Software Process Improvement and Management:

Approaches and Tools for Practical Development

Shukor Sanim Mohd Fauzi
Universiti Teknologi Mara, Malaysia

Mohd Hairul Nizam Md Nasir
University of Malaya, Malaysia

Nuraminah Ramli
Universiti Pendidikan Sultan Idris, Malaysia

Shamsul Sahibuddin
Universiti Teknologi Malaysia, Malaysia

Information Science
REFERENCE

Managing Director: Lindsay Johnston
Senior Editorial Director: Heather Probst
Book Production Manager: Sean Woznicki
Development Manager: Joel Gamon
Development Editor: Michael Killian
Acquisitions Editor: Erika Gallagher
Typesetters: Mackenzie Snader
Print Coordinator: Jamie Snavely
Cover Design: Nick Newcomer, Greg Snader

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Software process improvement and management: approaches and tools for practical development / Shukor Sanim Mohd Fauzi ... [et al.], editors.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-61350-141-2 (hardcover) -- ISBN 978-1-61350-142-9 (ebook) -- ISBN 978-1-61350-143-6 (print & perpetual access) 1. Software engineering--Case studies. I. Fauzi, Shukor Sanim Mohd, 1980-
QA76.758.S65619 2012
005.1--dc23
2011043142

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 11
Benefits of CMM and CMMI-Based Software Process Improvement

Maged Abdullah
University of Malaya, Malaysia

Rodina Ahmad
University of Malaya, Malaysia

Lee Sai Peck
University of Malaya, Malaysia

Zarinah Mohd Kasirun
University of Malaya, Malaysia

Fahad Alshammari
University of Malaya, Malaysia

ABSTRACT

Software Process Improvement (SPI) has become the survival key of numerous software development organizations who want to deliver their products cheaper, faster, and better. A software process ultimately describes the way that organizations develop their software products and supporting services; meanwhile, SPI on the other hand, is the act of changing the software process and maintenance activities. This chapter purposefully describes the benefits of software process improvement. The Capability Maturity Model (CMM) and the Capability Maturity Model Integration (CMMI) are briefly surveyed and extensively discussed. Prior literature on the benefits and impacts of CMM and CMMI-based software process improvement is also highlighted.

DOI: 10.4018/978-1-61350-141-2.ch011

INTRODUCTION

A new set of ideas on how to enhance the productivity and quality in software development organizations has emerged over the last decade under the term of Software Process Improvement (SPI) (Aaen, Arent, Mathiassen, & Ngwenyama, 2001). SPI has become the survival key of numerous software development organizations who want to deliver their products cheaper, faster, and better. A software process ultimately describes the way that organizations develop their software products and supporting services. Processes define what kind of steps the software development organizations should undertake at each phase of production and provide assistance in making good effort and schedule estimates, measuring quality, and developing plans (Gerry & Rory, 2007). Rico (2004) defines the software process improvement as "an approach to designing and defining a new and improved software process to achieve basic business goals and objectives." SPI is simply the act of changing the software process and maintenance activities. The aims are normally to decrease costs, increase efficiency, and also to increase profitability. For instance, SPI could be employed to create a new and enhanced process for software development organizations.

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 SPI to assist software development organizations realize its potential benefits. Thus, the search for new methodologies, ideas and innovations to enhance software development continues to be an essential focus for both academic and industrial research. In order to improve software development practices, many attempts have concentrated on defining, measuring, and monitoring development activities in an effort to identify and verify improvement areas. These attempts have led to the emergence of the term Process Model. A Process Model is defined as "a structured collection of practices that describe the characteristics of effective processes" (SEI, 2007). An organization can define a process improvement priorities and objectives and make its processes capable, stable, and mature by the help of a process model. Moreover, a process model provides a guideline for an organization to realize its current state; also to identify relevant improvement activities and to identify how to start these activities (SEI, 2007).

Effort spent in this area has resulted in several SPI models and standards such as Personal Software Process (PSP), Team Software Process (TSP) (Humphrey, 1995), ISO 9001 (Paulk, 1995), Six Sigma (Pyzdek, 2003) and the Carnegie Mellon Software Engineering Institute's Capability Maturity Model for Software (SW-CMM) (Paulk, Weber, Curtis, & Chrissis, 1995) and its most recent version, the Capability Maturity Model Integration (CMMI) (Chrissis, Konrad, & Shrum, 2007). The motivation for selecting CMM and CMMI as the base of this chapter is that they are influential, long-standing, and often-studied standard to SPI (Staples & Niazi, 2008). Moreover, CMMI-based SPI has led to quantifiable enhancement in how processes of software engineering are performed (Bollinger & McGowan, 2009). According to Jones and Soule (2002), among the software process improvement frameworks, CMMI became a standard model with high rate of acceptance.

This chapter is organized as follows; a brief introduction of software process improvement is introduced in Section 1. Section 2 describes the benefits of software process improvement. The Capability Maturity Model (CMM) and the Capability Maturity Model Integration (CMMI) are briefly surveyed in Section 3. Section 4 extensively discusses numerous prior literatures on the benefits and impacts of CMM and CMMI-based software process improvement. Section 5 gives a summary of this chapter. Finally, Section 6 presents our conclusions and recommendations.

BENEFITS OF SOFTWARE PROCESS IMPROVEMENT (SPI)

SPI is significant because it is the primary means by which a new and enhanced software development process is created. This is done in order to achieve important economic benefits at the least costs. Research shows that well designed software development process has a positive impact on the economic performance of software projects. Performance is usually measured in terms of productivity as well as the efficiency of the cost (Rico, 2004). On the other hand, poorly designed software development processes have negative consequences on the economic performance of software projects because poor software development process results in high operations cost, ineffective use of available resources, and lost opportunities in the market. According to Rico (2004), poorly designed processes result in a lack of quality and reliability, and poor customer satisfaction. That is why "Software Process improvement has emerged as an important paradigm for managing software development" (Ravichandran & Rai, 2003).

