11-1-2001

CMM and CMMI: A Comparison and Evaluation Of the Benefits of Integrated Approach.

Raghunath Shapkota

CMM and CMMI: A Comparison and Evaluation Of the Benefits of Integrated Approach

A Thesis Presented to the
Department of Computer Science
And the
Faculty of the Graduate College
University of Nebraska

In Partial Fulfillment
Of the Requirement for the Degree
Master of Science

University of Nebraska at Omaha

By
Raghunath Shapkota
November 2001

Thesis Acceptance

Acceptance for the faculty of the Graduate College, University of Nebraska, in Partial fulfillment of the Requirement for the degree Master of Science in Computer Science, University of Nebraska at Omaha.

Committee

Hassan Farhat
Department of Computer Science

Griff Elder
Department of Mathematics

Mansour Zand
Thesis Advisor and Committee Chairperson
Department of Computer Science

Date 11/30/01

Acknowledgement

I wish to express my gratitude and respect to Dr. Mansour Zand for his advice, help, suggestions and encouragement that he gave me in completing my thesis. I sincerely appreciate him and I will be forever indebted to him for being a mentor and guiding me throughout my thesis.

I also appreciate and thank my thesis committee member Dr. Hassan Farhat and Dr. Griff Elder for their time, consideration, and suggestions throughout my research.

My special gratitude to Dad and Mom, they always inspired me to achieve higher goals in life and always stood by me in achieving the milestone that I took for. I was able to achieve everything with their support and encouragement. Finally thanks to my brother and sister who helped and supported me in every aspect of life and for their love and understanding.

I dedicate my work to my Family.

Abstract

CMM and CMMI: A Comparison and Evaluation of the Benefits of Integrated Approach

Raghunath Shapkota, MS
University of Nebraska, 2001

Advisor: Mansour Zand

Model based process improvement involves the use of a model to guide the improvement of an organization's processes. Essentially, process capability is the inherent ability of a process to produce planned results. As the capability of a process increases, it becomes predictable and measurable, and the most significant cause of poor quality and productivity are controlled or eliminated. By steadily improving its process capability, the organization matures. One means of achieving this focus has been the use of a capability model. Models provide a common set of process requirements that capture best practice and practical knowledge in a format that can be used to guide priorities.

There are different model used in the industry for the process improvement, commonly and widely used is the CMM model for software. Recognizing the widespread use of CMMs throughout industry and the government, CMMI model was released in August 2000. This model provides an integrated approach across the enterprise for improving processes, while reducing the redundancy, complexity and cost resulting from

the use of separate and multiple models. CMMI model is analogous to CMM model and this integrated model might receive the same criticism from the small organizations as more process areas have been added to the integrated model. As the model is very new there is growing concern among the industries about the scope and applicability of the model. This thesis discusses on these issues and provides a set of recommendations that might be helpful for the organizations to decide the applicability and the scope of the integrated model. Thesis gives a brief description about the two models, CMM and CMMI and also present some recommendations to be consider while making the transition from the currently used model to the integrated model.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction and Problem Definition

1.1 Introduction

A software process can be defined as a set of activities, methods, practices and transformations that software organizations employ to develop and maintain software and associated products [27]. The principle objective of a software process is to produce quality products, which satisfy the customer's need. A software process must be predictable in that cost-estimates and schedule commitments have to be met with reasonable consistency. The resulting products, which are produced, have to meet the user's functional and quality expectations.

A process's capability is its inherent ability to produce planned results. A software organization should hence, try to improve the capability of the software process in order to enhance quality and overall productivity. Software process improvement (SPI) techniques are used towards the end of improving the capability of software process. A software organization, which produces a software product, ultimately has to concentrate on the quality of the product they produce. An organization can have a "process-oriented" way of thinking or a "result-oriented" way of thinking. The "result-oriented" way of thinking totally concentrates on the end product without concentrating on the way the product was produced. Though this may lead to the development of the end product, it cannot guarantee predictable results always. As the concept is based on the result rather than on the process, the end product may have to be tested extensively before being handed out to the customer. Under such circumstances, overall, there is little in the way of software-quality-assurance practices besides testing. The quality of the software product is influenced to a large extent by the software development process [7].

A "process-oriented" way of thinking for developing software is more practical. If an organization concentrates on the process rather on the end result, then it can claim that it can produce predictable results. A process-oriented approach requires that an organization make attempts to improve on its software process in order to achieve the ultimate goal of producing quality software.

A software development process involves the development of the software from the beginning to the end, when the software is finally handed over to the customer. A SPI technique/method is an integrated collection of procedures, tools and training for product quality, improving development team productivity or reducing development time. There is an increase interest in improving the software development process. SPI techniques involve a strategy of selective insertion of new methods and technology into the software development process in combination with actions to optimize the existing process. SPI technique enables an organization to do a better job with its cost and schedule estimations and this enables them to gain the customer confidence in them.

