Impact of CMMI-Based Process Maturity Levels on Effort, Productivity and Diseconomy of Scale

Majed Alyahya, Rodina Ahmad, and Sai Peck Lee
Department of Software Engineering, University of Malaya, Malaysia

Abstract: *The Software Capability Maturity Model Integration (CMMI) has become a popular Software Process Improvement (SPI) model for enhancing software development processes with the goal of developing high-quality software within budget and schedule. Since software development effort can be greatly affected by the organizational process maturity level, this study examines the impact of different CMMI-based process maturity levels on effort, productivity development team and diseconomy of scale for a standard project sizes. The COnstructive COst MOdel (COCOMO) is employed to compute the software development effort. The percentage of change (increase or decrease) in software development effort, productivity and diseconomy of scale is employed as a measure of effectiveness for this study. The results of this work demonstrate that each higher CMMI maturity level has a considerable impact in decreasing the development effort, increasing the productivity rate and reducing the diseconomy of scale. The results also indicate that the impact of CMMI-based maturity levels significantly increases with project sizes.*

Keywords: *CMMI, process maturity, COCOMO II, effort multipliers, scale factors, diseconomy of scale, productivity rate.*

Received February 19, 2010; accepted August 10, 2010

1. Introduction

Developing software to meet functional requirement with acceptable quality, on planned schedule, and within budget is a target pursued by every software development organization [26]. There is a widespread belief that a good software product is a result of mature and repeatable software processes, which have led to more focus on Software Process Improvement (SPI) to assist software development organizations realize its potential benefits. Thus, the search for reliable methodologies, ideas and innovations to enhance software development continues to be an essential focus for both academic and industrial research. Effort spent in this area has resulted in several SPI models and standards such as ISO 9001 [30], Six Sigma [33], and the Carnegie Mellon Software Engineering Institute's Capability Maturity Model for Software (SW-CMM) [31] as well as its most recent version, the Capability Maturity Model Integration (CMMI) [7]. This paper focuses on CMMI. The motivation for selecting CMMI as the base of this study is that it is influential, long-standing, and often-studied standard to SPI [35]. Moreover, CMMI-based SPI has led to quantifiable enhancement in how processes of software engineering are performed [5]. A cording to Jones and Soule [24], among the software process improvement frameworks, CMMI became a standard model with high rate of acceptance.

In this study, we investigate the impact of CMMI-based process maturity levels on software development effort, productivity rate and diseconomy of scale for a set of standard project sizes. COCOMO II model is employed to estimate the required effort in each level of maturity. It has a scale factor input, Process Maturity (PMAT), which is used to assess the organizational process maturity. Since the investigation presented in this work is primarily based on CMMI, and the existing COCOMO II's PMAT values were basically derived for CMM, therefore, based on the results of our recent research, a new set of COCOMO II's PMAT ratings values under CMMI has been derived and evaluated in [1]. Thus, this study relies on the new CMMI-based PMAT rating values.

The rest of this research is organized as follows: section 2 presents a brief background about the CMMI maturity levels in addition to the definition of the COCOMO Model. Research motivation and hypothesis are introduced in section 3. Section 4 surveys some previous studies which are related to this work. Sections 5 and 6 describe the method of this work and the evaluation measure respectively. The results and related discussions are presented in section 7. Finally, section 8 offers some conclusions of this work and presents recommended future works.

2. Background

2.1. CMMI Process Maturity Levels

The Software CMMI is used to rate an organization's process maturity. CMMI provides a number of requirements that all organizations can use in setting up the software processes used to control software product development. CMMI specifies "what" should be in the software process rather than "when" or "for how long". There are five levels of process maturity, level 1 (lowest) to level 5 (highest), where each level signifies the level of performance that can be expected from an organization. For example, maturity level 1 organizations have ad hoc processes, whereas maturity level 2 organizations have a basic project management system in place, and so on. The five CMMI maturity levels are: initial (maturity level 1), managed (maturity level 2), defined maturity level 3), quantitatively managed (maturity level 4), and optimizing (maturity level 5) [7].

2.2. COCOMO II Model

The Constructive Cost Model (COCOMO) was originally published in 1981 (COCOMO 81) [4], and became one of most popular parametric cost estimation models of the 1980s. But in the 90s, COCOMO 81 faced a lot of difficulties and complications in estimating the costs of software that were developed to a new life cycle processes such as non-sequential and rapid development process models, reuse-driven approaches, and object-oriented approaches [2]. Thus, COCOMO II was published initially in the annals of software engineering in 1995 with three sub models; an application-composition model, an early design model and a post-architecture model [2]. COCOMO II has as an input, a set of seventeen Effort Multipliers (EM) or cost drivers which are used to adjust the nominal effort Person-Months (PM) to reflect the software product being developed.

