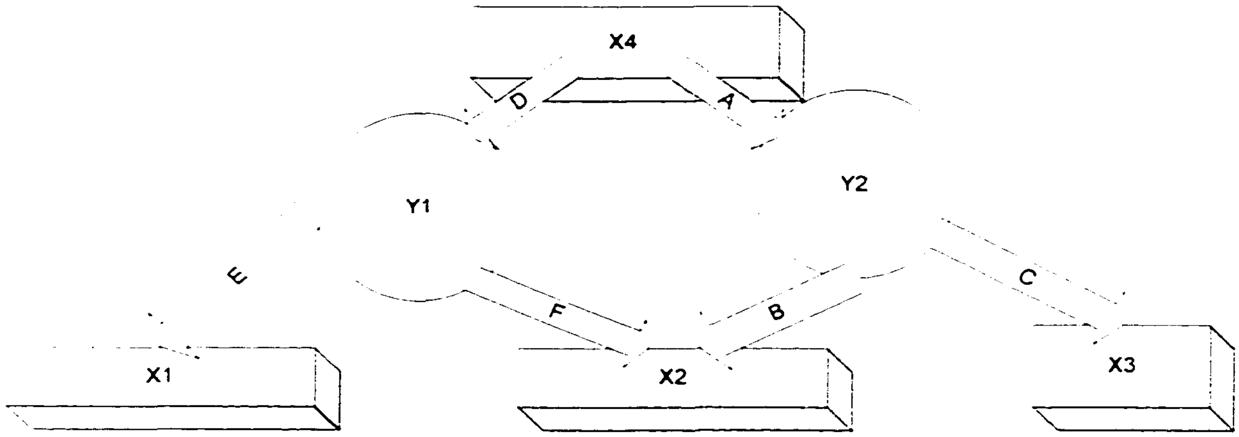


Figure 6 : Path Model Example

The rectangle represents the observed variables (X1. X2. X3. X4). The circle represents the latent variables (Y1. Y2). The arrow shows the causal relation. and all relations are linear. Two methods are commonly used to build (fit) the model: the path approach and the structure equation approach. We will apply each in the next two sections.

3.1 Path Approach

From the right side of Figure 6. the causal relations can be drawn as follows:

Covariance (X4. X3) = A * C
Covariance (X4. X2) = A * B + D * F
Covariance (X2. X3) = B * C

Three variables, three equations--so the model provides a "just determination" situation.

The left side is the same as the right side:

Covariance (X4, X1) = D * E
Covariance (X4, X2) = D * F + A * B
Covariance (X1, X2) = E * F

Again, three variables, three equations--so it provides a just determination situation, that is, the model can be fitted with the proper values of A, B, C, D, E, and F.

3.2 Structure Equation Approach

Each of the equations in Section 3.1 expresses a downstream variable as a function of the causal paths leading into it. There will be as many equations as there are downstream variables. Refer to the Path Model shown in Figure 7.

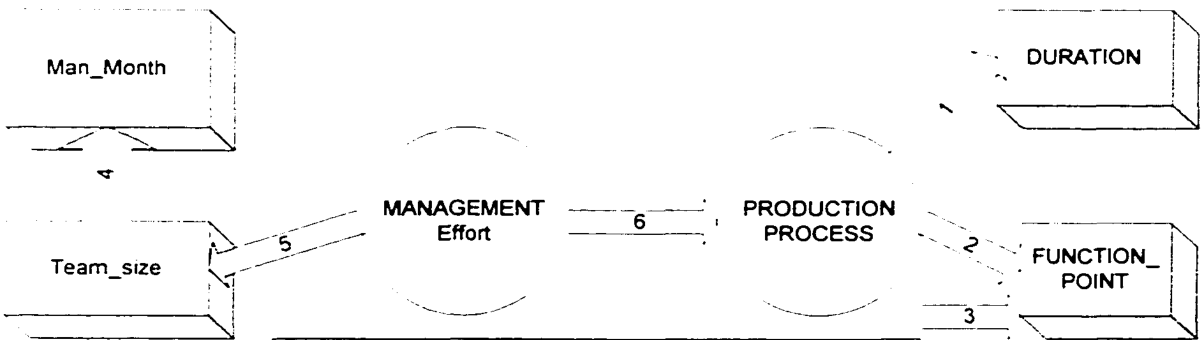


Figure 7 : Example Model / Structure Equation Approach

For this Path Model, the following hold true.

Function_Point = Production_Process * Alpha2 + Code * Alpha3 (Code is for clustered)
Man_Month = Team_size * Alpha4
Duration = Production_Process * Alpha1
Team_size = Management_Effort * Alpha5
Production_Process = Management_Effort * Alpha6

This research. the structure equation approach(SEA) was adopted.

3.3 Goodness-of-Fit Criteria

A variety of criteria have been used to describe how closely the correlation or covariance implied by a particular set of trial values conforms to the observed data. The three most commonly used criteria are the following:

- Ordinary least squares (OLS)
- Generalized least squares (GLS)
- Maximum likelihood (ML)

Refer to the Path Model shown in Figure 8.

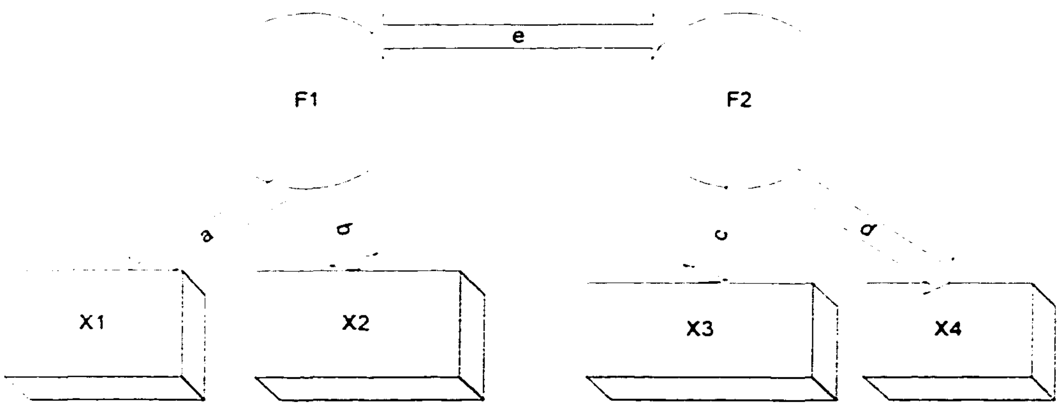


Fig 8 : Example Model / Calculation

The example Model (Fig 8) from the observed variables X1. X2. X3. X4 for the covariance is presented in the following table:

	X1	X2	X3	X4
X1	1			
X2	a	1		
X3	b	c	1	
X3	d	e	f	1

The various criteria (or discrepancy functions) are the differences between corresponding elements of the observed and implied covariance matrices. And the different methods to weight the differences, made the different optimization approach.

For the Model in Fig 8, let the vector S= (a,b,c,d,e,f)', and C is the corresponding estimated vector. In this case, the discrepancy function may be expressed as follows:

$$(S - C)^{t} W (S - C)$$

where $(S - C)^{t}$ means "transform the vector." W is a weight matrix, and different versions of it yield different criteria. When the W is an identity matrix, it becomes $(S - C)^{t} (S - C)$, Ordinary Least Squares.

If the observed variables have a distribution that is multivariate normal, the general expression $(S - C)^{t} W (S - C)$ can be simplified to

$$\tfrac{1}{2}\, tr\, ((S - C)\, V)^{2}, \quad (21)$$

where the tr transfers to the trace of a matrix (the sum of its diagonal elements), and V is a weight matrix. The choice of weight matrix defines the following:

$V = I$ OLS, ordinary least squares

$V = S^{-1}$ GLS, generalized least squares

$V = C^{-1}$ ML, maximum likelihood.

The goodness-of-fit index, $GFI = 1 - Tr(V(S - C))^{2}) / Tr((WS)^{2})$

3.4 Covariance versus Correlation in Model Fitting

In general, if one wants to take variance differences into account when comparing different groups, different times, and so on, one should analyze the covariance matrix.

Unless one wishes to ignore variance differences, one might choose to analyze standardized variables, that is, a correlation matrix.

Using a correlation matrix involves some technical issues (26). The statistical theory underlying *maximum likelihood* and *generalized least squares* solutions has mostly been developed for the case of covariance matrices rather than for correlation matrices.

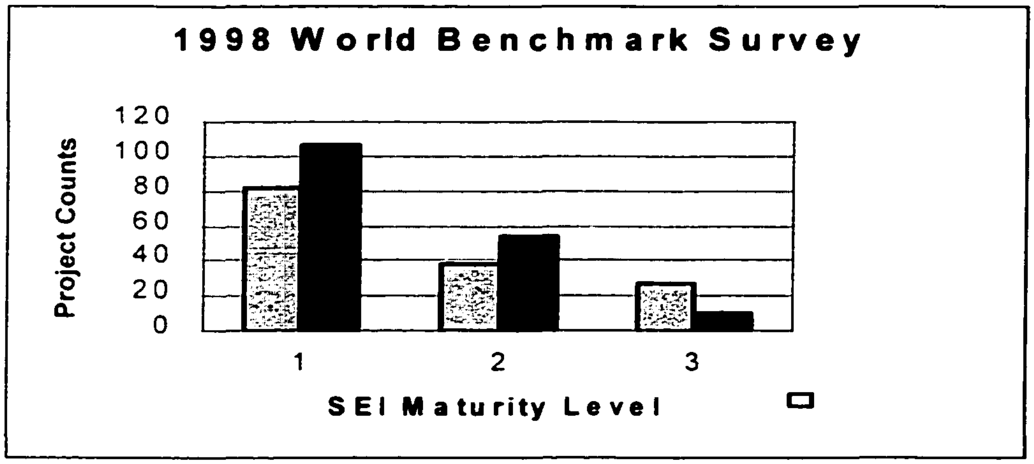
The covariance is used in this study.

4 THE SURVEY DATA

More than five thousand companies were surveyed in the 1998 World Benchmark Project: only six hundred of them practiced SEI/CMM. The data we used are the completed software projects from those six hundred companies, with the completed metrics, that is, *Function_point(line_of_code), Duration, Team_size, Man_month.*

The count of the completed records breaks out as shown in the following table and figure (the Cluster column shows the cluster analysis data):

Maturity Level	Complete Project in Function_point	Complete Project in Line_of_Code	Record used	Cluster A:B
I	82	106	82	79 : 3
II	38	54	92	90 : 2
III	27	10	37	36 : 1



4.1 Preliminary Data Selection

The Level I data is the candidate for the model's "goodness of fit." Using Capers Jones's conversion table, convert all *Line_of_Code* metrics into *Function_point* metrics (29). Because *Function_point* has such a large range, we take the natural log function and use the logged data to build the model.