There are several benefits of that can be gained from the adoption of one or more SPI models or standards. As seen in Figure 1, Gibson, Goldenson, and Kost, (2006) described the effect of process improvement. It is used in this chapter for CMM and CMMI-based software process improvement; anyhow, the same depiction can be applied anywhere else. The left box illustrates the cost of process improvement. Some of these might be some planned investments for process improvement; others might be expenses indirectly or directly related to process improvement. Process capability and organizational maturity are shown in the upper center box. Organizations enhance their processes in order to achieve other benefits, and they use process capability and organizational maturity to compare and evaluate their results. The box on the right hand side illustrates different categories of benefits that organizations most frequently struggle to attain as a result of their efforts in process improvement. Also a combination of the costs and benefits can be done to calculate Return on Investment (ROI) or some related measures, as shown in the bottom center box of the Figure 1.

As shown in Figure 1, potential benefits of process improvement may be achieved and classified in six categories; cost, schedule, productivity, quality, customer satisfaction, and the return on investment (ROI). According to (Gibson et al., 2006), enhancement in the above six categories

Figure 1. High-level model of CMMI impact (Gibson et al., 2006)

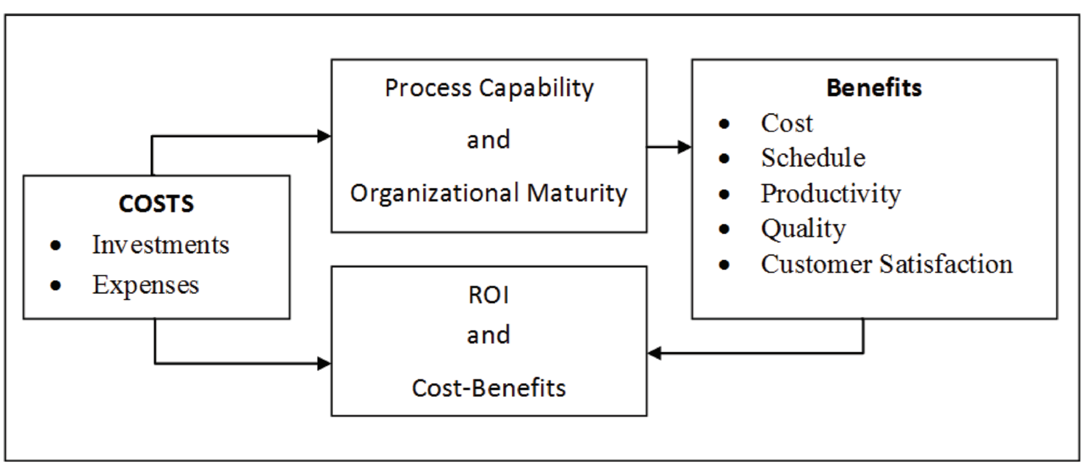

Table 1. Description of the six performance category

Category	Description
Cost	The cost category covers instances where organizations report changes in the cost of final or intermediate work products, changes in the cost of the processes employed to produce the products, and general savings attributed to model-based process improvement. It also includes increased predictability of costs incurred and other measures of variation.
Schedule	This category covers improvements in schedule predictability and reductions in the time required to do the work.
Productivity	This category includes various measures based on the amount of work accomplished in a given period of time.
Quality	Improvement in product quality is most frequently measured by reductions in numbers of defects.
Customer Satisfaction	This category generally includes changes based on customer surveys. Award fees also are sometimes used as surrogate measures.
Return on Investment	In addition to benefit-to-cost ratios, this category includes companion measures of net present value, internal rate of return, payback periods, and break even points.

can participate to further business goals, such as, reduced productivity time, lower cost for products, and higher quality.

Organizations typically seek to adopt some combination of the five basic categories of benefits which are shown in the box on the right hand side; any of the five benefits can be refined to include a variety of more particular measures. For example, an organization might be more concerned in decreasing the costs of its services and products while another organization might be interested in having more reliable predictable project costs, effort, or schedules. The above six categories of performance will be the primary focus of this chapter and it is described in Table 1.

OVERVIEW ON CMM AND CMMI

The Capability Maturity Model (CMM)

The Capability Maturity Model (CMM) focuses on the various processes involved in software development. It presents the key elements of an effective software process in describing an evolutionary improvement path for software organizations from ad hoc, immature processes to mature disciplined one (Paulk et al., 1995). It was created and developed by attentive observations of best practices in both software and non software organizations. The framework is thus based on actual practices while reflecting the best of the state of the practice as well as the needs for individuals performing software process improvement and software process appraisal (Paulk et al., 1995).

The Software Engineering Institute (SEI), at Carnegie Mellon University has released the Software Capability Maturity Model (SW-CMM) in 1991 as a model for enhancing the organization's software processes capabilities, and it is widely used by software organizations. The Capability Maturity Model is a staged evolutionary model. It categorized software process maturity into five levels; Level 1 (Lowest) to Level 5 (Highest), and a set of 18 Key Process Areas (KPA). The organization must demonstrate a capability in a certain number of KPA's assigned to a specific Level in order to be rated at that level. At different maturity levels, key process areas can be used for assessing the capability of existing processes as well as for identifying the area that need to be strengthened so as to move the process to a higher level of maturity. The five maturity levels of CMM are Initial, Repeatable, Defined, Managed, and Optimized.

Since (SW-CMM) was released, it was applied to number of areas; therefore, several capability maturity models have been provided. These included people CMM (P-CMM) (Curtis, Hefley, & Miller, 1995), system engineering CMM (SE-CMM) (EPIC, 1995), the software acquisition CMM (SA-CMM) (Cooper & Fisher, 2002), and the integrated product development CMM (IPD-CMM) (SEI, 1996). As these different models were built by deferent organizations, there were an overlapping in the application's scopes in addition to the lack of consistency in the terminology, assessment approach, and architecture. These problems led to increase the time and cost to adopt multiple models. Therefore, the Software Engineering Institute, SEI, has released in 2000 the Capability Maturity Model Integration (CMMI) in order to integrate all existing capability maturity models. On August, 2000, (CMM) was replaced by a new process model, which is the Capability Maturity Model Integration (CMMI). The Capability Maturity Model Integration (CMMI) was created to reduce redundancy, to support product and process improvement, and to eliminate undesired inconsistency that experienced by organizations that are using multiple models. The CMMI join all relevant process models found in CMM into one product suite (SEI, 2007).