2.2.1. Effort Estimation

In COCOMO II, effort is expressed as PM. Boehm in [3] defines the PM as "the amount of time one person spends working on the software development project for one month". The COCOMO II effort estimation equation is shown in 1.

$$PM_{nominal} = A \times SIZE^{E} \times \prod_{i=1}^{N} EM_i \tag{1}$$

Where

- A is a baseline multiplicative constant = 2.94. It is derived by the COCOMO team by calibrating to the actual effort values for the 161 projects currently in COCOMO II database.
- The exponential factor E is discussed in the next section.
- The EM are used to adjust the effort.
- N is the number of effort multipliers or cost drivers.

2.2.2. Scale Factors (SF)

In addition the 17 effort multipliers that are used as an input to the COCOMO II model, there is a set of five scale factors that account for the economies and diseconomies of scale in software development projects. When there are economies of scale, doubling the software size will result in effort being less than double the original. Whereas when diseconomies of scale are present for a software project, doubling the project size will result in more than double of the original project effort being needed to complete the project [3]. COCOMO II uses equation 2 to calculate if a project has economies or diseconomies of scale:

$$E = B + 0.01 \times \sum_{j=1}^{N} SF_j \tag{2}$$

Where B is a constant=0.91. It is derived by the COCOMO team by calibrating to the actual effort values for the 161 projects currently in COCOMO II database. The exponent E in equation 2 is an aggregation of five Scale Factors (SF). All scale factors have rating levels. These rating levels are Very Low (VL), Low (L), Nominal (N), High (H), Very High (VH) and Extra High (XH). Each rating level has a weight, W, which is a quantitative value used in the COCOMO II model. As shown in Table 1, the five COCOMO II scale factors are Precedentedness, Development Flexibility, Risk Resolution, Team Cohesion, and PMAT [3]. The procedure for determining PMAT- the factor of interest in this study is organized around the Software Engineering Institute's Capability Maturity Model (SEI-CMM).

Table 1. Rating levels and values for COCOMO II scale factors [3].

Scale Factor	**CMM Level 1 (Lower)**	**CMM Level 1 (Upper)**	**CMM Level 2**	**CMM Level 3**	**CMM Level 4**	**CMM Level 5**
	VL	**L**	**N**	**H**	**VH**	**EH**
Precedentedness (PREC)	6.2	4.96	3.72	2.48	1.24	0.00
Development Flexibility (FLEX)	5.07	4.05	3.04	2.03	1.01	0.00
Risk Resolution (RESL)	7.07	5.65	4.24	2.83	1.41	0.00
Team Cohesion (TEAM)	5.48	4.38	3.29	2.19	1.10	0.00
Process Maturity (PMAT)	7.80	6.24	4.68	3.12	1.56	0.00

As can be seen in the first row of Table 1, COCOMO II uses six ratings levels of maturity. The only difference between CMM and COCOMO II ratings levels is in the first maturity level which has been divided in COCOMO II into two halves, lower half and upper half. According to [8], the CMM level 1 (lower half) is for organizations that rely on "heroes" to do the job. They do not focus on processes or documenting lessons learned. The CMM level 1 (upper half) is for organizations that have implemented most of the requirements that would satisfy CMM level 2. In

CMM's published definition, level 1 (lower half) and (upper half) are grouped into level 1.

3. Research Motivation and Hypothesis

3.1. Research Motivation

An increasing number of software development organizations around the world have adopted CMMI to improve their software development processes [37]. In the literature, there are numerous studies on significance and benefits of increasing the CMM-based organizational maturity levels. However, it is observed that there are very limited studies on the impact and benefits of the CMMI-based process maturity. It seems CMM is still receiving much attention than CMMI even though there are continual and widespread demands on the evidence about the impact and benefits of CMMI-based process maturity from organizations that are adopting it [10]. Furthermore, most of the available studies and research which focused on CMMI are case studies based on quantitative data. To the best of our knowledge, qualitative studies on the effect of increasing the CMMI maturity levels are lacked in the literature. It is believed that CMMI, with its high acceptance rate as an SPI framework, requires a special investigation on the impact and benefits of increasing its maturity levels.

3.2. Research Hypothesis

The hypothesis of the work presented here is that increasing the level of CMMI-based process maturity will results in the following:

1. A decrease in the software development effort.
2. An increase in the productivity rate.
3. A reduce the in diseconomy of scale.

As stated earlier, this study relies on the new CMMI-based PMAT rating values which are shown in Table 2.

Table 2. The new PMAT rating values.