- *Level I Data.* The combined Level I data did not pass the statistical criteria. The combined data only showed as a single cluster, and the correlation are not significant (the P-value was less than 0.05; it was too small to accept these criteria). Also, when we used the combined data, the combined data did not fit any of the aforementioned models.

It may be that the *Line_of_Code* metric does not fit and so cannot be converted into *Function_point* for the Level I projects. As a result, in Level I data, we only choose the projects with *Function_point* metric. The cluster analysis yielded two clusters (79: 3) out of the 82 projects, and the 82 records also showed reasonable fit for our statistical model. Although the Level I organization could be an ad hoc environment, we still attempted to study it. The correlation for the Level I *Function_point* metric data, with records = 82 and cluster as (79:3), is shown in the following table.

	Function_pt	Man_month	Duration	Team_size
Function_pt	1	0.40436	0.27195	0.11373
Man_month	0.40436 p=(0.0002)	1	0.42826	0.67189
Duration	0.27195 p=(0.0135)	0.43826 p=(0.0001)	1	0.18576
Team_size	0.11373 p=(.3090)	0.67189 p=(0.0001)	0.18576 p=(0.0947)	1

Some P-values are acceptable (if 0.05 is the criterion), but only (Man_month, team_size) showed some correlation. (0.7 > 0.67--correlation > 0.5).

- *Level II Data.* Level II data converted all *Line_of_code* metrics into *Function_point* metrics and combined them with native *Function_point* metrics projects. The correlation for the Level II combined data, with combined records = 92 and cluster as (90:2), is shown in the following table:

	Function_pt	Man_month	Duration	Team_size
Function_pt	1	0.12765	-0.0145	0.02565
Man_month	0.12765 p=(0.2253)	1	0.3248	0.6333
Duration	-0.01453 p=(0.8907)	0.3248 p=(0.0016)	1	-0.05234
Team_size	0.02565 p=(0.8082)	0.6333 p=(0.0001)	-0.0523 p=(0.6202)	1

The P-value showed much greater improvement for the Level II data than for the Level I data. It showed the following:

Weak correlation	<*Man_month* versus *Team_size*>	0.6333
Light correlation	<*Man_month* versus *Duration*>	0.3248
No correlation	<*Man_month* versus *Function_point*>	012765
No correlation	<*Function_point* versus *Duration*>	-0.01453
No correlation	<*Function_point* versus *Team_size*>	0.02565
No correlation	<*Duration* versus *Team_size*>	-0.0523

- *Level III Data.* For Level III data we convert all *Line_of_code* metrics into *Function_point* metrics and then combine them with native *Function_point* metric projects. The correlation for the Level III combined data, with combined records = 37 and cluster as (36:1), is shown in the following table:

	Function_pt	Man_month	Duration	Team_size
Function_pt	1	0.44274	0.45706	0.22635
Man_month	0.44274 p=(0.0061)	1	0.81571	0.78109
Duration	0.45706 p=(0.0045)	0.81571 (0.0001)	1	0.54822
Team_size	0.22635 p=(0.1779)	0.78109 p=(0.0001)	0.54822 p=(0.0004)	1

The P-value somehow dropped out of the significance test. but it suggested higher correlation for Level II data than for Level II data. It showed the following:

Strong correlation	< Man_month versus Team_size >	0.78109
Strong correlation	< Man_month versus Duration >	0.81571
Weak correlation	<Man_month versus Function_point>	0.44274
Weak correlation	< Function_point versus Duration >	0.45706
Light correlation	< Function_point versus Team_size >	0.22635
Weak correlation	<Duration versus Team_size >	0.54822

From our observations. Level II and Level III data showed the most homogeneous nature. This taught us a significant fact: Level II and Level III organizations provide reliable/stable metrics for the software engineering process.

In general. if the correlation between two observed variables is greater than 0.7. they have a strong correlation. If the correlation is between 0.5 to 0.7. they have weak correlation. From our correlation analysis. only *Team_size* versus *Man _month* showed a weak/strong correlation among all the Level I. Level II. and Level III data. No other correlation had a consistent weak/strong correlation from Level I to Level III. So. the correlation analysis among our observed data suggested only the casual relation of *Team_size* versus *Man_month*.

5 BUILDING THE INTEGRATED MODEL

5.1 The Integrated Model

Note the following in the Integrated Model shown in Figure 9:

- The circles are the latent variables. *Management_effort* and *Production_process*
- The triangles are the observed variables. *Man_month. Team_size.Duration.*

Function_point.

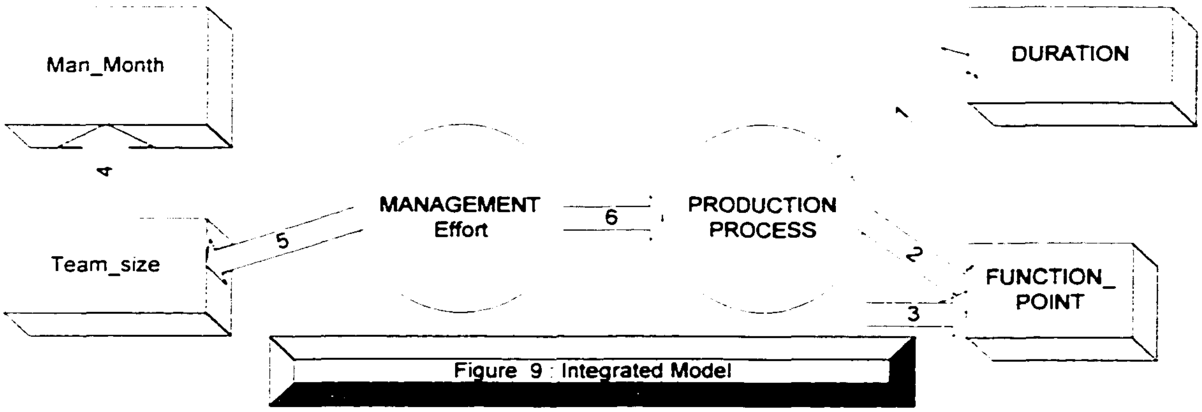


Figure 9 : Integrated Model

The arrow shows the causal relation. and we assume the following relations:

--The 3 is used to differentiate the clusters within *Function_point.*

--*Management_effort* has the control of the two relations.

--*Team_size*:5 and *Production_Process*:6 (the numbers correspond to the drawing)

--*Man_month* is the downstream variable of *Team_size*:4

--*Production_process* influences the two observed variables. *Duration:1* and *Function_Point:2*. and it is also the downstream variable of Management:6

Because the data range is too large. we have to transform it by taking natural log functions. then use the transformed data to build up our SEM (structure equation model). We tried three optimization methods: least squares. generalized least squares. and maximum likelihood. The least squares and generalized least squares did not converge, nor did the goodness of fit pass the T-test. Only the maximum likelihood had the good result. Next we will describe the results when we used maximum likelihood approach.

The SAS/Statistics package started to define this Path Model as a structure equation model. *Management_effort* will represented by "Management," and *Production_process* will represented by "Production." The initial estimates are as follows:

Function_point = ALPHA2 *PRODUCT + ALPHA3 * CODE + Error_02
Man_month = ALPHA4 *Team-size + Error_03
DURATION = ALPHA1 * PRODUCT + Error_01
Team_size = ALPHA5 * Management + Error_04
PRODUCT = ALPHA6 * Management + Error_05
The initial estimation; ALPHA2=0.5; ALPHA1=0.5; ALPHA5=0.5; ALPHA6=1

We then use SAS package to fit the model.

5.2 Level I Data SAS Run Result

For the Level I SAS Run results, the GFI (Goodness of Fit Index) was 0.8575, indicating a good fit. The fitted model was as follows (all variables are log transformed):

```
NFP     =   5.8509*FPRODUCT +  0.8034*CODE   +  1.0000 E_02
  Std Err    0.0280 ALPHA2      0.7652 ALPHA3
  t Value  208.6298             1.0499

MAN_MTH =    1.9369*TEAM_SZ +  1.0000 E_03
  Std Err    0.0785 ALPHA4
  t Value   24.6701

DURATION=    2.0450*FPRODUCT +  1.0000 E_01
  Std Err    0.0699 ALPHA1
  t Value   29.2422

TEAM_SZ =   2.9636*FMGMT   +  1.0000 E_04
  Std Err   0.0193 ALPHA5
```

t Value 153.3783

Latent Variable Equations

FPRODUCT=	0.8510*FMGMT	+ 1.0000 E_05
Std Err	0.0708 ALPHA6	
t Value	12.0202	

where *NFP* represents *Function_point*; *Man_mth* represents *Man_month*; *Team_sz* represents *Team_size*; *FMGMT* represents *Management_Effort*; and *FPRODUCT* represents *Production_Process*.

5.3 Level I Model

So for a Level I organization, the Integrated Model is built as follows:

Function_point = PRODUCTION $^{5.8509}$
Or = PRODUCTION $^{5.8509}$ * $e^{0.7652}$ for second cluster data
Man_month = Team_size $^{1.9369}$
Duration = PRODUCTION $^{2.0450}$
Team_size = Management $^{2.9636}$
PRODUCTION = Management $^{0.8510}$

The Level I Model showed the following relations:

- Function points have a very strong relation to the Production process, by a power of 5.8509. This may fit the observation that in a Level I environment, the individual has great influence over productivity, and the individual's capability might dominate the whole process. In an ad hoc environment, any direct improvement to the software process would yield a significant productivity increase.
- Team size shows a good overhead to Man month, almost the square of it, 1.9369.

- Duration has a positive effect on the Productivity. that is. a better Production process will cause a big overhead to the Scheduling (Duration of the project).
- Team size has a positive relationship to Management effort. The data suggested that in a Level I environment. Management effort tends to increase the Team size in an exponential manner.
- Production/management has an almost 1:1 relation to the whole process.

5.4. Level II Data SAS Run Result

For the Level II data SAS Run results. we used the same format for the process. We used the combined records (38 *Function_point* evaluated data. plus 54 records. which are converted from *Line_of_code* into *Function_point*). which yielded a total of 92 records. As a result. the Goodness of Fit Index (GFI) we obtained in the model was 0.6905. which is acceptable.