The Capability Maturity Model Integration (CMMI)

CMMI can be described as a group of optimum practices collected from the previous experiences with the preceding CMM, and other models and standards. CMMI defies how influential process must look like. It offers the practitioners with a suitable framework so that enhancement activities can be defined and organized. Moreover, CMMI enables the organization to deal with multi-disciplined activities and to easily combine process improvement aims with organizational business goals (SEI, 2007). The CMMI's product suite consists of process improvement models, training materials, and appraisal methods. Its models cover several disciplines, which are System Engineering (SE), Software Engineering (SW), Supplier Sourcing (SS), and Integrated Product and Process Development (IPPD) (Chrissis et al., 2007).

The Process Areas

The term Key Process Area (KPA) in CMM has been replaced with Process Area (PA) in CMMI model. PA is "a cluster of related practices in an area that, when implemented collectively, satisfies a set of goals considered important for making significant improvement in that area" (Chrissis et al., 2007). In CMMI there are 22 process areas, divided into four categories of process management, project management, Engineering and support.

Choosing a Representation

In CMMI, there are two CMMI models that can be selected for implementation by an organization, Staged and Continuous representation.

Staged Representation

The process areas in staged representation of CMMI are organized into five levels of maturity, which are: Initial (Maturity Level 1, ML1), Managed (Maturity Level 2, ML2), Defined Maturity Level 3, ML3), quantitatively managed (Maturity Level 4, ML4), and Optimizing (Maturity Level 5, ML5). According to Chrissis et al. (2007), the staged representation "prescribes the order for implementing each process area according to maturity levels, which define the improvement path for an organization from the initial level to the optimizing level."

Continuous Representation

The CMMI continuous representation provides the same process areas like the staged representation. However, no process area is allocated to a certain maturity level. The continuous representation grants software firms the flexibility to choose any process area they want to enhance and enable them to choose the order that meet their business objectives or reduces development risk. To measure the achievement of a certain process area for an organization, the continuous representation offers six capability levels, which are: Initial (Capability Level 0, CL 0), Performed (Capability Level 1, CL 1), Managed (Capability Level 2, CL 2), Defined (Capability Level 3, CL 3), Quantitatively managed (Capability Level 4, CL 4), and Optimizing (Capability Level 5, CL 5). The continuous representation offers organizations the flexibility to improve various processes at various rates. As an example, the organization may want to achieve CL 2 in a certain process area and CL 4 in another one.

The term "Maturity Level" is only belongs to staged representation and denotes to an organization's *overall* Process capability and organizational maturity, whereas the term "Capability Level" is only belongs to continuous representation and denotes to an organization's process improvement achievement for *each* process area (Kulpat & Johnson, 2008).

LITERATURE REVIEW ON THE BENEFITS OF CMM AND CMMI-BASED SPI

Much has been discussed on the benefits of increasing organizational process maturity levels as well as the impact and benefits of CMM and CMMI-based software process improvements as surveyed in the two subsequent sections.

Benefits of CMM-Based SPI

Numerous case studies in the SW-CMM have showed benefits from increasing organization's process maturity (Isacsson, Pedersen, & Bang, 2001; Keeni, 2000, Kuilboer & Ashrafi, 2000; Pitterman, 2000; Russwurm & Meyer, 2000; Krasner, 1999; Yamamura, 1999; Diaz & Sligo, 1997; Velden, Vreke, Wal, & Symons, 1996; Deephouse, Mukhopadhyay, Goldenson, & Keller 1996; Haley, 1996; Wohlwend & Rosenbaum, 1994b). Also researchers showed that software organizations adopting the CMM-based SPI tend to have higher quality software, higher development productivity, accurate predictability, and a faster schedule (Humphrey, Snyder, & Willis, 1991; Lipke & Butler 1992; Dion, 1993). Based of an investigation of 31 historical projects, the benefits of the SW-CMM adoption were studied with two primary measures (cost and schedule) which were extracted from U.S. Air Force contracts (Flowe, Lawlis, & Thordahl, 1996). The results of that investigation revealed that projects with more capable and mature processes typically perform much better in term of both cost and schedule performance in compared to those with lower levels of process maturity. Herbsleb and Goldenson (1996) showed solid evidence, in a sample of 61 software organizations, that high CMM-based software process maturity is associated with high performance.

Another survey-based study of individuals from SW-CMM-assessed software organizations revealed that higher maturity organizations is associated with better performance, including the ability to meet budget and schedule, and increase staff productivity, product quality, and customer satisfaction (Goldenson & Herbsleb, 1995). According to Diaz and Sligo (1997), each CMM level enhances the quality of the product and generally reduces the development schedule. They reported that the process maturity level also has some effect on software development schedule by indicating how software process improvement helped Motorola. Based on some measurements, Motorola's software development schedule was around eight times faster at CMM level 5 than at CMM level 1. A study analyzing the development of 30 software products revealed that high software process maturity is positively associated with high software quality (Harter & Slaughter, 2003).

El Emam and Goldenson (2000), 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, and higher customer satisfaction.