PMAT Description	CMMI Level 1 (Lower)	CMMI Level 1 (Upper)	CMMI Level 2	CMMI Level 3	CMMI Level 4	CMMI Level 5
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
New PMAT Values	7.55	5.71	3.81	2.08	1.03	0.00

Since COCOMO II considers CMM level 2 as the nominal rating for PMAT, the same consideration is for CMMI level 2. Thus, the percentage of increase or decrease in effort, productivity and diseconomy of scale, will be compared against the nominal CMMI-based PMAT ratings.

4. Related Literature

Numerous researches and case studies have shown many benefits of enhancing organizational process maturity by using different assessment approaches [6, 22, 23, 27, 36]. Girish *et al.* [18] conducted an empirical study to investigate the effects of CMM on two critical factors in Information Systems (IS) implementation strategy, which are project performance and software quality. They claimed that CMM levels are associated with IS implementation strategies and higher CMM levels are relate to higher project performance and software quality that lead to noticeable reduction in software development effort and schedule. From a review of seventeen published articles, Galin and Avrahami [15] explored CMM-based benefits such as defects, rework, schedule, productivity, error defection effectiveness, and Return on Investment (ROI), concluding that a good investment in CMM programs leads to enhanced software development and maintenance. Diaz and King [11] claimed that increase in CMM process maturity results in an improvement in quality, phase containment, productivity and rework. In order to explore the impact of process maturity on software development effort, and based on CMM with the aid of 161-project sample, Clark in [8] isolated the effects on effort of process maturity versus effects of other factors, and found that an increasing of one organizational process maturity level can result in a reduction in software development effort by 4% to 11%. But this reduction seemed like a generalization across all five levels of CMM process maturity, i.e., the percentage of effort reduction is not the same among all levels. Memon in [29] has reported that CMM maturity levels considerably influences the software development effort and productivity. El-Emam and Goldenson [13] in an comprehensive review of studies and publications on the implementation of SPI methodologies, including CMM, reported qualitative performance improvements in terms of, higher quality, higher productivity, improved ability to meet development schedules. Donald *et al.* [12] conducted an empirical research to find out the relationship between quality of the products, organizational process maturity, development effort, and project schedule. Their findings indicated that process maturity has an effect in reducing software development schedule and effort. Another survey-based study of individuals from SW-CMM-assessed software organizations revealed that higher maturity organizations are associated with better performance, including the ability to meet budget and schedule as well as increase staff productivity, product quality, and customer satisfaction [20]. Herbsleb and Goldenson in [21] showed solid evidence, in a sample of 61 software organizations, that high CMM-based software process maturity is associated with high performance.

Despite the numerous studies that have investigated the performance assessment results of CMM-based software process maturity and its impact on software development effort, there are still very limited works on the overall benefits of CMMI-based software process maturity [37]. Case studies have also shown benefits from CMMI-based software process maturity in [14, 16, 17, 19, 25, 28, 32, 34, 37]. Goldenson and Gibson in [19] reported some great and credible quantitative evidence that CMMI-based software process maturity can help an organization achieve higher quality products and better project performance with lower cost and decreased project effort. Due to the limitation of the performance results provided in [19], Gibson *et al.* [17] continued the assessment performance of CMMI-based software process improvement and provides empirical tangible evidence about the performance results that can achieve as an outcome of CMMI-based process improvement. They reported, "There now is ample evidence that process improvement using the CMMI Product Suite can result in marked improvements in schedule and cost performance, product quality, return on investment and other measures of performance outcome".

5. Research Method

In order to investigate the impact of CMMI-based process maturity level on a variety of software product sizes, it is more recommended to classify the software sizes in an appropriate manner since size is considered the most influential factor in predicting effort of the software product [9]. Boehm in [4] has classified the software product sizes as small, intermediate, medium, large and very large shown in Table 3.

Table 3. Software size classification according to [4].

Classification	Size (KLOC)
Small (S)	2
Intermediate (I)	8
Medium (M)	32
Large (L)	128
Very Large (VL)	512

The scale factor PMAT is used here to capture the impact of different process maturity levels on software development effort for standard size projects classified above.

5.1. Effort Estimation

The basic idea in our research method is quantifying the impact of PMAT versus other factors that influence the software development effort. To do so, we separate the impact of PMAT from other factors because when different kinds of improvements are carried out concurrently in the organization, project managers will have no idea on how to determine the amount of improvement gained from process maturity with the presence of other factors [8]. In this context, COCOMO II model is employed in order to estimate the effort of software development. Subsequently, all effort multipliers are set to be nominal, i.e., the value of 1. Furthermore, as described in Table 4, all scale factors except PMAT (the one related to the process maturity) are also assumed to be nominal. The effort multipliers and scale factors (except PMAT) are set to their nominal values in order to isolate their potential effects on software development effort. As an example, for a nominal PMAT rating and a standard large size project, by substituting values in equations 1 and 2, we get:

$$PM_{nominal}=2.94\times 128^{0.91+0.01\times(18.10)}=585.21\ PM$$

Table 4. Nominal values for all scale factors in all rating levels (except CMMI-based PAMT).