We obtained the following relation:

```
NFP     =   1.8004*FPRODUCT + 0.7694*ORIG   + 1.0000 E_02
   Std Err   0.0726 ALPHA2     0.2480 ALPHA3
   t Value  24.7934            3.1021

MAN_MTH =   2.0022*TEAM_SZ  + 1.0000 E_03
   Std Err   0.0629 ALPHA4
   t Value  31.8213

DURATION=   0.7234*FPRODUCT + 1.0000 E_01
   Std Err   0.0463 ALPHA1
   t Value  15.6280

TEAM_SZ =   0.9600*FMGMT    + 1.0000 E_04
   Std Err   0.0558 ALPHA5
```

t Value 17.1952

Latent Variable Equations

FPRODUCT= 1.0776*FMGMT + 1.0000 E_05
Std Err 0.0953 ALPHA6
t Value 11.3125

5.5 Level II Model

For a Level II organization, the Integrated Model builds as follows:

Function_point = PRODUCTION $^{1.8004}$
Or = PRODUCTION $^{1.8004}$ * e $^{0.7694}$ for second-cluster data
due to 79:3 cluster distribution, the second cluster could be ignored)

Man_month = Team_size $^{2.0022}$
Duration = PRODUCTION $^{0.7234}$
Team_size = Management $^{0.9600}$
PRODUCTION = Management $^{1.0776}$

Function points showed how the Production process was positively affected and demonstrated the improvement of the Production process. The exponential factor of 1.8 means that the Production increase would bring a significant increase in Function points. This might suggest that the Level II organization is a good candidate for introducing new techniques. This suggestion was as we anticipated: Level II organizations are ready to adopt new techniques, procedures, and so on. In addition,

- Man month has a positive relation with Team size. This explained that the communication overhead tends to be huge when Team size is big.
- Duration has a positive relation but much less effect on the Productivity.
- Team size has a positive and stronger relation to the Management effort.
- Productivity has a positive and stronger relation to the Management effort.

5.6 Level III data SAS Run Result:

we also used the combined record (27 records that were using the *Function_point* metric. 11 records which were converted from Line_of_code). total of 35 records used the same process. and

the Goodness of Fit Index (GFI) was 0.7099. which is good. We obtained the following relations:

```
NFP     =    2.9697*FPRODUCT + 1.1923*ORIG    + 1.0000 E_02
   Std Err   0.1216 ALPHA2     0.4899 ALPHA3
   t Value   24.4187           2.4336

MAN_MTH =    1.7872*TEAM_SZ  + 1.0000 E_03
   Std Err   0.0486 ALPHA4
   t Value   36.7659

DURATION=    0.9210*FPRODUCT + 1.0000 E_01
   Std Err   0.0616 ALPHA1
   t Value   14.9533

TEAM_SZ =    1.3102*FMGMT    + 1.0000 E_04
   Std Err   0.1051 ALPHA5
   t Value   12.4695
```

Latent Variable Equations

```
FPRODUCT=    0.8196*FMGMT    + 1.0000 E_05
   Std Err   0.1110 ALPHA6
   t Value   7.3838
```

5.7 Level III Model

So, for a Level III organization the Integrated Model builds as follows:

$$Function\_point = PRODUCTION^{2.9697}$$

$$Or = PRODUCTION^{2.9697} * e^{1.1923}$$ for second-cluster data

$$Man\_month = Team\_size^{1.7872}$$

$$Duration = PRODUCTION^{0.9210}$$

$$Team\_size = Management^{1.3102}$$

$$PRODUCTION = Management^{0.8196}$$

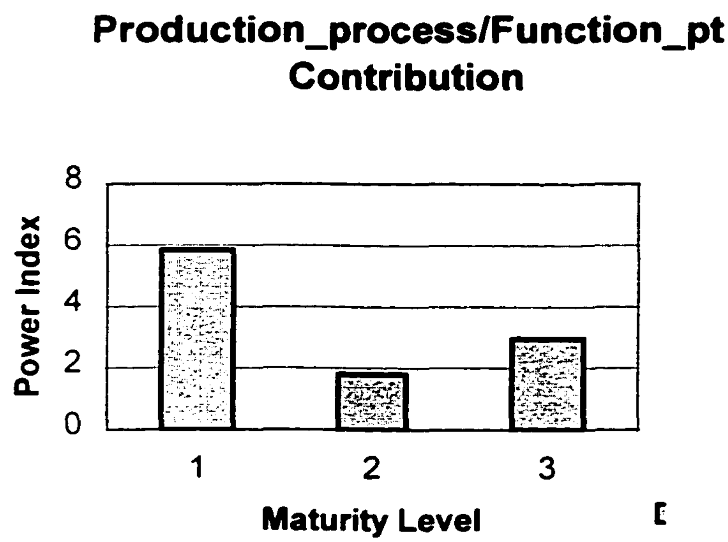
- Function points have a positive, strong effect on Productivity. For example, bigger projects tend to be more efficient for the Productivity. This might suggest that as the project's size increases, the more beneficial this for a higher maturity level.
- Man month has a positive relation with Team size. This explained why the communication overhead does not overcome the contribution from additional staff.
- Duration has a positive though smaller effect on Productivity.
- Team size has a positive and stronger relation to the Management effort.
- Productivity has a positive and stronger relation to the Management effort.

5.8 General Model Comparison

We also compared the Integrated Model behavior among the three levels. In the Integrated Model. the two latent variables we introduced had the following meaning:

- Production process--means all direct influences. new techniques.
- Management Effort--Team size adjustment. schedule pressure. etc.

Now we'll discuss the parameters' differences among the three levels.



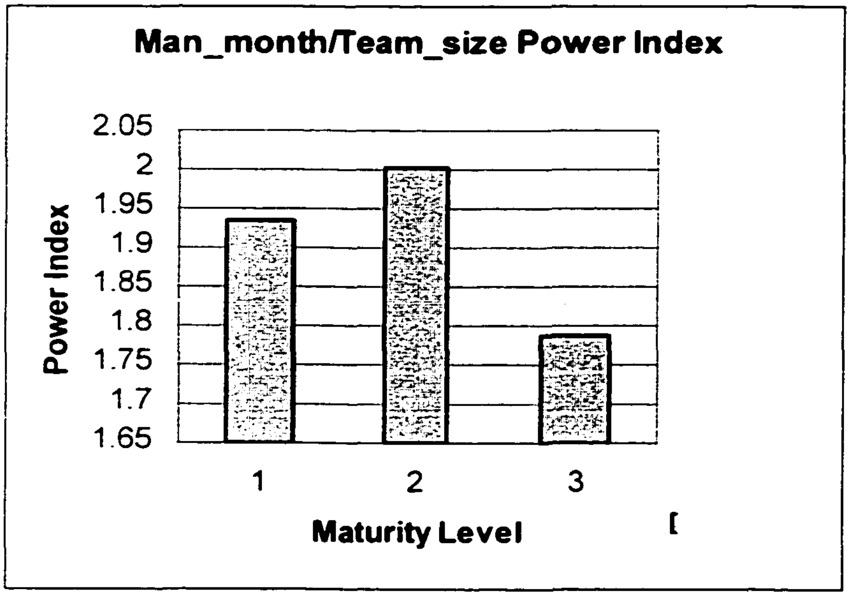
- For the Function points versus Production process (see the preceding figure).

Function_point = $PRODUCT^{5.8509}$ Level I

Function_point = $PRODUCT^{1.8004}$ Level II

Function_point = $PRODUCT^{2.9697}$ Level III

The power indexes of *Production_process* showed strong advantage when advancing from Level II to Level III. As for the Level I environment, management sees significant benefit if it can simply modify the *Production_process*. Any improvement will yield a huge return in a *Production_process* increase.



- For the Man month versus Team size (see preceding figure).

Man_month = Team_size $^{1.9369}$ Level I

Man_month = Team_size $^{2.0022}$ Level II

Man_month = Team_size $^{1.7872}$ Level III

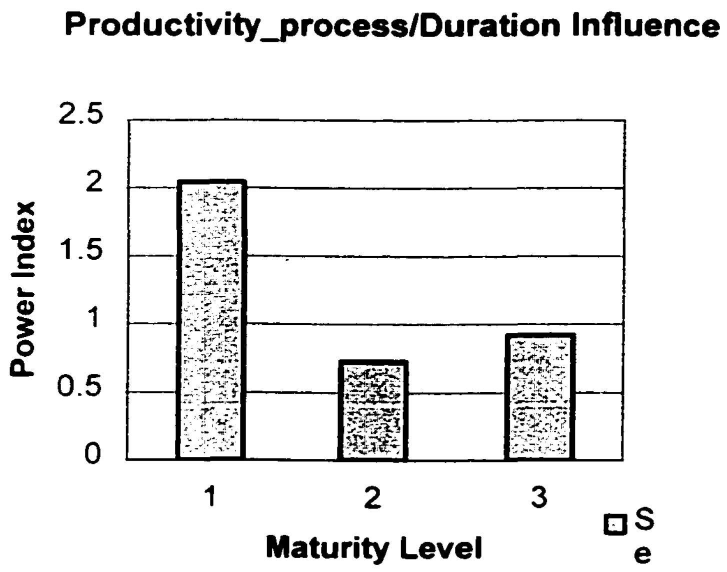
The power indices of the Team size showed the improvement of Level III over Level II for larger *Team_size*. But the overheads of the Team size among all three levels are close. The sustained power index (1.93, 2.00, 1.78 corresponding to level I, II, III) showed a stubborn overhead for the Team size.

- For the Duration versus Productivity process (see the following figure).

$$DURATION = PRODUCT^{2.0450} \quad \text{Level I}$$

$$DURATION = PRODUCT^{0.7234} \quad \text{Level II}$$

$$DURATION = PRODUCT^{0.9210} \quad \text{Level III}$$



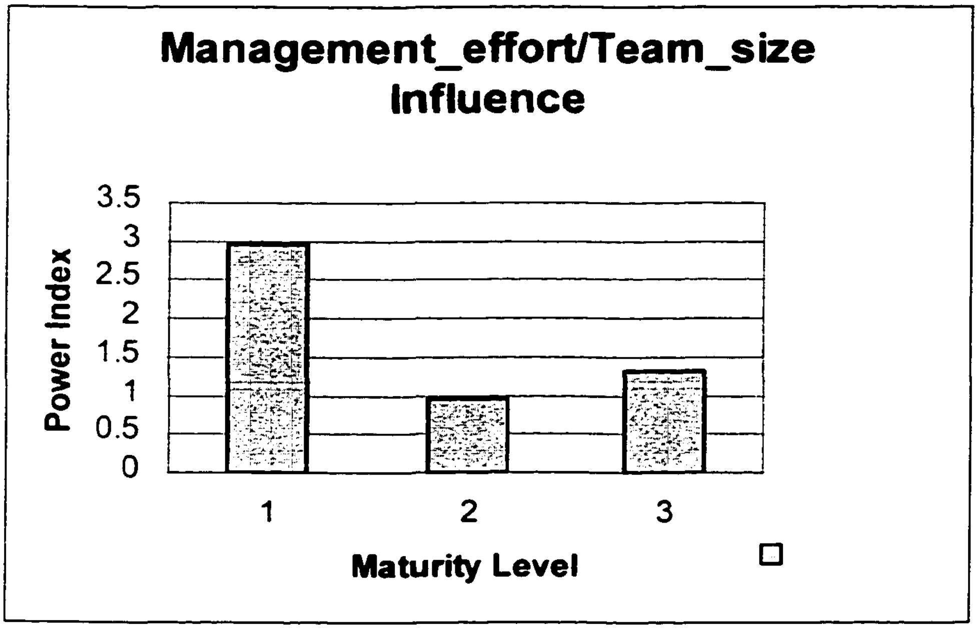
The power indexes of the *Production_process* showed the significant improvement from Level II over Level I. Level II and level III are close. but the Production/Duration indexes suggested some overhead for level III over Level II.