In his PhD dissertation, Clark (1997) has reported "the motivation behind the CMM is that a mature software development process will deliver the product on time, within budget, within requirements, and of high quality." He also used six mathematical models to determine the extent that CMM process maturity affects software development effort with the presence of some other factors. He conducted his work by the aid of 112 historical data which completed in 1990, and his results demonstrated that on level change in organizational process maturity results in a 15% to 20% increase in effort reduction. Based on this improvement, Clark proved that process maturity has a great effect on software development effort, and a measure of process maturity should be included in all software cost estimation models. He reported that the positive effects of process maturity are 1) a net saving in effort due to reduction in rework produced by implementing most CMM's key process areas (KPAs); 2) positional high saving in effort for some KPAs such as Peer Review process area. He also indicated that many KPAs are overlapped and should be considered in a certain implementation sequence to take the advantage of this overlap. Girish, James, and Klein (2007) conducted an empirical study to investigate the effects of the 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. This would lead to noticeable reduction in software development effort and schedule. Manish and Kaushal (2007) focused exclusively on CMM level 5 software projects from several organizations to investigate the effects of highly matured processes on development effort, quality, and schedule. Based on historical data projects from 37 CMM level 5 of four software development organizations and by using a linear regression model, they found that high process maturity levels reduce the impacts of factors that were believed to affects software development quality, effort, and schedule such as personnel capability, requirements volatility, and requirements specifications. They also claimed that the only factor found to be important in determining effort, schedule, and quality was the software size. On the average, their developed models estimated effort and schedule around 12% and defects to about 49% of the actuals, across organizations. In general, their results indicated that some of the biggest advantages from high levels of organizational process maturity come from the obvious reduction in variance of software development outcomes that were previously caused by some factors other than size of the software. In a 2004 article, Jiang, Klein, Hwang, Huang, and Hung examined the relationship between CMM-based SPI activities and project performance. From the responses of 154 software project developers, the results revealed that CMM-based SPI is positively associated with project performance.

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 (2000) 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. Despite several researches and case studies have shown many benefits of enhancing organizational process maturity by using different assessment approaches (Humphrey et al., 1991; Herbsleb, Carleton, Rozum, Siegel, & Zubrow, 1994; Wohlwend & Rosenbaum, 1994b; Butler, 1995; McGibbon, 1996), none of them attempts to isolate an individual factors that affect productivity like what Clark did when he separated the impacts of process maturity on development effort versus other factors. Nevertheless, they indicated that increasing organizational maturity levels will generally have some considerable effects. Donald, Krishnan, and Slaughter (2000) have conducted an empirical research to find out the relationship between quality of the products, organizational process maturity, development effort, and project schedule for a set of 30 software products in software development firms. Their findings indicated that process maturity has an effect in reducing software development schedule and effort. Diaz and King (2003) claimed that increase in CMM process maturity results in an improvement in for quality, phase containment, productivity and rework. Herbsleb, Zubrow, Goldenson, and Paulk (1997) showed that organizations with high maturity levels have “excellent” or “good”’ ratings in a number of project performance dimensions such as the ability to stay within budgets, ability to meet schedules, etc. Recently, a review study of seventeen published articles, Galin and Avrahami (2006) 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.

Benefits of CMMI-Based SPI

Despite numerous studies which have investigated the performance assessment results of CMM-based software process maturity and its impact on software development effort and schedule, there are still very limited works on the overall CMMI-based software process maturity (Huang, Sheng, Ching, & Tsen, 2006). (SEI, 2007) reported several benefits of the CMMI-based process improvement, which include:

- Enhanced schedule and budget estimation.
- Enhanced cycle time.
- Increased productivity.
- Enhanced the product’s quality.
- Increased the customer satisfaction.
- Enhanced employee morale.
- Increased return on investment.
- Decreased proposed cost of quality.

Case studies have also shown benefits from CMMI-base SPI (El Emam, 2007; Liu, 2007; Sapp, Stoddard, & Christian, 2007; Peter & Rohdi, 2007; Garmus & Iwanicki, 2007; McGibbon, Grader, & Vienneau, 2007; Gibson et al., 2006; Huang et al., 2006; and Goldenson & Gibson, 2003). Unlike previous studies in the literature that addressed the benefits of CMMI-based software process maturity and in terms of six performance dimensions of Customer, Finance, Quality, Process, Organization, and Employee, Huang et al. (2006) considered performance assessment for both tangible and intangible benefits of CMMI adoption. They presented the results of performance assessment of the CMMI-based Software process improvement based on an empirical study from 18 software firms in Taiwan, which have already obtained CMMI maturity level 2 and 3 certifications. They argued that their empirical study revealed that the CMMI-based software process improvement generally has a positive effect on the six performance dimensions in their investigated software firms but some assessment items do not revealed an important improvement.

Goldenson and Gibson (2003) reported some great and credible quantitative evidence that Capability Maturity Model Integration CMMI-based software process improvement can help an organization achieve higher quality products and better project performance with lower cost and decreased project schedule. The reported

Table 2. Summary benefits and impact from CMMI adoption–Cost

#	Results	Organization
1	33% decrease in the average cost to fix a defect	Boeing, Australia
2	20% reduction in unit software costs	Lockheed Martin M&DS
3	15% decrease in defect find and fix costs	Lockheed Martin M&DS
4	4.5% decline in overhead rate	Lockheed Martin M&DS
5	Improved and stabilized Cost Performance Index	Northrop Grumman IT1

results were drawn from a set of 12 cases from 11 independent firms. Most of the case studies mentioned in this report are provided by organizations that are adopting the CMMI Product Suite. Their results are categorized into four basic categories: cost, schedule, quality, and customer satisfaction. Only the cost and schedule categories are shown here since it is the most relevant to this research. Table 2 summarizes, according to Goldenson and Gibson's results (2003), the benefits and impact of CMMI-based software process improvements from the Cost perspective. The cost category "covers instances where organizations report reductions in the cost of final or intermediate work products, reductions in the cost of the processes employed to produce the products, and general savings attributed to model-based process improvement" (Goldenson & Gibson, 2003).

Table 3 summarizes the results of Goldenson and Gibson (2003) in terms of the benefits and impact of CMMI-based software process improvements from the Schedule perspective. The schedule category "covers two aspects of schedule: improvements in schedule predictability and reductions in the time required to do the work" (Goldenson & Gibson, 2003).