CMMs have been in use for various disciplines with the intent of providing a model of best practices for each of the intended disciplines. Users of these models have demonstrated that product and process improvements are achievable by institutionalizing processes consistent with the practices. In a complex environment, such as development where several of these disciplines are employed, the collective use of individual models has resulted in redundancies, additional complexity, increased costs and at times, discrepancies. To improve the efficiency of model use and increase the return on investment, the CMMI project was created to provide a single integrated set of models. Since not all organization employ every discipline, the project also provides CMMI models for individual disciplines. Since not all processes apply equally to all organizations, the CMMI models are tailorable to an organizations mission and business objectives and criteria for tailoring area provided.

Initially, the CMMI project includes the disciplines of Systems engineering, Software engineering and Integrated Product and Process Development (IPPD). A framework is provided that generates product for each of these disciplines as well as allowing for new disciplines that can be added in the future. A common set of process areas is provided that forms the core of an integrated capability model and applies to all disciplines. Although the initial intent of the CMMI project was to focus on processes used by developers of systems and products, the common process areas as defined will support other disciplines and should be considered for other use. To completely define the discipline, process areas unique to a discipline are also provided. Each of the process areas provides a model of best practices.

Recognizing the widespread use of CMMs throughout industry and the government, the CMMI project has included the objective of preserving the investment that have been made to improve processes. The intent is to allow industry and the government to continue to improve by building on the investment they have already made in process improvement.

1.2 Research Objective

A number of different models exists for Software process improvement these are CMM, CMMI, ISO 9000-3, BOOTSTRAP, Trillium, SPICE and AMI. CMM has been widely used as a standard model in the software industry for many years. CMMI SE/SW was released in August 2000. The CMMI effort is intended to support process and product improvement and to reduce redundancy and eliminate inconsistency when using separate stand-alone models. The goal is to improve efficiency, return on investment, and effectiveness by using models that integrate discipline such as systems and software engineering. Although this model has been released recently very limited information is available for the organization that wish to make a transition. As from my research point of view there are no data available about the success of the new CMMI model. At this stage the organization are reluctant to make the transition. There is confusion among the organizations and industry leaders as whether to make a transition or not? The objective of my thesis work is to explore the CMMI model; it mainly focuses on the issues of How to help the organization in making the right decision about the transition? This thesis compares between the CMM and CMMI model to give the user the difference between the two models, and then recommends the conditions that have to be considered in moving from CMM to CMMI. The thesis also recommends the conditions that are favorable for making a transition and the conditions that are not favorable in making transition. This could be a very helpful guide and wealth of information for the organization that are confused about the new model and the organization that are planning for making a transition.

1.3 Thesis Organization

The thesis is organized as follows:

Chapter 1: This chapter provides the introduction, research objectives and the current problem.

Chapter 2: This chapter provides an introduction and the objectives of the software process improvement. The need for carrying out the software process improvement and managing software process Improvement.

Chapter 3: This chapter provides an introduction to the software process improvement models. The two models namely CMM and CMMI are discussed in greater detail based on the architecture, assessment and different criteria. It explains the need for the transitions.

Chapter 4: This chapter provides some useful recommendations that have to be considered in making the transition and the conditions that are favorable for making the transitions and the conditions that are not favorable for transition.

Chapter 5: This chapter provides the contributions made in this thesis in brief and the direction for further research.

Chapter 2

Software Process Improvement

2.1 Introduction

Developing reliable and usable software that is delivered on time and within budget is a difficult endeavor for many organizations. Products that are late, over budget, or don't work as expected also cause problems for the organization's customer. As software projects continue to increase in size and importance, these problems become magnified. These problems can be overcome through a focused and sustained effort at building a process infrastructure of effective software engineering and management practices. As software organization throughout the world strive to increase their processes and products, those organization that develop a culture that effectively implements sound engineering practices ultimately will beat out the competition [9]. Software process Improvement (SPI) is the software industry's most recent approach to increase effectiveness. Since its start in the late 80s the software process improvement has gained considerable importance and is now a dominant approach for improving quality, productivity, and adherence to schedule in software developing companies. Many of the excellent guidelines that support practitioners in implementing SPI in their organization rely on normative models for good software engineering practice to support the SPI process [10]. The Most popular and widely used models are the Capability Maturity Model, BOOTSTRAP, and SPICE. These models contains maturity levels indicating good software practice and to prioritize future improvements.