- For the Team size versus Management effort (see following figure).

Team_size = Management $^{2.9636}$ level I

Team_size = Management $^{0.9600}$ level II

Team_size = Management $^{1.3102}$ level III



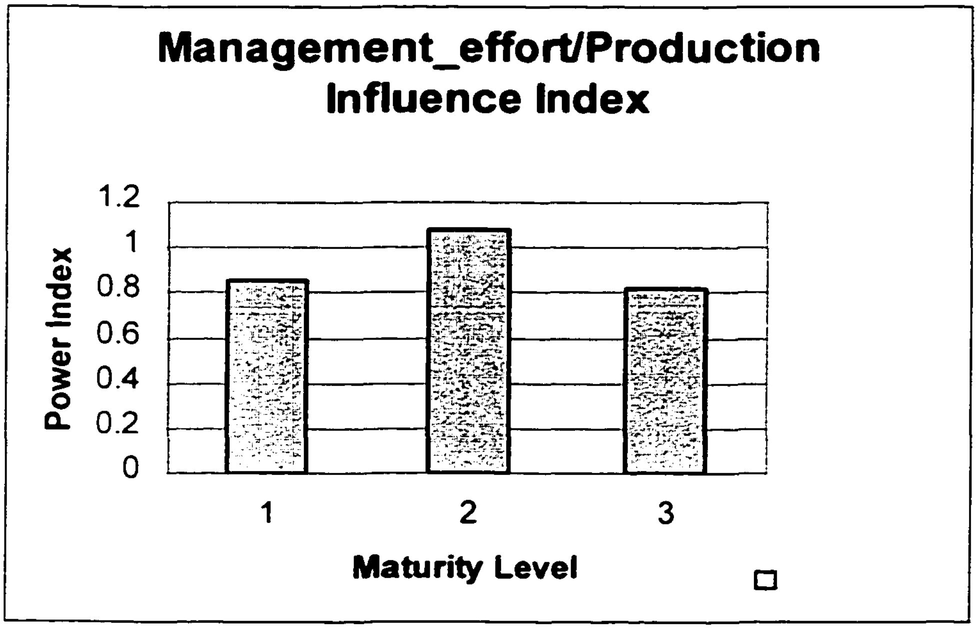
The power indexes of the *Management_effort* demonstrate the diminishing influence from Level II over Level I. Moreover. Level I tended to overreact. Level II and Level III are close. but *Management_effort* influence to *Team_size* showed that Level III had more influence by the power factor of 0.4.

- For the latent variable. Management effort versus Production process (see next figure).

PRODUCT = Management $^{0.8510}$ level I

PRODUCT = Management $^{1.0776}$ level II

PRODUCT = Management $^{0.8196}$ level III



All three levels have similar influence or relationships with respect to *Production_process* versus *Management_effort*.

6 BUILDING A MODIFIED COCOMO MODEL

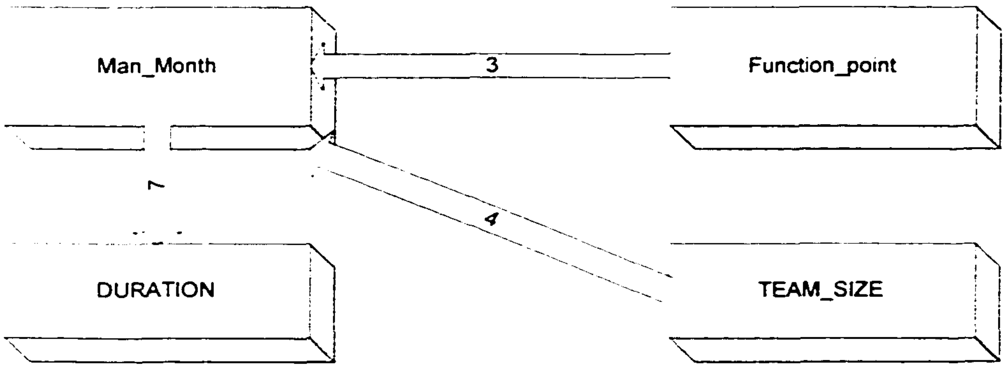


Figure 10 : Modified COCOMO Model

We use the same data we used in the previous model. We discarded 106 Level I records, which are using the *Line_of_code* metric.

Maturity Level	Completed Project in Function_point	Complete Project in Line_of_code	Total Record
I	82	-	82
II	38	54	92
III	27	10	37

The maximum likelihood is used to fit the model.

6.1 Level I Data SAS Run

For the Level I data, the Goodness of Fit Index (GFI) is 0.9473.

LEVEL I RESULTS WITH LOG TRANSFORMATIONS
ESTIMATION OF PARAMETERS

Covariance Structure Analysis: Maximum Likelihood Estimation
Manifest Variable Equations

MAN_MTH =	0.4100*NFP	+ 0.5347*TEAM_SZ	+ 1.0000 E_01
Std Err	0.0313 ALPHA3	0.0995 ALPHA4	
t Value	13.0962	5.3736	

```
DURATION=   0.6336*MAN_MTH + 1.0000 E_02
Std Err     0.0192 ALPHA7
t Value     32.9755
```

6.2 Level I Model

The Modified COCOMO Model for a Level I organization is as follows:

Man_month = Function_point $^{0.41}$ Team_size $^{0.5347}$
Duration = Man_month $^{0.6636}$

If the environment with a fixed *Team_size*, that is, *Team_size* $^{0.5347}$, becomes constant, then it becomes a standard COCOMO model (3).

6.3 Level II Data SAS Run

For the Level II data, the Goodness of Fit Index (GFI) is 0.9021.

LEVEL II RESULTS WITH LOG TRANSFORMATIONS
ESTIMATION OF PARAMETERS

Covariance Structure Analysis: Maximum Likelihood Estimation
Manifest Variable Equations

```
MAN_MTH =   0.1861*NFP     + 1.3898*TEAM_SZ + 1.0000 E_01
  Std Err   0.0386 ALPHA3    0.1164 ALPHA4
  t Value   4.8254           11.9441

DURATION=   0.6026*MAN_MTH + 1.0000 E_02
  Std Err   0.0196 ALPHA7
  t Value   30.6794
```

6.4 Level II Model

The Modified COCOMO Model for Level II organizations is as follows:

Man_month = Function_point $^{0.1861}$ Team_size $^{1.3898}$
Duration = Man_month $^{0.6026}$

6.5 Level III Data SAS Run

For the Level III data, the Goodness of Fit Index (GFI) is 0.9244.

LEVEL THREE RESULTS WITH LOG TRANSFORMATIONS
ESTIMATION OF PARAMETERS
Covariance Structure Analysis: Maximum Likelihood Estimation
Manifest Variable Equations

MAN_MTH =	0.1368*NFP	+ 1.3715*TEAM_SZ	+ 1.0000 E_01
Std Err	0.0393 ALPHA3	0.1150 ALPHA4	
t Value	3.4842	11.9271	

DURATION=	0.5015*MAN_MTH	+ 1.0000 E_02
Std Err	0.0167 ALPHA7	
t Value	30.0303	

6.6 Level III Model

The Modified COCOMO Model for Level III Organization is as follows:

$$\textbf{Man\_month} = \textbf{Function\_point}^{0.1368}\ \textbf{Team\_size}^{1.3715}$$
$$\textbf{Duration} = \textbf{Man\_month}^{0.5015}$$

6.7 General Comparison

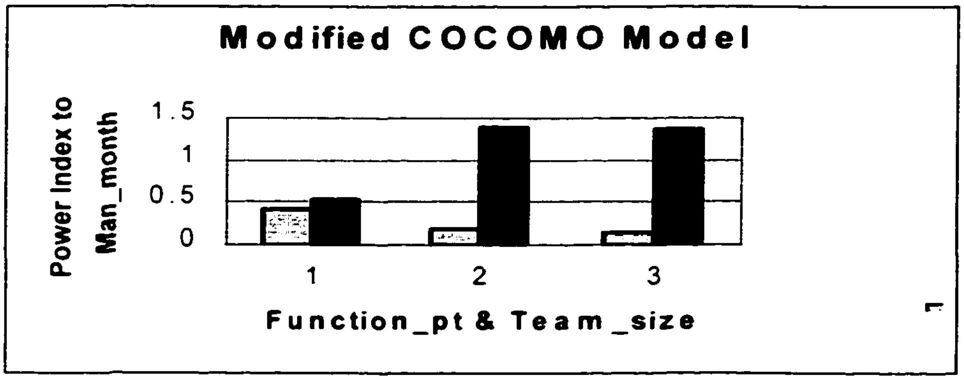
Next we compared the differences among the three Modified COCOMO Models.

- For the *Man_month* estimation function .