Since the performance results provided by Goldenson and Gibson (2003) are limited, Gibson et al. (2006) continued the assessment performance of CMMI-based software process improvement and provide empirical tangible evidence about the performance results that can achieved as a 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." Results are drawn from a variety of small and huge organizations around the world. They reported that most of their results come from higher maturity organizations, but some notable enhancements also have been achieved by lower maturity organizations. In their

Table 3. Summary benefits and impact from CMMI adoption– Schedule

#	Results	Organization
1	Reduced by half the amount of time required to turn around releases	Boeing, Australia
2	60% reduction in work and fewer outstanding actions following pre-test and post-test audits	Boeing, Australia
3	Increased the percentage of milestones met from approximately 50% to approximately 95%	General Motors
4	Decreased the average number of days late from approximately 50 to fewer than 10	General Motors
5	Increased through-put resulting in more releases per year	JP Morgan Chase
6	30% increase in software productivity	Lockheed Martin M&DS
7	Improved and stabilized Schedule Performance Index	Northrop Grumman IT1

Table 4. CMMI performance results summary

Percentage Category	Median Improvement	Number of Data Points	Lowest Improvement	Highest Improvement
Cost	34%	29	3%	87%
Schedule	50%	22	2%	95%
Productivity	61%	20	11%	329%
Quality	48%	34	2%	132%
Customer Satisfaction	14%	7	-4%	55%
Return on Investment	4.0:1	22	1.7:1	27.7:1
Note:*The performance results in this table express change over varying periods of time.*				

results, the quantitative results obtained for all six performance categories discussed earlier which are Cost, Schedule, Productivity, Quality, Customer Satisfaction, and Return on Investment. The quantitative performance results of Gibson et al. (2006) are summarized in terms of percentage change in Table 4.

CHAPTER SUMMARY

Improving software development or software processes is a primary challenge for present community since the software has become the core and the crucial component of any modern service or product. Therefore, ensuring its quality is essential and should not be ignored. Consequently, SPI is one of the most important and critical efforts that any software development organization pursues.

The Software Engineering Institute (SEI), at Carnegie Mellon University, recommended a set of key software process improvement (SPI) areas. Later on, these new activities were composed into a framework called the Capability Maturity Model (CMM). Since the release of CMM, it has been received wide acceptance and interest in the software engineering community. It has also spread beyond its origins and has been used by thousands of major software development organizations all over the world (Fltzgerald & O'Kane, 1999). After the emergence of CMMI in 2000, it became one of the widely adopted standards for SPI as many software organizations shifted to adopting it (Huang et al., 2006). CMMI has tremendously proven itself as an influential and successful SPI model, and "we can regard the currently matured CMMI as being a pragmatically proven model" (Bollinger & McGowan, 2009). Therefore, it is not surprising to see an increasing number of software development organizations worldwide adopt CMMI-based SPI. Some researchers argued that current SPI frameworks have not yet proven its effectiveness and successfulness. Conradi and Fuggetta (2002) assert that existing SPI initiatives (including CMM and CMMI) are incapable of handling the critical challenges of software organizations. Nevertheless, reviewing the literature on CMM and CMMI-based SPI do indicate its expected benefits. The literature has also has supported the basic hypothesis behind CMM and CMMI which is improving the maturity of the processes should result in better project performance and product quality.

In order to simplify the picture to the reader and to summarize the extensive literature reviewed in this chapter, Table 5 and 6 respectively classify the literature according to the six performance categories discussed in Gibson et al., (2006). In Table 5 and 6, the symbol "×" denotes a certain study has addressed the correspondent performance category. For example, as shown in Table 5, Wohlwend and Rosenbaum (1994a) addressed

Table 5. Summary of the literature on CMM-based SPI

Study	Performance categories					
	Cost	Schedule	Productivity	Quality	customer satisfaction	Return on Investment (ROI)
(Wohlwend & Rosenbaum, 1994a)	-	×	-	×	-	×
(Herbsleb et al., 1994)	×	×	×	×	-	-
(Goldenson & Herbsleb, 1995)	×	×	×	×	×	-
Haley, Ireland, Wojtaszek, Nash, & Dion, 1995)	×	-	×	×	-	×
(Flowe et al., 1996)	×	×	-	-	-	-
(Herbsleb & Goldenson, 1996)	×	×	×	×	×	-
(Lowe & Cox, 1996)	×	×	-	-	×	×
(Krishnan, 1996)	×	×	-	×	-	-
(Yamamura & Wigle, 1997a)	×	×	×	×	-	×
(Yamamura & Wigle, 1997b)	×	×	×	×	-	×
(Herbsleb et al., 1997)	-	×	×	×	-	-
(Vu, 1997)	×	×	×	×	-	×
(Diaz & Sligo, 1997)	-	×	-	×	-	×
(McGarry et al., 1999)	-	×	×	×	-	-
(Oldham, Putman, Peterson, Rudd, & Tjoland, 1999)	-	×	-	-	-	×
(Ferguson, Leman, Perini, Renner, & Sehagiri, 1999)	×	×	×	×	-	×
(Keeni, 2000)	×	×	-	-	-	-
(Clark, 1997; 2000)	×	-	×	-	-	-
(Donald et al., 2000)	×	×	×	×	-	-
(Curtis, 2000)	-	×	-	-	-	-
(El Emam & Goldenson, 2000)	-	×	×	×	×	-
(Pitterman, 2000)	-	×	-	×	-	×
(Blair, 2001)	×	×	×	-	-	×
(McGarry & Decker, 2002)	-	-	×	×	-	×
(Reo, 2002)	-	-	×	-	-	-
(Isaac, Rajendran, & Anantharaman, 2003)	-	-	-	×	-	×
(Diaz & King, 2003)	×	-	×	×	-	×
(Ashrafi, 2003)	-	-	-	×	-	-
(Harter & Slaughter, 2003)	-	-	-	×	-	-
(Girish et al., 2007)	-	-	-	×	-	-
(Manish & Kaushal, 2007)	×	×	-	×	-	-

the benefits of CMM-based SPI on three categories of performance, which are Schedule, Quality, and Return on Investment (ROI). While Goldenson and Gibson (2003) studied the impact of CMMI-based SPI on the whole six categories of performance, as shown in Table 6.