Scale Factor (SF)	CMMI Level 1 (Lower)	CMMI Level 1 (Upper)	CMMI Level 2	CMMI Level 3	CMMI Level 4	CMM I Level 5
	Very Low	Low	Nominal	High	Very High	Extra High
Precedentedness (PREC)	3.72	3.72	3.72	3.72	3.72	3.72
Development Flexibility (FLEX)	3.04	3.04	3.04	3.04	3.04	3.04
Risk Resolution (RESL)	4.24	4.24	4.24	4.24	4.24	4.24
Team Cohesion (TEAM)	3.29	3.29	3.29	3.29	3.29	3.29
New Process Maturity (PMAT)	7.55	5.71	3.81	2.08	1.03	0.00
Summation of All SF	21.84	20.00	18.10	16.37	15.32	14.29

5.2. Productivity Rate

In order to test our hypothesis which is increasing the level of CMMI-based process maturity increases the productivity rate, equation 3 is applied for each estimated effort.

$$Productivity=\frac{Size}{Effort} \tag{3}$$

Where *Size* is the standard size used, and it is measured in this formula by thousand lines of codes (KLOC), and *Effort* is the effort estimated in each PMAT level for all standard sizes.

As an example for the productivity, for nominal PMAT rating and standard large size project, equation 3 will be applied to the effort produced in the previous section.

$$Productivity=\frac{128}{585.21}=218.72$$

5.3. Diseconomy of Scale

Diseconomy of scale refers to relatively more increase in effort as compared to the increase in size of a software product. That is, doubling the project size will result in more than double of the original project effort being needed to complete the project. To make this concept more clear, Memon [29] suggested dividing the standard project sizes discussed earlier by the small size (2 KLOC) and then called from small to intermediate (from S to I), from small to medium (from S to M), from small to large (from S to L), and from small to very large (from S to VL). Similarly, their corresponding efforts are also divided by the effort of small size in order to visualize the effect of diseconomy of scale shown in Table 5.

Table 5. Software sizes ratio.

Classification	Size Ration
From S to I	4
From S to M	16
From S to L	64
From S to VL	256

6. Evaluation Measure

In order to evaluate and compare our proposed results, measures of effectiveness are required. In this study, the percentage change in software development effort, productivity rate and diseconomy of scale are used as the primary Measures of Effectiveness (MOE). The percent change means either increase or decrease in effort, productivity rate and diseconomy of scale. This criterion measure will determine the magnitude of the effect of different process maturity levels on development effort for all standard sizes of software projects. To compute these percentages, we will assume the nominal rating of PMAT as the base case. Hence, the percent of change in effort, productivity, and diseconomy of scale is measured by using equation 4.

$$Percent\ Change = \frac{Parameter\text{-}Parameter_{nomi=al}}{Parameter_{nominal}} \quad (4)$$

Where *Parameter* refers to the computed value of effort, productivity, or diseconomy of scale at a particular PMAT rating. Whereas $Parameter_{nominal}$ refer to the *nominal* value of the same *parameter*. A combination of positive and negative changes will be seen in the resulted values. A negative value indicates percentage reduction in the *parameter* value and a positive one show percentage increase in the *parameter* value.

7. Results and Discussions

7.1. Effort

After applying our method, COCOMO II has estimated the effort for all standard project sizes in each PAMT rating, Table 6 shows the resulted effort, Table 7 shows the percentage change in effort in each process maturity level for all standard size projects, and Figure 1 is a visual representation of Table 7.

Table 6. Estimated effort in all CMMI-based PMAT ratings for all standard sizes.

Project		Effort (PM) Based on PMAT Ratings and Project Size					
Classification	Size	Very Low	Low	Nominal	High	Very High	Extra High
Small	2	6.43	6.35	6.26	6.19	6.14	6.10
Intermediate	8	30.72	29.56	28.42	27.42	26.82	26.25
Medium	32	146.81	137.74	128.96	121.46	117.12	113.01
Large	128	701.65	641.73	585.21	538.09	511.37	486.44
Very Large	512	3353.42	2989.76	2655.59	2383.91	2232.77	2093.81

Table 7. Percent change of effort in all CMMI-based PMAT ratings for all standard sizes.