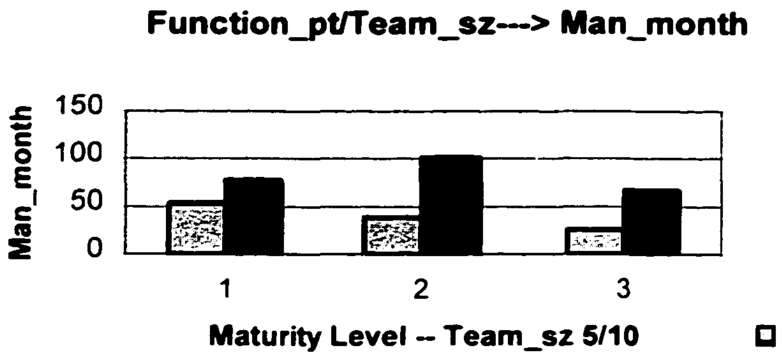
$$Man\_month = Function\_point^{0.4100}\ Team\_size^{0.5347} \quad \text{Level I}$$

$$Man\_month = Function\_point^{0.1861}\ Team\_size^{1.3898} \quad \text{Level II}$$

$$Man\_month = Function\_point^{0.1368}\ Team\_size^{1.3715} \quad \text{Level III}$$



The equations mentioned here suggested that in Level II or Level III. the Team size has the similar but slightly lower overhead to the Man_month. For Level I. although the Team size power index (0.534) is small. but the power index of Function_point is much bigger than the indexes of Level II and III. So. the Function_point influence will eventually over-write the Team_size influence. Now let's illustrate the mutual influence by an example. a project with 2000 *Function_point* and 5/10 *Team_size*. the *Man_month* diagram is as shown in the next page figure:



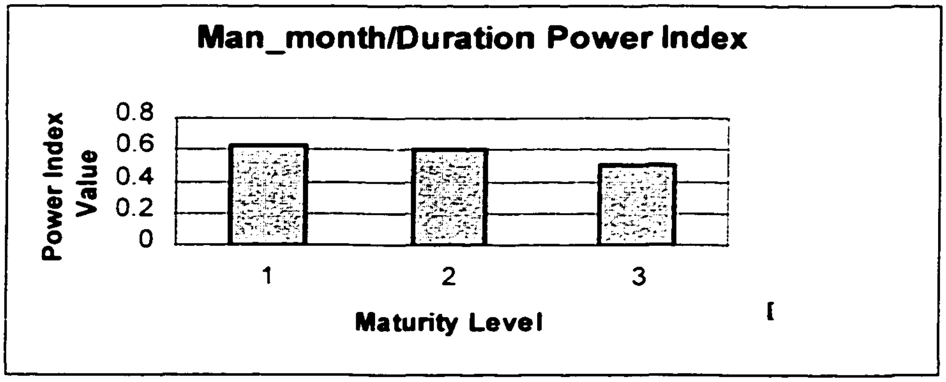
For the *Team_size*=5 (left side of the graphic). the Man month decreased over higher maturity levels. However. for *Team_size*=10(right side of graphic). the Man month increased first. then decreased over the higher maturity level.

- For the Duration estimation (see the following figure).

Duration = $Man\_month^{0.6336}$ Level I

Duration = $Man\_month^{0.6026}$ Level II

Duration = $Man\_month^{0.5015}$ Level III



The equation just given clearly suggests that the more the level advances. the shorter the Duration (schedule) becomes.

7 FACTORS CROSS MODEL COMPARISON

Now we compare the Integrated and Modified COCOMO Models to see if there is any consistency over the higher maturity level.

7.1 *Team_size* Overhead

In the Integrated Model, the *Team_size* overhead is about the same among the three levels, showing only a slight advantage for Level III. This suggests that the *Team_size* overhead will be slightly reduced at a Level III organization. In the Modified COCOMO Model, Level II and Level III had almost the same overhead for the *Team_size*. Level I had lighter overhead for a larger *Team_size*, which might be caused by the nature of sampled data--data were from small enterprises that could run very efficiently (29).

7.2 Duration (Scheduling)

In the Integrated Model, Duration showed significant advantage when advancing from Level I to Level II. The duration of Level III is very similar to the duration of Level II although Level III has a slightly higher overhead. This provides a good explanation for the fact that Level I evolves to Level II—there is a huge Scheduling reduction.

In the Modified COCOMO Model, all three levels had very similar relations to the **Man month**. From the equations for the modified COCOMO model, we get

$$\text{Man\_month} = \text{Function\_point}^{X}\ \text{Team\_size}^{Y}$$
$$\text{Duration} = \text{Man\_month}^{Z}$$

The Z (0.66. 0.60. 0.50) corresponding to Level I. II. and III strongly suggested that the organization with the higher maturity level would do more efficient scheduling.

7.3 Man Month Factor

In the Integrated Model. Man month is regulated by the Team size. The influence showed significant decrease from Level II to Level III. although moving from Level I to Level II slightly increased the overhead.

In the Modified COCOMO Model. the Man month factor showed significant advantage from Level I to Level II. and some advantage from Level II to Level III. with the power index as (0.41. 0.18. 0.13). Now. we'll induce the direct relation between Man month and Function point.

The Integrated Model showed the following:

Function_point = Production_process ** C2 .....(1)

Duration = Production_process ** C1 .....(2)

Man_month = Team_size ** C4 .....(3)

Team_size = Management_effort ** C5 .....(4)

Production_process = Management_effort ** C6 ....(5)

where C1. C2. C4. C5. and C6 are parameters in the Integrated Model. From (1) and (5). we get. Function_point = Management_effort ** (C2 * C6)

From (3) and (4).

Man_month = Management_effort ** (C4 * C5)

Combine the two above. and we get

Man_month = Function_point ** ((C4 * C5)/(C2 * C6))

Recalculate the Integrated Model for the Man_month versus Function Index as follows:

Maturity Level	C1	C2	C4	C5	C6	**MM:FP Index**	***Modified COCOMO***
I	2.045	5.8509	1.9369	2.9636	0.851	**1.1529**	***0.41***
II	0.7234	1.8004	2.0022	0.96	1.0776	**0.9907**	***0.18***
III	0.921	2.9697	1.7872	1.3102	0.8196	**0.9620**	***0.13***

The last two columns showed the huge drop from Level I to Level II and slight advantage from Level II to Level III.

7.4 *Management_effort* versus *Production_process*

From the data we studied. the relations are about 1:1 among all three levels. and they are only available in the Integrated Model. This might support Mr. Capers Jones observation that

"Small Enterprises (the sampled software projects in this research) can run very efficiently without much software engineering effort" (29).

7.5 Summary : Advantages of Higher Maturity Level

The advantages of advancing maturity levels can be summarized as follows:

- The scheduling (Duration) showed significant improvement from Level II versus Level I. Although the Integrated Model showed slightly more overhead in Level III. Level III had a significant *Production_process*/Function points increase.
- For *Production_process* versus Function points. Level III outperformed Level II by almost an exponential of 1 (1.8 versus 2.9). This suggested strong

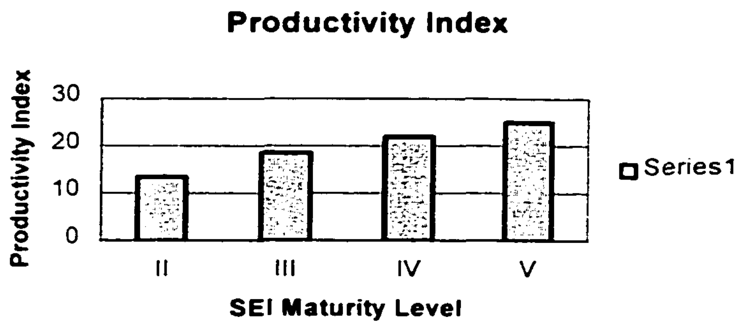
productivity increases if Maturity Level II advances to Level III.

- *Man_month* reduction. Viewed from the modified COCOMO Model.

Man_month = Function_point ^{X} Team_size ^{Y}
Duration = Man_month ^{Z}

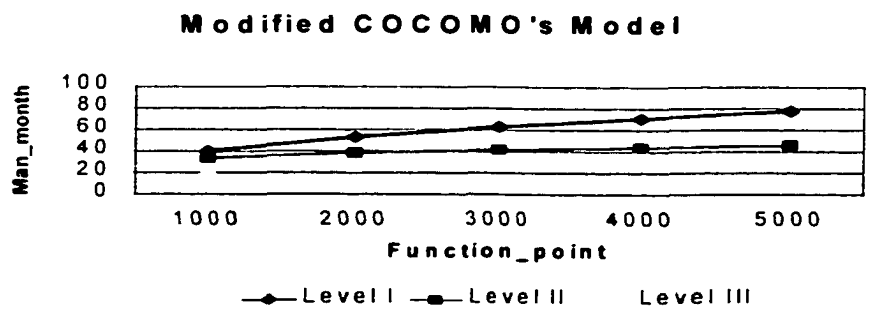
The X values were 0.41, 0.18, and 0.13, which corresponded to Level I, II, and III, respectively. This strongly suggested that the higher the maturity level, the better productivity.

As the following figure shows, Lawrence H. Putnam found that there was a relation between SEI level and Productivity Index and SEI Maturity Level (36). The PI (productivity index) value at which the transaction occurs (13.5, 18.5, 22.0, 25.0)



corresponds to Maturity Level II to Level V.

There was some similarity in our Modified COCOMO Mode. For example, when *Team_size* = 5, the model provides the similar trend.



8. DYNAMIC MODEL

8.1 Reasoning for Dynamic Model

Abdel-Hamid Tare's massive work has introduced a Dynamic Model (1). In his integrated approach, all elements will influence each other in a circle (40). This approach will provide a more realistic environment as time passes and will provide management with a chance to simulate the process through different strategies (40).

8.2 Dynamic Model Introduction

We used the **ithink™** product and its symbolic language to construct the model (25). There were five basic elements: *stocks, flows, clouds, converters, and feedbackloop.* They were defined as follows:

- "Stock"--represents accumulations: it is represented by a rectangular box.
- "Flows"—these fill up the stocks and are represented by an arrow. The "Work Rate" is used to control the flow.
- "Cloud"--means unlimited supply, or the boundaries of the model.
- "Converter"--converters are used to elaborate the details of stock and flow structure. Their symbol is a circle with one or more line emanating from it.
- "Feedbackloop"--stepwise via the period of time. In our model, *management_effort* is the starting point, and the cycle ended at *management_effort.*

Based on the Integrated Model shown in Figure 9. we'll select the elements as follows:

Stock: Work_completed(Function_point)

Converter: Management_effort. Production_process. Duration. Function_point. Team_size. Estimated Schedule. and another six introduced (calculated) converters:

- Monthly_Production (work completed: calculated from Function_point and Duration)
- Schedule_deliverable: calculated from the Remaining_Work_load / Monthly_Production
- Deadline (input variable)
- Total_Function_point (input variable)
- Schedule_pressure: determined by comparing Estimated_schedule and Deadline
- Project Status: determined by comparing Schedule_deliverable and Schedule_pressure

The Integrated Dynamic Model is shown in Figure 11. next page.