Table 6. Summary of literature on CMMI-based SPI

Study	Performance categories					
	Cost	Schedule	Productivity	Quality	customer satisfaction	Return on Investment (ROI)
(Goldenson & Gibson, 2003)	×	×	×	×	×	×
(Huang et al., 2006)	×	×	×	×	×	-
(Gibson et al., 2006)	×	×	×	×	×	×
(El Emam, 2007)	×	×	-	×	-	-
(Liu, 2007)	×	-	-	×	×	-
(Sapp et al., 2007)	×	×	-	×	-	-
(Peter & Rohde, 2007)	×	-	×	-	-	-
(Garmus & Iwanicki, 2007)	×	×	×	×	-	-
(McGibbon et al., 2007)	-	-	×	×	-	×

CONCLUSION AND RECOMMENDATIONS

At the end of this chapter, a number of observations can be concluded:

- Although numerous research and case studies have reported the performance assessment results and benefits of CMM-based SPI since its release in 1991 till recently, there are still very limited studies on the impacts and benefits of the CMMI-based SPI. Despite the release of CMMI in 2000, it seems that CMM still receiving much attention than CMMI.
- Most of the available studies and paper research which focus the effects and benefits of CMMI-based SPI are case studies which based on quantitative data. Qualitative studies on CMMI-based SPI are rare in the literature and should be further addressed in future research.
- Various researchers have investigated the effect of CMM and CMMI-based SPI on a variety of performance measures such as cost, schedule, and quality. To the best of our knowledge, no previous study has explicitly studied the impact of CMM or CMMI-based SPI on one of the aforementioned classes of performance, which is "Customer satisfaction."
- Prior literatures indicate great improvement in different performance categories when adopting CMM or CMMI-based SPI. It is not yet obvious the kind of correlation which exists between the six performance measures discussed in this chapter. This is a potential topic which deserves more attention in future investigations.
- 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 or key practice has a greater impact on software development effort, productivity of the development team, and diseconomy of scale.
- The literature lack research on the CMMI-based SPI from continuous representation's point of view. All attention is being paid to CMMI staged representation.
- While available and limited case studies on CMMI-based SPI provide great and positive performance results, (Gibson et

al., 2006) stated that these results "cannot necessarily be generalized elsewhere." Therefore, studies are needed to explore the reasons for varying success in varying environment.

From the literature reviewed in this chapter, much has been learned on the effect and benefits of CMM and CMMI-based software process improvement. Now, there is substantial evidence that software process improvement based on CMM and CMMI can result in a considerable improvements in cost, schedule, return on investment (ROI), product quality, and other performance measures.

REFERENCES

Aaen, I., Arent, J., Mathiassen, L., & Ngwenyama, O. (2001). A conceptual map of software process improvement. *Scandinavian Journal of Information Systems*, *12*, 123–146.

Ashrafi, N. (2003). The impact of software process improvement on quality: In theory and practice. *Information & Management*, *40*(7), 677–690. doi:10.1016/S0378-7206(02)00096-4

Blair, R. B. (2001, April). *Software process improvement: What is the cost? What is the return on investment?* Paper presented at the Pittsburgh PMI Chapter meeting.

Bollinger, T., & McGowan, C. (2009). A critical look at software capability evaluations. *IEEE Software*, *17*(4), 89–96.

Butler, K. (1995). The economic benefits of software process improvement. *Crosstalk, The Journal of Defense Software Engineering*, 14-17.

Chrissis, M., Konrad, M., & Shrum, S. (2007). *CMMI: Guidelines for process integration and product improvement* (3rd ed.). Addison-Wesley.

Clark, B. (1997). *The effects of software process maturity on software development effort* (Doctoral dissertation, University of Southern California).

Clark, B. (2000). Quantifying the effects of process improvement on effort. *IEEE Software*, *17*(6), 65–70. doi:10.1109/52.895170

Conradi, H., & Fuggetta, A. (2002). Improving software process improvement. *IEEE Software*, *19*(4), 92–99. doi:10.1109/MS.2002.1020295

Cooper, J., & Fisher, M. (2002). *Software acquisition capability maturity model (SA-CMM)*. (Technical Report CMU/SEI-2002-TR-010). Carnegie Mellon University Software Engineering Institute.

Curtis, B. (2000). The cascading benefits of software process improvement. *Proceedings of the Second International Conference on Product Focused Software Process Improvement. Lecture Notes In Computer Science, vol. 1840, PROFES 2000*. Oulu, Finland: Springer-Verlag.

Curtis, B., Hefley, W. E., & Miller, S. (1995). *People capability maturity model.* (Technical Report CMU/SEI-95-MM-02). Carnegie Mellon University Software Engineering Institute.

Deephouse, C., Mukhopadhyay, T., Goldenson, D. R., & Keller, M. I. (1996). Software processes and project performance. *Journal of Management Information Systems*, *12*(3), 187–205.

Diaz, M., & King, J. (2003). How CMM impacts quality, productivity, rework, and the bottom line. *Crosstalk*. *The Journal of Defense Software Engineering*, *15*(3), 9–14.

Diaz, M., & Sligo, J. (1997). How software process improvement helped Motorola. *IEEE Software*, *14*(5), 75–81. doi:10.1109/52.605934

Dion, R. (1993). Process improvement and the corporate balance sheet. *IEEE Software*, *10*(4), 28–35. doi:10.1109/52.219618

Donald, H., Krishnan, M., & Slaughter, A. (2000). Effects of process maturity on quality, cycle time, and effort in software product development. *Management Science*, *46*(4), 451–466. doi:10.1287/mnsc.46.4.451.12056

El Emam, K. (2007). TrialStat Corporation: On schedule with high quality and cost savings for the customer. *SoftwareTech*, *10*(1), 24–29.