Project		Average % Change in Effort					
Classification	Size	Very Low	Low	Nominal	High	Very High	Extra High
Small	2	2.63	1.33	0.00	-1.19	-1.91	-2.61
Intermediate	8	8.09	4.03	0.00	-3.53	-5.62	-7.62
Medium	32	13.84	6.81	0.00	-5.82	-9.19	-12.37
Large	128	19.90	9.66	0.00	-8.05	-12.62	-16.88
Very Large	512	26.28	12.58	0.00	-10.23	-15.92	-21.15

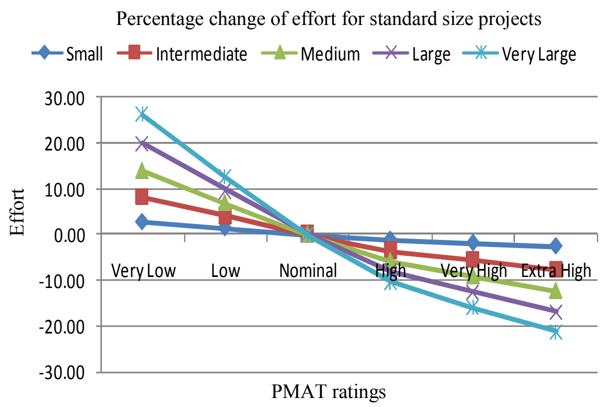

Figure 1. Percent change of effort in all PMAT ratings for all standard sizes.

As shown in Table 6, the results indicate that for each improvement in the PMAT rating, there is a decrease in the required software development effort. It can be noticed that these improvements in the effort are more considerable for larger projects size than for smaller sizes. Also the percentage change shown in Table 7 and graphically in Figure 1 obviously gives the same indications. The percent change in required effort varies from increase of 2.63% for very low rating of PMAT to a decrease of 2.61% for a rating of extra high (a total change of 5.24%) for small size projects, whereas for very large projects, the *percent change* in required effort varies from an increase of 26.28% for very low rating of PMAT to a decrease of 21.15% for a rating of extra high (a total change of 47.43%).

7.2. Productivity Rate

Table 8 and Figure 2 show the resulted productivity. While the percent change in productivity in each process maturity level for all standard size projects are shown in Table 9 and Figure 3.

Table 8. Productivity rate in all CMMI-based PMAT ratings for all standard sizes.

Project		Productivity (KLOC/PM) Based on ISF-PMAT Ratings and Project Size					
Classification	Size	Very Low	Low	Nominal	High	Very High	Extra High
Small	2	311	315	319.34	323	326	328
Intermediate	8	260	271	281.50	292	298	305
Medium	32	218	232	248.13	263	273	283
Large	128	182	199	218.72	238	250	263
Very Large	512	153	171	192.80	215	229	245

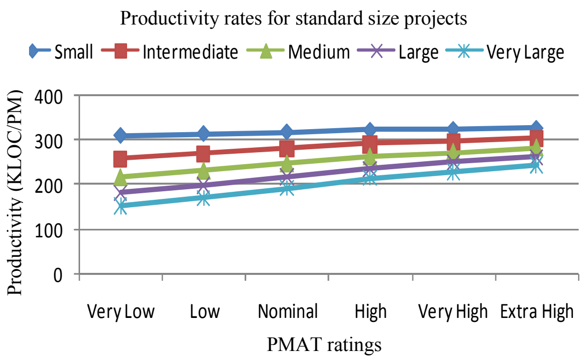

Figure 2. Productivity rate in all PMAT ratings for all standard sizes.

As can be seen in Table 8 and visually in Figure 2, there is a considerable growth in the productivity rate in all PMAT levels. They also give a clear indication that the productivity rate changes (increases) more rapidly for larger projects as compared to smaller projects.

Table 9. Percent change of productivity in all CMMI-based PMAT ratings for all standard sizes.

Project		Average % Change in Productivity					
Classification	Size	Very Low	Low	Nominal	High	Very High	Extra High
Small	2	-2.56	-1.31	0.00	1.21	1.95	2.68
Intermediate	8	-7.48	-3.87	0.00	3.66	5.95	8.24
Medium	32	-12.16	-6.37	0.00	6.18	10.11	14.12
Large	128	-16.60	-8.81	0.00	8.76	14.44	20.31
Very Large	512	-20.81	-11.18	0.00	11.40	18.94	26.83

As shown in Table 9 and graphically in Figure 3, the percent change in productivity rate varies from decrease of 2.56% for very low rating of PMAT to an increase of 2.68% for a rating of extra high (a total change of 5.24%) for small size projects. Whereas for very large projects, the percent change in required effort varies from a decrease of (20.81%) for very low rating of PMAT to an increase of 26.83% for a rating of extra high (a total change of 47.64%).

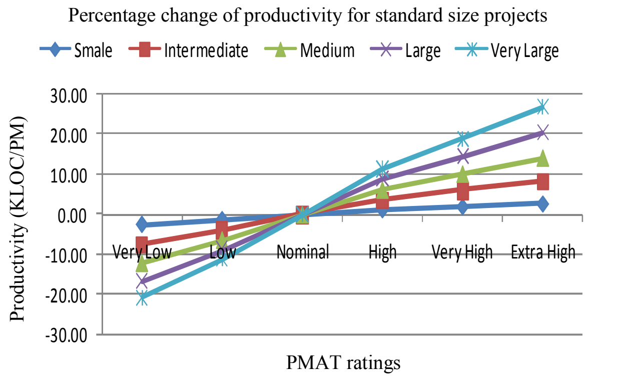

Figure 3. Percent change of productivity in all PMAT ratings for all standard sizes.