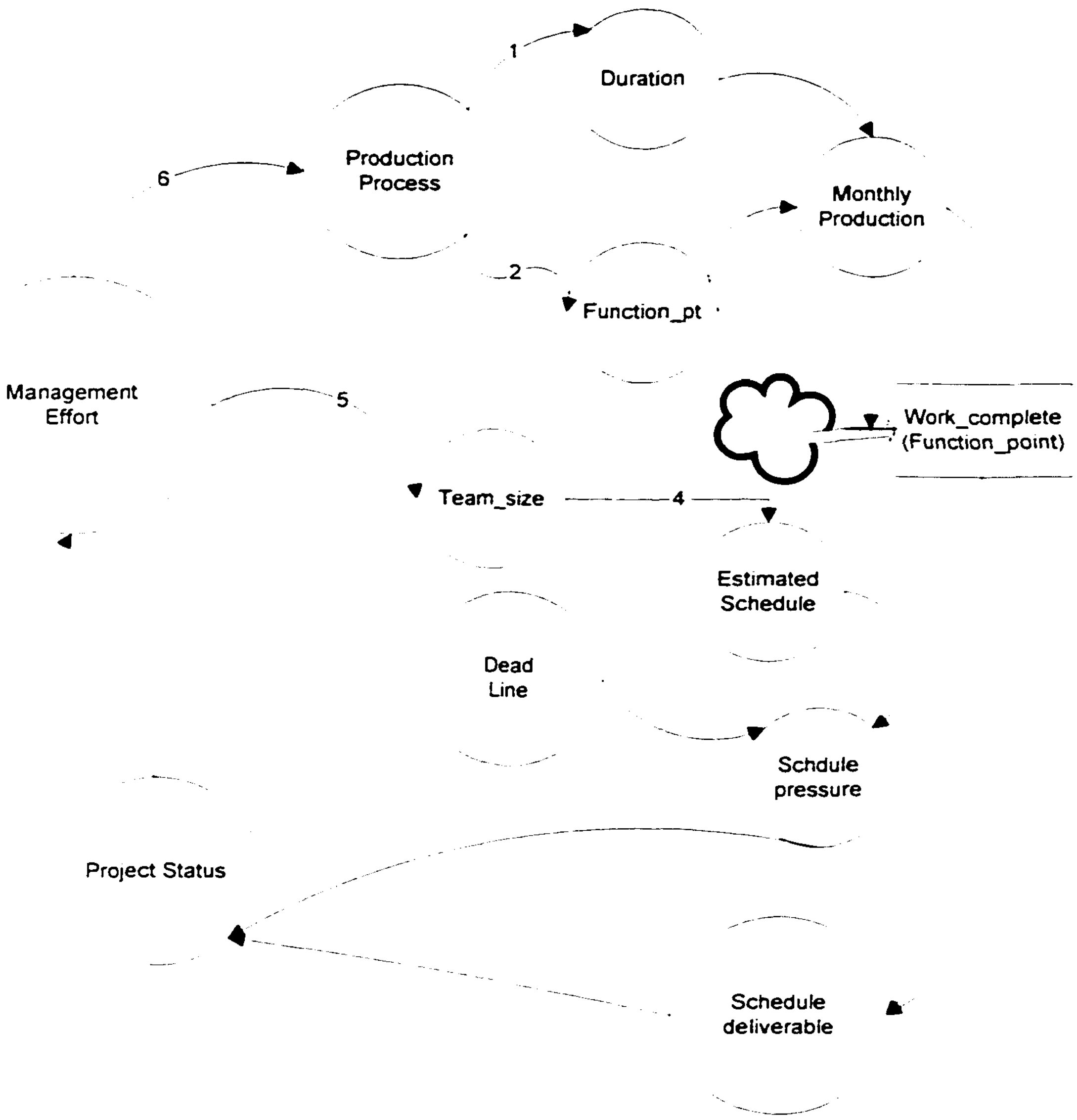


Figure 11 : Integrated Dynamic Model

8.3 Simulation Program

```
Control Variables : /* input Variables */
Total_Function_Point.
Deadline (Total_month_planned)
Team_size = Initial_size
Alpha1.Alpha2...Alpha6 /* level 1. level 2. level 3 correspondingly */
Management_effort_Upper_limit. Management_effort_Lower_limit. /* review in 8.5 */
Management_effort_delta =
 (Management_effort_Upper_limit – Management_effort_Lower_limit) / Deadline
/* Initial control variables */
Management_Effort = Team_size ** (1/Alpha5)
Work_complete = 0 /* Function_point */
Month_counter = 0
/* Management_increase is based on
/* LOOP */
Loop :
Month_counter = Month_counter + 1
   /* Part 1. Real Schedule under current productivity */
   Production_Process = Management_effort * Alpha6
  Team_size = (Management_effort * Alpha5)
  Monthly_Production = (Production_process * Alpha2) /
                             (Production_process * Alpha1)
  Work_complete = Work_complete + Monthly_Production
IF (Work_complete not< Total_Function_point) GOTO FINISH :
   Schedule_deliverable = (Total_Function_Point – Work_complete) /
Monthly_Production
  /* Part 2. Management_Schedule */
   Estimated_schedule = Team_size ** (Alpha4) / Team_size
   Schedule_pressure = 1/2 * (Dead_line + Estimated_schedule) – Month_counter
  /* Decision for next Management Move */
   IF (Schedule_pressure > Schedule_deliverable)
  /* Add Management_Effort to Speed up */
    THEN Management_effort = Management_effort + Management_effort_delta
            IF (Management_effort > Management_effort_Upper_limit)
             THEN Management_effort = Management_effort_Upper_limit :
    ELSE    IF (Schedule_pressure < Schedule_deliverable)
            Management_effort = Management_effort - Management_effort_delta
               IF (Management_effort < Management_effort_Lower_limit)
               THEN Management_effort = Management_effort_Lower_limit :
PRINT
(Month_coun.Management_effor:.Team_size:Monthly_Production:Estimated_schedule)
GOTO Loop:
Finish :
```

8.4 Simulation Control Variables

- *Management_effort* is the only pivotal variable for the whole cycle. It is estimated from Team_size +/- 2 * standard_deviation (this would cover 95% of the range).

	Sample Size	Mean	Standard Deviation	Lower Bound	Upper Bound
Level I	82	10	17	1	44
Level II	92	14	17	1	48
Level III	37	23	28	1	79

--The Management_effort_Upper-limit = Team_size_Upper_Bound ** (1/Alpha5)

--Management_effort_Lower_limit = Team_size_Lower_Bound ** (1/Alpha5)

- The *dead_line* would also be a key variable for the simulation. There is a very good likelihood that the simulation will answer the following question: "Different Maturity Levels will make how much of a difference in Time Pressure?"
- *Initial Team_size*. This variable might tell people "Is it a critical variable for the process?"

The following table shows the simulation constants.

	Level I	Level II	Level III
Alpha1	2.045	0.7234	0.921
Alpha2	5.8509	1.8004	2.9697
Alpha4	1.9369	2.0022	1.7872
Alpha5	2.9636	0.96	1.3102
Alpha6	0.851	1.0776	0.8196

9. CONCLUSION

The Team size in our data varies from ten to sixty. According to Capers Jones's observation (29), this size of sampled data indicates small enterprises. Small enterprises can run very efficiently without too much software engineering effort. Thus, "to put in an effort to advance to a higher maturity level" may not be that crucial for them. This could be the reason why our model did not show significant advantages from Level III over Level II.

The practice of COCOMO/COCOMO II started with a fairly refined questionnaire for fitting/classifying individual cases based on its specific model for choosing multipliers and scale factors. This practice is viewing the software process from *micro* perspectives. The SEI maturity framework is viewing from the *macro* perspective–that is, maturity level is only grouped from certain common behaviors. This purpose of our research is to try to find some common denominators among the different levels. Although the sampled data is from small enterprises, *it still proves that it is worthwhile to advance to a higher maturity level.* The advantages are significant.

The model built in this study has omitted a very important factor--*cost*. Dr. Howard Rubin's World Benchmark Projects has a cost factor associated with each project. It would be a very challenging model if the cost factor could be included. "Cost Effective Integrated Model" would be a valuable model to simulate Cost/Productivity/Scheduling "Dynamic Behavior." It might provide the final answer whether it is **worthwhile to advance to a higher maturity level by knowing the cost of the additional effort.**

The pros and cons discussed in our model are based on productivity and scheduling only (defect and quality are not discussed). The major focus/control of maturity Level IV and V is quality. Will there be a consistent model for covering the defect of all the levels (or if the defect is insignificant for the lower maturity levels, such as II or III)? This will be another challenge.

Capers Jones has observed that "Some maturity level III Navy organizations were found to be deficient in terms of software quality" (29). If there is such a Dynamic/Integrated Model by way of simulation, perhaps it would provide an answer for the Navy's case.