El Emam, K., & Goldenson, D. R. (2000). An empirical review of software process assessments. *Advances in Computers*, *53*, 319–422. doi:10.1016/S0065-2458(00)80008-X

Enterprise Process Improvement Collaboration (EPIC). (1995). *Systems engineering capability maturity model,* version 1.1. *(*Technical Report CMU/SEI-95-MM-003). Carnegie Mellon University Software Engineering Institute.

Ferguson, P., Leman, G., Perini, P., Renner, S., & Sehagiri, G. (1999). *Software process improvement works! (*Technical Report CMU/SEI-99-TR-027). Carnegie Mellon University Software Engineering Institute.

Flowe, R., Lawlis, P., & Thordahl, J. (1996). A correlational study of the CMM and software development performance. *Crosstalk*. *The Journal of Defense Software Engineering*, *8*(8), 21–25.

Fltzgerald, B., & O'Kane, T. (1999). A longitudinal study of software process improvement. *IEEE Software*, *16*(3), 37–45. doi:10.1109/52.765785

Galin, D., & Avrahami, M. (2006). Are CMM program investments beneficial? Analyzing past studies. *Software IEEE*, *23*(6), 81–87. doi:10.1109/MS.2006.149

Garmus, D., & Iwanicki, S. (2007). Improved performance should be expected from process improvement. *SoftwareTech*, *10*(1), 14–17.

Gerry, C., & Rory, O. (2007). Using grounded theory to understand software process improvement: A study of Irish software product companies. *Information and Software Technology*, *49*(6), 654–667. doi:10.1016/j.infsof.2007.02.011

Gibson, D., Goldenson, D., & Kost, K. (2006). *Performance results of CMMI-based process improvement. (*Technical Report CMU/SEI-2006-TR-004 ESC-TR-2006-004). Carnegie Mellon University Software Engineering Institute.

Girish, H., James, J., & Klein, G. (2007). Software quality and IS project performance improvements from software development process maturity and IS implementation strategies. *Journal of Systems and Software*, *80*, 616–627. doi:10.1016/j.jss.2006.06.014

Goldenson, D., & Gibson, D. (2003). *Demonstrating the impact and benefits of CMMI: An update and preliminary results.* (Technical Report CMU/SEI-2003-SR-009). Carnegie Mellon University Software Engineering Institute.

Goldenson, D., & Herbsleb, J. (1995). *After the appraisal: A systematic survey of process improvement, its benefits, and factors that influence success.* (Technical Report CMU/SEI-95-TR-009). Carnegie Mellon University Software Engineering Institute.

Haley, T. (1996). Software process improvement at Raytheon. *IEEE Software*, *13*(6), 33–41. doi:10.1109/52.542292

Haley, T., Ireland, B., Wojtaszek, E., Nash, D., & Dion, R. (1995). *Raytheon electronic systems experience in software process improvement. (Technical Report CMU/ SEI 95-TR-17)*. Carnegie Mellon University Software Engineering Institute.

Harter, D. E., & Slaughter, S. A. (2003). Quality improvement and infrastructure activity costs in software development: Longitudinal analysis. *Management Science*, *49*, 784–800. doi:10.1287/mnsc.49.6.784.16023

Herbsleb, J., Carleton, A., Rozum, J., Siegel, J., & Zubrow, D. (1994). *Benefits of CMM-based software process improvement: Initial results.* (Technical Report CMU/SEI-94-TR-13). Carnegie Mellon University Software Engineering Institute.

Herbsleb, J., Zubrow, D., Goldenson, D., & Paulk, M. (1997). Software quality and the capability maturity model. *Communications of the ACM*, *40*(6), 30–40. doi:10.1145/255656.255692

Herbsleb, J. D., & Goldenson, D. R. (1996). A systematic survey of CMM experience and results. In *Proceedings of the 18th International Conference on Software Engineering* (pp. 323–330). Washington, DC: IEEE Computer Society.

Huang, S., Sheng, T., Ching, C., & Tsen, Y. (2006). Assessing the adoption performance of CMMI-based software process improvement in 18 Taiwanese firms. *Journal of Software Engineering Studies*, *1*(2), 96–104.

Humphrey, W. S. (1995). *A discipline for software engineering*. Addison-Wesley.

Humphrey, W. S., Snyder, T. R., & Willis, R. R. (1991). Software process improvement at Hughes Aircraft. *IEEE Software*, *8*(4), 11–23. doi:10.1109/52.300031

Isaac, G., Rajendran, C., & Anantharaman, R. N. (2003). Do quality certifications improve software industry's operational performance?—Supplemental material. *Software Quality Professional, 6*(1).

Isacsson, P., Pedersen, G., & Bang, S. (2001). Accelerating CMM-based improvement programs: The accelerator model and method with experiences. *Software Process Improvement and Practice*, *6*, 23–34. doi:10.1002/spip.133

Jiang, J., Klein, G., Hwang, G., Huang, J., & Hung, S. (2004). An exploration of the relationship between software development process maturity and project performance. *Information & Management*, *41*, 279–288. doi:10.1016/S0378-7206(03)00052-1

Jones, L., & Soule, A. (2002). *Software process improvement and product line practice: CMMI and the framework for software product line practice.* (Technical Report CMU/SEI-2002-TN-012). Carnegie Mellon University Software Engineering Institute.

Keeni, G. (2000). The evolution of quality processes at Tata Consultancy Services. *IEEE Software*, *17*(4), 79–88. doi:10.1109/52.854073

Krasner, H. (1999). The payoff for software process improvement: What it is and how to get it. In El Emam, K., & Madhavji, N. H. (Eds.), *Elements of software process assessment and improvement* (pp. 151–176). Los Alamitos, CA: IEEE Computer Society Press.