7.3. Diseconomy of Scale

When the standard sizes were divided by the small size (2 KLOC), similarly, their corresponding efforts are also divided by the effort of small size to visualize the effect of diseconomy of scale as shown in Table 10 and Figure 4. To visualize the effect of diseconomy of scale, the size ratio has been plotted in Figure 4. The percent change in diseconomy of scale in each process maturity level for all standard size projects are shown in Table 11 and Figure 5.

Table 10. Diseconomy of scale in all CMMI-based PMAT ratings for all standard sizes.

Project		Diseconomy of Scale Based on CMMI-Based PMAT Ratings and Project Size					
Classification	Size Ratio	Very Low	Low	Nominal	High	Very High	Extra High
From S to I	4	4.78	4.66	4.54	4.43	4.37	4.30
From S to M	16	22.84	21.71	20.59	19.63	19.06	18.53
From S to L	64	109.17	101.13	93.44	86.96	83.24	79.75
From S to VL	256	521.74	471.14	424.02	385.24	363.45	343.27

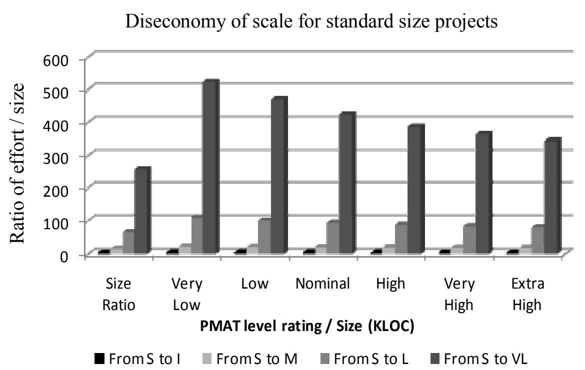

Figure 4. Diseconomy of scale in all PMAT ratings for all standard sizes.

Table 10 and Figure 4 show that diseconomy of scale improves desirably with improvement in PMAT rating levels. They also give an obvious and considerable indication that the productivity rate changes (increases) more rapidly for larger projects as compared to smaller projects.

Table 11. Percent change of diseconomy of scale in all CMMI-based PMAT ratings for all standard sizes.

Project		Average % Change in Diseconomy of Scale					
Classification	Size	Very Low	Low	Nominal	High	Very High	Extra High
From S to I	4	5.32	2.67	0.00	-2.37	-3.78	-5.14
From S to M	16	10.93	5.41	0.00	-4.68	-7.42	-10.02
From S to L	64	16.83	8.22	0.00	-6.94	-10.92	-14.65
From S to VL	256	23.05	11.11	0.00	-9.15	-14.29	-19.04

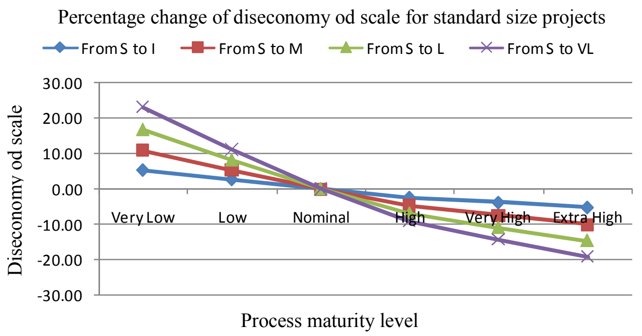

Figure 5. Percent change of diseconomy of scale in all PMAT ratings for all standard sizes.

As shown in Table 11 and graphically in Figure 5, the percent change in diseconomy of scale varies from an increase of 5.32% for very low rating of PMAT to a decrease of 5.14% for a rating of extra high (a total change of 10.46%) for small size projects. Whereas for very large projects, the percent change in required effort varies from an increase of 23.05% for very low rating of PMAT to a decrease of 19.04% for a rating of extra high (a total change of 42.09%).

8. Conclusions and Future Work

This paper investigates the impact of CMMI-based process maturity levels on software development effort, productivity rate and diseconomy of scale for a set of standard project sizes. The result of this investigation has boosted our hypotheses and revealed that the process maturity rating levels based on CMMI considerably affect the software development effort in terms of productivity and diseconomy of scale, i.e., for each higher level of PMAT; there is a decrease in effort required, increase in productivity rate and reduction in diseconomy of scale, although the percentage of (increase/decrease) is not uniform among levels. The results also demonstrate that this effect is more considerable and significant for larger projects as compared to smaller projects. This necessitates the adoption of CMMI-based process improvement for software organizations that are developing projects of large sizes. As a result of investigating the impact of process maturity on effort, it seems reasonable to support the suggestion that PMAT is a significant input to software cost estimation models.