The aggregated behaviors of different maturity levels have their value/advantages for illustrating software development processes. This research as viewed from the macro perspective will hopefully provide an incentive to other researchers to build models from the same perspective.

```
/* SAS Input to Build Integrated Model=;all Level  Run */
OPTIONS LS= 75;
CMS FILEDEF S1 DISK L1 TEXT A;

/**************************/
/* INPUT DATA FROM A DISK */
/**************************/

DATA A;INFILE S1 MISSOVER ;INPUT
ORIG  NFP MAN_MTH DURATION TEAM_SZ ;

/* ***************************************************/
/* SAS MACRO 'CL' PERFORM CLUSTER ANALYSIS CLUSTER=4 */
/* ***************************************************/

%MACRO CL;
TITLE ' CLUSTER ANALYSIS ';
PROC CLUSTER METHOD=AVE OUTTREE=TREE DATA=A NOPRINT;
VAR  NFP MAN_MTH DURATION TEAM_SZ;
PROC TREE DATA=TREE NCL=2 OUT=P NOPRINT;
COPY  NFP MAN_MTH DURATION TEAM_SZ;
PROC SORT DATA=P;BY CLUSTER;

PROC MEANS NOPRINT DATA=P;
VAR  NFP MAN_MTH DURATION TEAM_SZ;
OUTPUT OUT=PP MEAN=; BY CLUSTER;
PROC PRINT DATA=PP;
%MEND;

/*********************************************/
/* CALCULATE PEARSON CORRELATION COEFFICIENTS */
/*       WITH ORIGINAL SCORES                 */
/*********************************************/

TITLE 'CORRELATION COEFFICIENTS AMONG THE FOUR ORIGINAL VARIABLES';RUN;
PROC CORR ; VAR NFP MAN_MTH DURATION TEAM_SZ;

/*********************************************/
/* MAKING NATURE LOG TRANSFORMATIONS          */
/*********************************************/

DATA A;SET A;
MAN_MTH=LOG(MAN_MTH);NFP=LOG(NFP);
TEAM_SZ=LOG(TEAM_SZ);DURATION=LOG(DURATION);

/*********************************************/
/* CALCULATE PEARSON CORRELATION COEFFICIENTS */
/*       WITH NATURAL LOG TRANSFORMATION      */
/*********************************************/
```

```
PROC CORR ; VAR NFP MAN_MTH DURATION TEAM_SZ;
TITLE 'CORRELATION AMONG THE TRANSFORMED VARIABLES';RUN;

/*********************************************/
/* RUN CLUSTER ANALYSIS - STARTS WITH 4 GROUPS */
/*********************************************/

%CL;
TITLE1 'LEVEL ONE RESULTS WITH LOG TRANSFORMATIONS '; RUN;

/*********************************************/
/* RUN SYSTEM REGRESSION WITH ALL DATA AS ONE  */
/*********************************************/

DATA A1(KEEP=CLUSTER CODE INTERCEP);SET P;
IF CLUSTER=1 THEN CODE=0;ELSE IF CLUSTER=2 THEN CODE=1; ELSE
CODE=.;
IF CODE NE .;
INTERCEP=1;
DATA A;SET A;SET A1;

TITLE2 'ESTIMATION OF PARAMETERS';RUN;
PROC CALIS UCOV DATA=A MAXITER=50000 MAXFU=50000
OMETHOD=NR  SE GTOL=0.001 METHOD=ML;
LINEQS
DURATION= ALPHA1 (0.5)   FPRODUCT                        + E_01,
NFP      = ALPHA2 (0.5)   FPRODUCT  + ALPHA3     CODE + E_02,
MAN_MTH = ALPHA4          TEAM_SZ                        + E_03,
TEAM_SZ = ALPHA5 (0.5)   FMGMT                           + E_04,
FPRODUCT= ALPHA6 (1.0)   FMGMT                           + E_05;
      STD
          E_01-E_05 = ESP1-ESP5 (5*3.0),
          FMGMT=PHI (1.0);
      COV
          E_03 E_04 = ESP2(.2);
      BOUNDS
          ALPHA6 > 0;


*         E_01 E_04 = ESP1(.2),

*%MODEL1;

/*
%MACRO MODEL1;

TITLE2 'ESTIMATION OF PARAMETERS';RUN;
PROC CALIS UCOV DATA=A MAXITER=500000 MAXFU=500000
```

```
OMETHOD=NR  METHOD=LSGLS
EDF=74 SE;
LINEQS

DURATION=  1.0   FPRODUCT                    + E_01,
NFP      =  1.0   FPRODUCT   + ALPHA3 CODE  + E_02,
MAN_MTH =  1.0   TEAM_SZ                     + E_03,
TEAM_SZ =  1.0   FMGMT                       + E_04,
FPRODUCT=  ALPHA1 (0.5) FMGMT                + E_05;

       STD
           E_01-E_05 = ESP1-ESP5 (5*3.0),
           FMGMT=PHI (0.5);
       COV
           E_01 E_04 = ESP1(.2),
           E_03 E_04 = ESP2(.2);
%MEND;

/*
DURATION=  1.0   FPRODUCT                    + E_01,
NFP      =  1.0   FPRODUCT   + ALPHA3 CODE  + E_02,
MAN_MTH =  1.0   TEAM_SZ     + ALPHA4 CODE  + E_03,
TEAM_SZ =  1.0   FMGMT       + ALPHA5 CODE  + E_04,
FPRODUCT=  ALPHA1 (0.5) FMGMT                + E_05;
*/
```

```
/* SAS Input for Modified COCOMO MODEL   */
OPTIONS LS= 75;
CMS FILEDEF S1 DISK L1 TEXT A;
 /*************************/
 /* INPUT DATA FROM A DISK */
 /*************************/

DATA A;INFILE S1 MISSOVER ;INPUT
ORIG  NFP MAN_MTH DURATION TEAM_SZ ;

/* ***************************************************/
/* SAS MACRO 'CL' PERFORM CLUSTER ANALYSIS CLUSTER=4 */
/* ***************************************************/

%MACRO CL;
TITLE ' CLUSTER ANALYSIS ';
PROC CLUSTER METHOD=AVE OUTTREE=TREE DATA=A NOPRINT;
VAR  NFP MAN_MTH DURATION TEAM_SZ;
PROC TREE DATA=TREE NCL=2 OUT=P NOPRINT;
COPY  NFP MAN_MTH DURATION TEAM_SZ;
PROC SORT DATA=P;BY CLUSTER;

PROC MEANS NOPRINT DATA=P;
VAR  NFP MAN_MTH DURATION TEAM_SZ;
OUTPUT OUT=PP MEAN=; BY CLUSTER;
PROC PRINT DATA=PP;
%MEND;

/*********************************************/
/* CALCULATE PEARSON CORRELATION COEFFICIENTS */
/*      WITH ORIGINAL SCORES                  */
/*********************************************/

TITLE ' CORRELATION COEFFICIENTS AMONG THE FOUR ORIGINAL
        VARIABLES';RUN;

PROC CORR ; VAR NFP MAN_MTH DURATION TEAM_SZ;

/*********************************************/
/* MAKING NATURE LOG TRANSFORMATIONS          */
/*********************************************/


DATA A;SET A;
MAN_MTH=LOG(MAN_MTH);NFP=LOG(NFP);
TEAM_SZ=LOG(TEAM_SZ);DURATION=LOG(DURATION);
/*********************************************/
/* CALCULATE PEARSON CORRELATION COEFFICIENTS */
/*      WITH NATURAL LOG TRANSFORMATION       */
```

```
/**********************************************/

PROC CORR ; VAR NFP MAN_MTH DURATION TEAM_SZ;
TITLE 'CORRELATION AMONG THE TRANSFORMED VARIABLES';RUN;



/***********************************************/
/* RUN CLUSTER ANALYSIS - STARTS WITH 4 GROUPS */
/***********************************************/

%CL;

TITLE1 'LEVEL RESULTS WITH LOG TRANSFORMATIONS '; RUN;

/***********************************************/
/* RUN SYSTEM REGRESSION WITH ALL DATA AS ONE  */
/***********************************************/

DATA A1(KEEP=CLUSTER CODE INTERCEP);SET P;

IF CLUSTER=1 THEN CODE=0;ELSE IF CLUSTER=2 THEN CODE=1; ELSE
               CODE=.;
IF CODE NE .;
INTERCEP=1;

DATA A;SET A;SET A1;

TITLE2 'ESTIMATION OF PARAMETERS';RUN;

PROC CALIS UCOV DATA=A MAXITER=500 MAXFU=1250 METHOD=ML
OUTEST=P1 OUTSTAT=P2;
LINEQS
MAN_MTH = /*ALPHA1   INTERCEP+ ALPHA2      CODE +  */
         ALPHA3       NFP +  ALPHA4   TEAM_SZ + E_01,
DURATION= /*ALPHA5   INTERCEP+ ALPHA6      CODE +  */
         ALPHA7   MAN_MTH +   E_02;
      STD
          E_01-E_02 = ESP1-ESP2;
      COV
                     E_01 E_02 = ESP3;
          RUN;
```

The Survey Data: Level I

0_origin	FP	man_mth	duration	team-sz	yr-complet
0	925	26.9	5	3	1995
0	4990	67.2	11.5	8	1995
0	3000	316	30	14	1997
0	1600	51	3.8	12	1994
0	400	31	4.4	8	1994
0	450	10	4.6	5	1994
0	200	17	3	6	1993
0	300	1.5	1.4	3	1994
0	3854	26	11	6	1993
0	400	20	4	6	1994
0	200	4	2	3	1994
0	9312	1234	26	70	1986
0	1570	60.78	20	4	1993
0	937	25.26	19	4	1993
0	1939	86.84	14	15	1992
0	818	37.1	7	9	1993
0	693	29.21	12	2	1993
0	3238	106	9	12	1994
0	630	14	17	2	1994
0	617	8	7	2	1993
0	6839	257	25	11	1994
0	1099	77.4	18	11	1993
0	2571	214.9	21	10	1993
0	350	38.5	12	4	1994
0	121	7.2	8	1	1994
0	367	103.3	14	9	1994
0	2831	39.25	18	7	1988
0	1176	65.75	26	8	1990
0	4128	121.6	10	6	1991
0	2348	91.8	13	7	1991
0	3470	102	26	9	1990
0	138	2	4	1	1994
0	66	0.75	2	1	1994
0	188	11.88	7	3	1994
0	91	24	7	5	1994
0	60	78	15	6	1994
0	125	98	13	9	1995
0	500	54	24	3	1995
0	23684	200	18	10	1994
0	129	4.8	16.5	54	1994
0	192	21.7	15.5	47	1994
0	63	15.8	9	54	1994
0	580	693.4	18	120	1994
0	779	8.9	4	2.5	1996
0	1697	27	17	1.5	1996
0	1060	17.9	14	1.3	1996

0	770	16	14	1.2	1996
0	1147	97	9	15	1993
0	700	200	13	21	1993
0	346	10.07	9.87	5	1994
0	4494	40.58	16.4	9	1994
0	2449	33.22	12.7	7	1994
0	348	51.59	19	5.5	1992
0	961	90	14	13	1994
0	1489	257	55	6	1994
0	507	69	18	4.25	1994
0	692	109	22	8	1995
0	447	38	10	7	1995
0	1384	120	22	7	1995
0	596	42.9	16	5	1995
0	25	11.6	12	2	1995
0	350	38.5	12	4	1994
0	121	7.2	8	1	1994
0	1842	146	13	14	1996
0	616	23	8	4	1996
0	682	8.6	3	5	1996
0	35	12.5	12	2.8	1996
0	36	7.5	8	5	1996
0	50	9.5	9	4	1996
0	316	24	12	4	1998
0	535	27	4	8	1996
0	408	38	3	10	1996
0	186	23	3	6	1996
0	697	111	13	13	1996
0	602	47	13	9	1996
0	853	88	7	13	1996
0	125	18	9	7	1996
0	101	13	12	5	1996
0	110	6	3	3	1996
0	70	3	1	5	1997
0	738	69	18	4	1997
0	251	25.3	7	5	1997

The Survey Data: Level II

0_origin	FP	man_mth	duration	team-sz	yr-complet
0	212	14.8	21.45	2	1993
0	236	20.9	27.85	2	1994
0	176	48.75	26.1	8	1994
0	1217	70.85	7.6	15	1994
0	290	46.25	25.05	7	1994
0	212	296	429	5	1993
0	236	418	557	4	1994
0	176	975	522	5	1994
0	1217	1417	152	17	1994
0	290	925	501	5	1994
0	1411	94	14	10	1994
0	569	7	4	3	1994
0	644	20	8	4	1994
0	206	10	4	3	1994
0	6083	94	17	10	1994
0	4769	169	28	7	1994
0	4763	93	12	7	1994
0	2983	53	22	4	1994
0	2250	76	7	7	1994
0	1000	18	3	10	1994
0	100	0.8	2	1	1992
0	2117	17.7	6	2.5	1993
0	1680	95	14	6	1994
0	2788	151	16.6	10	1994
0	4300	659	28.8	34	1994
0	12000	2163	30	60	1995
0	6000	890	40	41	1995
0	6700	467	31	28	1995
0	2100	80	18	4	1995
0	98	6	4	2	1997
0	1389	131	17	13	1997
0	2,342	147	12	11.6	1996
0	2,500	1200	12	80	1997
0	3,000	450	18	40	1997
0	6000	24	6	4	1997
0	57	24	10	4	1996
0	57	24	10	4	1996
1	51.02	3	6	2	1993
1	18.74	3	6	2	1994
1	3203.13	727	22	50	1995
1	3437.5	636	24	48	1995
1	65.42	14	8	3	1994
1	37.38	1	1	2	1993
1	62.5	43	14	9	1994
1	101.56	30	6	5	1994
1	1326.53	2200	48	50	1995