2.2 The need for a Software process

When a team attempts to build a new software program they are undertaking what is unquestionably the most complex undertaking of modern society. The fact that software ever works is something of a tribute to the determination of many very smart people, plus the development of sophisticated tools and techniques to help manage complexity. When measuring software difficulties, most people look to examples of software cost and schedule overruns, and there are certainly numerous examples that we can look at. In fact, however, the more significant costs are often to be found well outside the bounds of the software development project and within the business model itself. As shown in the figure, the true cost of software failure shows up in the form of lost business and reduced overall profitability [8]. Let us consider for example a business that fails to recognize the need for a new software program to keep up with competitors or with lead competitor, may find itself unprofitable and subject to acquisition or even failure.

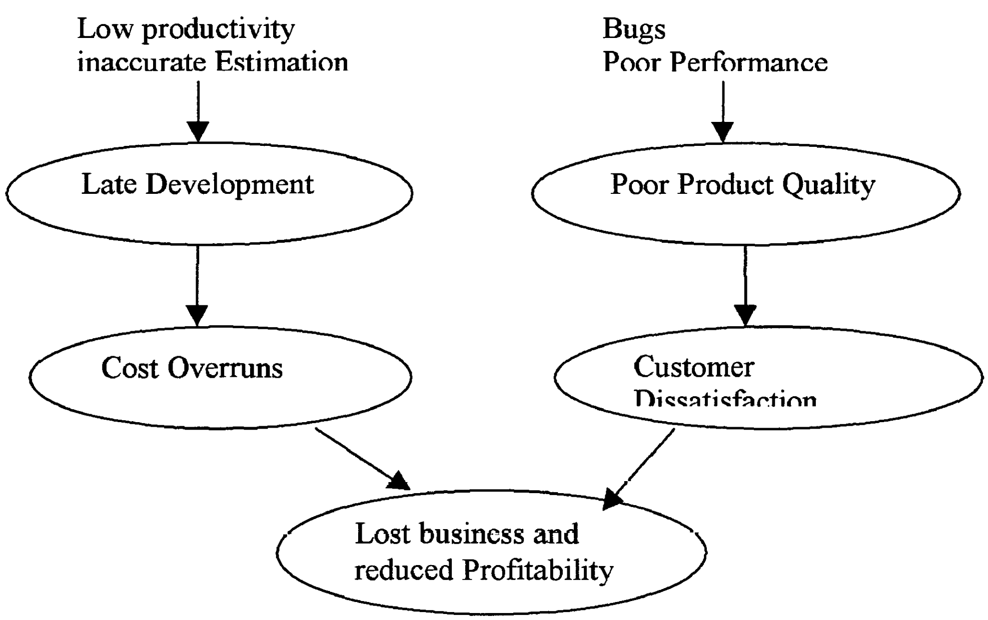


Figure 2.1 Example of Business Failure

Many organizations have invested significant efforts in understanding and quantifying the skills with which organization build software. It has been proved all the time that leading edge companies score above average in terms of consistent, predictable software development. Trailing edge companies recognize that there is a problem, but mistakenly look for quick fixes. Hence they tend to be very susceptible. A significant part of the problem is caused by a chaotic software development process. Success or failure is dependent on individual skills and experience. Significant effort is wasted throughout the development project because of duplication, false starts, and lack of software reuse. Learning curves are high for new employee or the existing employee just joining the team further complicating the problem. As the project ramps up, experienced staff spend more and more time training new staff, or correcting errors caused by inadequately trained new staff. It has been found out that the solution lies in managing software as a repeatable, controlled process rather than an ad hoc chaotic environment. It has been found out that

- Repeated processes should be standardized and be repeatable.
- Training should communicate these processes.
- Metrics should be gathered to allow repeatable process to be predicted.
- Deviations from past performance should be identified and analyzed.
- Processes should improve with time as more is learned.

Software process measurement models look at an organization in terms of its software development maturity; this maturity is a strong indicator of the software process capability of the organization.

2.3 Why carry out the Software Process Improvement?

The increase in maturity within an organization requires that processes are being institutionalized. This implies the need for a framework and a culture to support the methods, practices and procedures so that they are incorporated with the business. This results in software processes that are effective, usable and consistently applied by the organization. When the process capability within an organization increases, the predictability and the performance are enhanced. This implies that the organization will be able to make a better prediction of the ability to meet its goals for a project. This in turn leads to several benefits for both the customer and the development organization. Due to the improved working process and the better estimates of efforts, time and risks of a project; the development and maintenance cost decreases. This leads to an increasing ability to meet quantitative and qualitative objectives for the organization. For the customer this means that the development organization can be more responsive to the customer and that the user satisfaction is maximized since the demands are fulfilled.

2.4 Making Software Process Improvement Happen

Every software developing organization seeks to meet at least three fundamental objectives they are as follows:

1. The software fulfils the requirements.
2. The software is developed on schedule.
3. The software is developed within budget.