Future work in the area of CMMI-based process maturity requires collecting historical data for each of the 22 process areas used in CMMI in order to examine which process area has most impact on effort, productivity, and diseconomy of scale. In addition, unlike SW-CMM, CMMI has two different representations; Staged and Continuous. This study focused on the staged representation; therefore, further research is recommended to be conducted on the impact of CMMI-based Process capability from the continuous representation perspective.

References

[1] Alyahya A., Ahmad R., and Lee S., "Impact of CMMI Based Software Process Maturity on COCOMO II's Effort Estimation," *The International Arab Journal of Information Technology*, vol. 7, no. 2, pp. 129-137, 2010.

[2] Boehm B., Clark B., Horowitz E., Westland C., Madachy R., and Selby R., "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0," *in Proceedings of Special Volume on Software Process and Product Measurement*, Amsterdam, pp. 45-60, 1995.

[3] Boehm B., Horowitz E., Madachy R., Reifer D., Clark B., Steece B., Brown A., Chulani S., and Abts C., *Software Cost Estimation with COCOMO II*, Prentice Hall, 2000.

[4] Boehm B., *Software Engineering Economics*, Prentice Hall, 1981.

[5] Bollinger T. and McGowan C., "A Critical Look at Software Capability Evaluations," *IEEE Software*, vol. 26, no. 5, pp. 80-83, 2009.

[6] Butler K., "The Economic Benefits of Software Process Improvement," *in Proceedings of Crosstalk, Hill AFB*, USA, pp. 14-17, 1995.

[7] Chrissis M., Konrad M., and Shrum S., *CMMI: Guidelines for Process Integration and Product Improvement*, 3ed, Addison-Wesley, 2003.

[8] Clark B., "Quantifying the Effects of Process Improvement on Effort," *IEEE Software*, vol. 17, no. 6, pp. 65-70, 2000.

[9] Clark, B., "The Effects of Software Process Maturity on Software Development Effort," *PhD Dissertation*, University of Southern California, 1997.

[10] Damian D., Zowghi D., Vaidyanathasamy L., and Pal Y., "An Industrial Experience in Process Improvement: An Early Assessment at the Australian Center for Unisys Software," *in Proceedings of International Symposium on Empirical Software Engineering*, USA, pp. 111-123, 2002.

[11] Diaz M. and King J., "How CMM Impacts Quality, Productivity, Rework, and the Bottom Line," *The Journal of Defense Software Engineering*, vol. 15, no. 3, pp. 9-14, 2003.

[12] Donald H., Krishnan M., and Slaughter A., "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development," *Journal of Management Science*, vol. 46, no. 4, pp. 451-466, 2000.

[13] El-Emam K. and Goldenson D., "An Empirical Review of Software Process Assessments," *Advances in Computers*, vol. 53, pp. 319-423, 2000.

[14] El-Emam K., "TrialStat Corporation: On Schedule with High Quality and Cost Savings for the Customer," *Journal of Software Technology,* vol. 10, no. 1, pp. 24-29, 2007.

[15] Galin D. and Avrahami M., "Are CMM Program Investments Beneficial? Analyzing Past Studies," *Software IEEE*, vol. 23, no. 6, pp. 81-87, 2006.

[16] Garmus D. and Iwanicki S., "Improved Performance Should Be Expected from Process Improvement," *Journal of Software Technology*, vol. 10, no. 1, pp. 14-17, 2007.

[17] Gibson D., Goldenson D., and Kost K., "Performance Results of CMMI-Based Process Improvement," *Technical Report*, Software Engineering Institute CMU/SEI-2006-TR-004 ESC-TR-2006-004, 2006.

[18] Girish H., James J., and Klein G., "Software Quality and IS Project Performance Improvements from Software Development Process Maturity and IS Implementation Strategies," *Journal of Systems and Software*, vol. 80, no. 4, pp. 616-627, 2007.

[19] Goldenson D. and Gibson D., "Demonstrating the Impact and Benefits of CMMI: An Update and Preliminary Results," *Technical Report*, CMU/SEI-2003-SR-009, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003.

[20] Goldenson D. and Herbsleb J., "After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success," *Technical Report*, CMU/SEI-95-TR-009, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1995.

[21] Herbsleb J. and Goldenson D., "A Systematic Survey of CMM Experience and Results," *in Proceedings of 18th International Conference on Software Engineering*, Germany, pp. 323-330, 1996.