1	39.06	72	18	5	1994
1	93.75	90	14	8	1995
1	169693.9	288	36	8	1995
1	302000	684	36	18	1995
1	13120.9	54	6	13	1994
1	10909.09	384	46	48	1995
1	510.2	51	9	4	1996
1	58.11	19	12.2	2	1996
1	17800	28	15	3	1996
1	357.82	66	12	3.4	1996
1	192.53	13.5	8.5	2	1996
1	486.95	13	14	0.9	1996
1	224.49	42	13	4	1996
1	2343.75	60	12	6	1993
1	22.45	345.6	48	85	1998
1	1272.73	150	10	15	1995
1	625	20	5	5	1995
1	507.81	102	13	20	1996
1	468.75	62	15	8	1996
1	54.69	60	9	3	1996
1	156.25	3	6	3	1997
1	234.38	24	6	4	1996
1	367.19	180	7	45	1997
1	218.75	117	6	35	1997
1	781.25	150	18	10	1997
1	1326.53	300	24	18	1997
1	781.25	100	17	15	1997
1	7755.1	600	38	28	1997
1	2000	280	26	12	1997
1	1745.46	180	12	18	1997
1	35000	14	12	2	1997
1	8181.82	25	12	30	1997
1	727.27	2	6	3	1997
1	78.13	1	4	2	1997
1	12500	120	30	10	1998
1	437.5	68	14	14	1997
1	195.31	23	12	12	1997
1	818.18	3	7	3	.
1	117.19	1	6	2	.
1	750	48	12	6	1998
1	35	14	12	2	1997
1	781.25	150	18	10	1997
1	1326.53	300	24	18	1997
1	781.25	100	17	15	1997
1	7755.1	600	38	28	1997
1	2000	280	26	12	1997

The Survey Data: Level III

0-orgin	FP	Mam_mth	duration	team-sz	yr-complet
0	9500	840	23	105	1994
0	1420	51	15	5.5	1994
0	1180	49.5	11	6.5	1994
0	900	40.5	10	5	1994
0	450	19	5.5	5	1994
0	450	19	5.5	5	1994
0	192	7.7	7.5	4	1994
0	497	18.7	7	3	1994
0	1267	54	23	6	1994
0	1300	50	17	13	1994
0	12000	800	15	50	1993
0	3000	75	7	5	1994
0	6000	460	10	20	1993
0	1100	150	13	15	1994
0	1500	150	12	18	1997
0	1800	180	12	22	1997
0	10000	1200	18	110	1997
0	3500	183	12	20	1997
0	3018	72	15	10	1997
0	707	30	5	12	1997
0	530	20	4	6	1997
0	4500	311	20	19	1996
0	171	14.8	4	5	1997
0	681	79	10	14	1997
0	486	24	5	14	1997
0	296	29	7	14	1997
1	6250	900	15	80	1995
1	4822511.7	1796	40	60	1997
1	70290.53	6	2	3	1996
1	11246.48	4	2	2	1996
1	163.27	14	10	2	1997
1	816.33	108	27	6	1997
1	112500	1548	36	36	2000
1	5527.27	1100	32	36	1999
1	897.96	225	13	20	1998
1	14054.55	2250	36	75	1999
1	4127.27	75	9	10	1998

REFERENCES

1. Abdel-Hamid. Tarek. and Stuart E. Madnick. "Software Project Dynamics. An Integrated Approach." READING (Prentice-Hall 1991).
2. Albrecht. Allan J.. and John E. Gaffney Jr.. "Software Function. Source Lines of Code. and Development Effort Prediction: A Software Science Validation." IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. November 1983. PP. 639-648
3. Boehm. Barry W.. "Software Engineering Economics." READING (Prentice-Hall 1981).
4. Boehm. Barry W.. "Software Engineering Economics" IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. January 1984. PP.4-21
5. Boehm. Barry W.. "Improving Software Productivity." COMPUTER. September 1987.
6. Boehm. Barry W.. "A Spiral Model of Software Development and Enhancement." IEEE COMPUTER. May 1988. PP.61-72
7. Boehm. Barry W.. "Software Risk Management : Principles & Practices." IEEE SOFTWARE. January 1991.
8. Boehm. Barry W. and Bradford K. Clark. "An Overview of the COCOMO 2.0 Software Cost Model." Software Technology Conference. April 1995.
9. Boehm. Barry W.. "Anchoring the Software Process." IEEE SOFTWARE. July 1996.
10. Boehm. Barry W.,Bradford Clark. Ellis Horowitz. and Chris Westland. "Cost Model for Future Life Cycle Processes – COCOMO 2.0." ANNALS OF SOFTWARE ENGINEERING.

11. Boehm. Barry W..Bradford Clark. Ellis Horowitz.Chris Westland. "The COCOMO 2.0 Software Cost Estimation Model." International Society of Parametric Analysts. May 1995.

12. Box. George E.P.. and Gwilym M. Jenkins. "Time Series Analysis. Forecasting and Control." READING (Holden-Day 1976).

13. Brooks. Frederick P.. Jr.. "The Mythical Man-Month. Essays on Software Engineering." READING: Addison-Wesley 1975.

14. Curtis. Bill. Sylvia B. Sheppard. Phill Milliman. M.A. Borst. and Tom Love. "Measuring and Psychological Complexity of Software Maintenance Tasks with the Halstead and McCabe Metrics." IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. March 1979.

15. Curtis. Bill. William E. Hefley. Sally Miller. Michael Konrad. and Sandra Bond. "Increasing Software Talent." AMERICAN PROGRAMMER. December 1994.

16. DeMarco. Tom. "Controlling Software Projects." READING (Yourdon Press. Prentice-Hall PTR. 1982).

17. Hansen. George A.. "Simulation Software Development Processed." COMPUTER. January 1996. PP.73-77

18. Herbsleb. J.. David Zubrow. Jane Siegel. James Roum. and Anita Carleton. "Software Process Improvement. State of the Payoff." AMERICAN PROGRAMMER. September 1994. PP.2-11

19. Hosmer. David W.. and Lemeshow. "Applied Logistic Regression." READING (Wiley. New York. 1989).

20. Humphrey. W. S.. and W. L. Sweet. "A Method for Assessing the Software Engineering Capability of Contractors." CMU/SEI-87-TR-23. September 1987.

21. Humphrey. Watts S.. "Characterizing the Software Process : A Maturity Framework." IEEE SOFTWARE. March 1988. PP.73-79

22. Humphrey. Watts S.. "Managing the Software Process" READING (Addison-Wesley 1989).

23. Humphrey. Watts S.. "A Personal Commitment to Software Quality." AMERICAN PROGRAMMER. December 1994. PP.3-11

24. Humphrey. Watts S.. "A Discipline for Software Engineering." READING (Addison-Wesley 1995).

25. **ithink™** Users Manual. High Performance Systems. Hanover. NH. 1992.

26. Loehlin. John C.. "Latent Variable Models." READING (Lawrence Erlbaum Associates. Inc. 1998).

27. Jensen. Randall W.. and Charles C. Tonies. "Software Engineering." READING (Prentice-Hall 1979).

28. Jones. Capers. "Assessment and Control of Software Risks." READING (Yourdon Press. PTR Prentice Hall. 1994).

29. Jones. Capers. "Applied Software Measurement." READING (McGraw-Hill. 1996).

30. Jones. T.C.."Measuring Programming Quality and Productivity." IBM SYSTEM JOURNAL 17. no. 1. 1978. PP.39-63

31. Kemerer. Chris F.. "An Empirical Validation of Software Cost Estimation Model." COMMUNICATION OF ACM. May 1987. PP.416-429

32. Kitson. David H.. and Stephen M. Masters. "An Analysis of SEI Software Assessment Results: 1987-1991." IEEE SOFTWARE. August 1993. PP.68-77

33. Major. John E.. "Software Excellence 6 Beyond." First World Congress for Software Quality. June 1995.

34. Paulk. Mark C.. Bill Curtis. Mary Beth Chrissis. and Charles V. Webber. "Capability Maturity Model. Version 1.1." IEEE SOFTWARE. July 1993. PP.18-27

35. Paulk. Mark C.. "The Capability Maturity Model." READING (Addison Wesley. 1995).

36. Putnam. Lawrence H.. "The Economic Value of Moving up the SEI Scale." MANAGING SYSTEM DEVELOPMENT. July 1994. PP.1-6

37. Ring. J.. "How to Break the Software Logjam." FORTUNE. September 1989.

38. Rubin. Howard A.. "Software Process Maturity : Measuring Its Impact on Productivity Quality." INTERNATIONAL SOFTWARE METRICS PROCEEDING. May 1993.

39. Rubin. Howard A.. Margaret Johnson. and Ed Yourdon. "With the SEI As My Copilot. using Software Process 'Flight Simulation' to Predict the Impact of Improvements in Process Maturity." AMERICAN PROGRAMMER. September 1994. PP.50-57

40. Rubin. Howard A.. Margaret Johnson. and Ed Yourdon. "Process Flight Simulation." MANAGING SYSTEM DEVELOPMENT. June 1994. PP.1-7

41. Rubin Howard A.. IT METRICS STRATEGIES. September 1997.

42. Stienen. Hans. and Ernst Lebsanft. "BOOTSTRAP: Practical Experiences with Europe's Software Process Assessment & Improvement Method." First World Congress for Software Quality. June 1995.

43. Symons. Charles R.,"Function Point Analysis Difficulty and Improvements." IEEE TRANSACTIONS OF SOFTWARE ENGINEERING. January 1988.

44. Variable. Tony. Bob Rosetta. Mike Steffen. Howard Rubin. and Ed Yourdon. "Modeling the Maintenance Process." AMERICAN PROGRAMMER. March 1994.

45. Walston. C. E., and C. P. Felix. "A Method of Programming Measurement and Estimation." IBM SYSTEM JOURNAL. no. 1. 1977. PP.54-73

46. Wohlwend. Harvey. and Susan Rosenbaum. "Schlumberger's Software Improvement Program." IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. November 1994. PP.833-839