Krishnan, M. S. (1996). *Cost and quality considerations in software product management* (Doctoral dissertation, Graduate School of Industrial Administration, Carnegie Mellon University).

Kuilboer, J. P., & Ashrafi, N. (2000). Software process and product improvement: An empirical assessment. *Information and Software Technology*, *42*, 27–34. doi:10.1016/S0950-5849(99)00054-3

Kulpat, M., & Johnson, K. (2008). *Interpreting the CMMI: A process improvement approach* (2nd ed.). Germany: Auerbach Publications.

Lipke, W. H., & Butler, K. L. (1992). Software process improvement: A success story. *Crosstalk. The Journal of Defense Software Engineering*, *38*, 29–31.

Liu, A. Q. (2007). Motorola Software Group's China center: Value added by CMMI. *SoftwareTech*, *10*(1), 18–23.

Lowe, D. E., & Cox, G.M. (1996). Implementing the capability maturity model for software development. *Hewlett- Packard Journal*, 1–11.

Manish, A., & Kaushal, C. (2007). Software effort, quality, and cycle time: A study of CMM level 5 projects. *IEEE Transactions on Software Engineering*, *33*(3), 145–156. doi:10.1109/TSE.2007.29

McGarry, F., & Decker, B. (2002). Attaining level 5 in CMM process maturity. *IEEE Software*, *19*(6), 87–96. doi:10.1109/MS.2002.1049397

McGarry, F., Pajerski, R., Page, G., Waligora, S., Basili, V., & Zelkowitz, M. (1999). *Software process improvement in the NASA Software Engineering Laboratory.* (Technical Report CMU/SEI-94-TR-22). Carnegie Mellon University Software Engineering Institute.

McGibbon, T. (1996). *A business case for software process improvement*. Utica, New York: Data and Analysis Center for Software, Kaman Sciences Corporation.

McGibbon, T., Grader, M., & Vienneau, R. (2007). The DACS ROI dashboard-Examining the results of CMMI® improvement. *SoftwareTech*, *10*(1), 30–35.

Oldham, G., Putman, B., Peterson, M., Rudd, B., & Tjoland, K. (1999). Benefits realized from climbing the CMM ladder. *Crosstalk*. *The Journal of Defense Software Engineering*, *12*(5), 7–10.

Paulk, M., Weber, C., Curtis, B., & Chrissis, M. (1995). *The capability maturity model: Guidelines for improving the software process*. Addison-Wesley.

Paulk, M. C. (1995). How ISO 9001 compares with the CMM. *IEEE Software*, *12*(1), 74–83. doi:10.1109/52.363163

Peter, J., & Rohde, L. (2007). Performance outcomes of CMMI®-based process improvements. *SoftwareTech*, *10*(1), 5–8.

Pitterman, B. (2000). Telcordia Technologies: The journey to high maturity. *IEEE Software*, *17*(4), 89–96. doi:10.1109/52.854074

Pyzdek, T. (2003). *The Six Sigma handbook: The complete guide for greenbelts, blackbelts, and managers at all levels*. McGraw-Hill.

Ravichandran, T., & Rai, A. (2003). Structural analysis of the impact of knowledge creation and knowledge embedding on software process capability. *IEEE Transactions on Engineering Management*, *50*(3), 270–284. doi:10.1109/TEM.2003.817278

Reo, D. (2002, April). *CMM for the right reasons*. Paper presented at the ASQF CMM Day, European Software Institute.

Rico, D. F. (2004). *ROI of software process improvement: Metrics for project managers and software engineers. J*. Ross Publishing.

Russwurm, W., & Meyer, L. (2000). Integrated evaluation procedure for software/hardware system development processes based on the software capability maturity model. *Software Process Improvement and Practice*, *5*, 231–242. doi:10.1002/1099-1670(200012)5:4<231::AID-SPIP128>3.0.CO;2-#

Sapp, M., Stoddard, R., & Christian, T. (2007). Cost, schedule and quality improvements at Warner Robins Air Logistics Center. *SoftwareTech*, *10*(1), 10–13.

Software Engineering Institute. (1996). *EPIC: An integrated product development capability maturity model,* version 0.9. Carnegie Mellon University, Software Engineering Institute. Retrieved from http://www.sei.cmu.edu

Software Engineering Institute. (2007). *Capability maturity model integration (CMMI.) version 1.2.* Carnegie Mellon University, Software Engineering Institute. Retrieved from http://www.sei.cmu.edu

Staples, M., & Niazi, M. (2008). Systematic review: Systematic review of organizational motivations for adopting CMM-based SPI. *Information and Software Technology*, *50*(7-8), 605–620. doi:10.1016/j.infsof.2007.07.003

Velden, M., Vreke, J., Wal, B., & Symons, A. (1996). Experiences with the capability maturity model in a research environment. *Software Quality Journal*, *5*, 87–95. doi:10.1007/BF00419772

Vu, J. D. (1997). *Software process improvement journey (from level 1 to level 5)*. Paper presented at the 2nd Annual European Software Engineering Process Group Conference.

Wohlwend, H., & Rosenbaum, S. (1994a). Software improvements in an international company. *Proceedings of the 15th International Conference on Software Engineering*.

Wohlwend, H., & Rosenbaum, S. (1994b). Schlumberger's software improvement program. *IEEE Transactions on Software Engineering*, *20*(11), 833–839. doi:10.1109/32.368125

Yamamura, G. (1999). Process improvement satisfies employees. *IEEE Software*, *16*(5), 83–85. doi:10.1109/52.795105

Yamamura, G., & Wigle, G. B. (1997a). SEI CMM level 5: For the right reasons. *Crosstalk*. *The Journal of Defense Software Engineering*, *10*(8), 3–6.

Yamamura, G., & Wigle, G. B. (1997b). Practices of an SEI CMM level 5 SEPG. *Crosstalk*. *The Journal of Defense Software Engineering*, *10*(11), 19–22.