[22] Herbsleb J., Carleton A., Rozum J., Siegel J., and Zubrow D., "Benefits of CMM-Based Software Process Improvement: Initial Results," *Technical Report*, CMU/SEI-94-TR-13, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1994.

[23] Humphrey S., Snyder R., and Willis R., "Software Process Improvement at Hughes Aircraft," *IEEE Software,* vol. 8, no. 4, pp. 11-23, 1991.

[24] Jones L. and Soule A., "Software Process Improvement and Product Line Practice: CMMI and the Framework for Software Product Line Practice," *Technical Report*, CMU/SEI-2002-TN-012, Pittsburg, Pennsylvania: Carnegie Mellon University, Software Engineering Institute, 2002.

[25] Liu A., "Motorola Software Group's China Center: Value Added by CMMI," *Journal of Software Technology*, vol. 10, no. 1, pp. 18-23, 2007.

[26] Manish A. and Kaushal C., "Software Effort, Quality, and Cycle Time: A Study of CMM Level 5 Projects," *IEEE Transactions on Software Engineering*, vol. 33, no. 3, pp. 145-156, 2007.

[27] McGibbon T., "A Business Case for Software Process Improvement," *Final Report*, Contract Number F30602-92-C-0158, Data and Analysis Center for Software, Kaman Sciences Corporation, New York, 1996.

[28] McGibbon T., Grader M., and Vienneau R., "The DACS ROI Dashboard-Examining the Results of CMMI® Improvement," *Journal of Software Technology*, vol. 10, no. 1, pp. 30-35, 2007.

[29] Memon G., "Influence of Process Maturity Level on Software Development Effort of Standard Size Projects," *Journal of Information and Communication Technology*, vol. 2, no. 1, pp. 10-18, 2008.

[30] Paulk M., "How ISO 9001 Compares with the CMM," *IEEE Software*, vol. 12, no. 1, pp. 74-83, 1995.

[31] Paulk M., Weber C., Curtis B., and Chrissis M., *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, 1995.

[32] Peter J. and Rohde L., "Performance Outcomes of CMMI-Based Process Improvements," *Journal of Software Technology*, vol. 10, no. 1, pp. 5-8, 2007.

[33] Pyzdek T., *The Six Sigma Handbook: The Complete Guide for Greenbelts*, *Blackbelts, and Managers at All Levels*, McGraw-Hill, 2003.

[34] Sapp M., Stoddard R., and Christian T., "Cost, Schedule and Quality Improvements at Warner Robins Air Logistics Center," *Journal of Software Technology*, vol. 10, no. 1, pp. 10-13, 2007.

[35] Staple M. and Niazi M., "Systematic review: Systematic Review of Organizational Motivations for Adopting CMM-Based SPI," *Journal of Information and Software Technology*, vol. 50, no. 7-8, pp. 605-620, 2008.

[36] Wohlwend H. and Rosenbaum S., "Schlumberger's Software Improvement Program," *Journal of IEEE Transactions on Software Engineering*, vol. 20, no. 11, pp. 833-839, 1994.

[37] Yi T., Sheng T., Ching C., and Huang S., "Assessing the Adoption Performance of CMMI-Based Software Process Improvement in 18 Taiwanese Firms," *Journal of Software Engineering Studies*, vol. 1, no. 2, pp. 96-104, 2006.

Majed Alyahya is a researcher in the area of software engineering. He holds PhD degree in computer science from University of Malaya, Malaysia in 2010. He received his BSc degree in computer science from Zarqa Private University, Jordan in 2002 and MSc degree in computer science from University of Jordan in 2004. His research interests include project management, software process models, software process assessment and improvement, and software cost estimation.

Rodina Ahmad is senior lecturer in software engineering and a research member of ICT cluster at the University of Malaya, Malaysia. She holds PhD degree in information system from the National University of Malaysia in 2006. She obtained her BSc and MSc degree in computer science from Rensselaer Polytechnic Institute, United States. She has more than 19 years of experience in teaching and researching in the area of software engineering and information systems. She has published many conference and journal papers at the national and international level. At present, she supervises seven PhD students in requirements engineering education, software release planning, CMMI framework evaluation, software dynamic metrics and information system. She also leads and teaches courses at both BSc and MSc levels in computer science and software engineering.

Sai Peck Lee is a professor at Faculty of computer science and information Technology, University of Malaya. She obtained her MSc of Computer science from University of Malaya in 1990, her Diplôme d'Études Approfondies (DEA) in computer science from University of Pierre et Marie Curie Paris VI in 1991 and her PhD degree in Computer Science from University of Panthéon-Sorbonne Paris I in 1994. Her current research interests include software reuse, application and persistence frameworks, requirements and design engineering, object-oriented techniques and CASE tools. She has published more than 80 research papers in local and international journals and conferences.