Many software-developing organizations find it difficult to meet these fundamental objectives [11]. SPI if followed helps to improve the performance of a software-developing organization towards meeting these objectives. There are many ways that can be considered in making SPI happen within an organization, this thesis discusses some of the most important method as seen and widely used in the next section under management view on software process improvement.

2.5 Managing Process Improvement

This section describes the problems in managing a process improvement program, motivating people, and getting commitment. It focuses on presenting a management view of a process improvement program.

2.5.1 Fundamental Question

In the preface of his book, "Managing the Software Process", Watts S. Humphrey lists three fundamental questions that must be answered when improving the software process

1. How good is my current software process?
2. What must I do to improve it?
3. Where do I start?

For the software manager, planning to carry out a software process improvement program, the two most important questions are:

1. How do I manage the improvement program?

2. How do I motivate the organization to take on the challenge?

If we can find answers to these five fundamental questions, then we have the means to carry out a successful process improvement program [12].

2.5.2 Software Process Assessment

A software process assessment is required to identify the highest priority areas for improvement in the current process, and to provide recommendations on how to make the improvements. The goal of the assessment should be to recognize the problems, not to provide immediate solutions, as solutions to problems that are not defined are seldom effective [12].

An assessment can be carried out by following a software process assessment model like CMM or SPICE. These models contain thorough questions for each software process area to assess and rate their level of maturity. These models have mainly been designed for large software organizations, to be carried out by trained assessors. There are three phases in carrying out an assessment: preparation, assessment, and recommendations. The preparation phase seeks the management's commitment and approval for the assessment. The assessment itself is carried out by a trained assessor and a team of interviewers. Finally, recommendations are prepared for a process improvement program.

In small organizations and in organizations where process maturity is low, an assessment can also be made by interviewing the key people who carry out the current process. The problem is not the difficulty of finding out where the problems are, but to list and define them, and then select and prioritize the items for improvement.

Using a process assessment model in a low maturity organization can result in people becoming less motivated towards the process improvement program. The assessment questionnaire can contain too many questions that have to be answered: "no our process does not comply." Personnel interviews, on the other hand can be a good opportunity to motivate each person to commit himself to improve the problems that he or she points out during the interview.

An assessment, based on an assessment questionnaire or personnel interviews should follow the five assessment principles set by Humphrey [12]:

1. Base the assessment on a process model, a common view of the current software process to avoid being sidetracked to intuitive exploration.
2. Obey strict confidentiality, do not record interviews or report problems to the management against people's wishes, which would only make peoples attitude reserved towards to process improvement program and make them less likely to talk about the problems openly.
3. Involve senior management personally in the program to make them committed in approving the process improvement plans and follow-up programs.
4. The assessment has to be made with an open mind and with respect to the views of the interviewed people.
5. The assessment has to focus on action, on the current problems and on how to solve them.

2.5.3 Software Process Change

A failure to carry out the recommended process improvement actions can be a result of overlooking the requirements for process change. Several facts have to be followed in order to change the current process: committed resources, proper planning and coordination, step-by-step improvement, and training [12].

To change the software process, someone must work on it. Managers should note that merely asking people to work harder would not result in change. Especially in small organizations people are already too busy to take the extra work, no matter how committed they are. The same applies to planning, as everyone must participate in making the plans for improvement to be committed in the new process.

People do not naturally perform well in complex tasks. As a result the process improvement actions should be made in small steps that let people experiment and learn the new process in a series of small steps rather than changing everything overnight and trying to manage with the resulting chaos. Some of the changes might require courses to be prepared or specialists to be hired from outside to help with new tools or ways to perform the new process. Training can be expensive, but not always as expensive as reverting to the trial and error method of learning or, at worst, to the old way of doing the tasks.

2.5.4 Management View on Software Process Improvement

2.5.4.1 Management Responsibility

The managers of process improvement programs have the difficult task of managing a project governed mainly by human factors. The reason for this is that as software process is carried out mainly by human beings, with very little automation, the changes must also be made to the way people act and carry out their tasks. As a result the manager has to make a major effort on tackling change resistance, motivating people and communicating change. The change process starts by understanding the problems that were found during the assessment, and making the organization aware of them. From there on, the manager's job is to set the process improvement goals and motivate and guide the organization through all the phases required to define solutions to the problems and reach the goals set for the process improvement program.

An important task for the process improvement program manager is to remind and remind again about the process improvement actions that are to be done. People tend to do the tasks that they are reminded most often about and tasks that are demanded by the most influential managers.

2.5.4.2 Motivation

The whole organization has to make a commitment and take part in the improvement process. Setting goals that will help the people improve their everyday jobs and see the progress that they are making helps to motivate them to carry out the process improvement actions. If the goals are set too far in the future, it becomes very difficult to ever reach the goals. The goals have to be measurable in order for people to know when they have reached them, and reachable for the work to be motivating.

Software companies are highly technical and knowledge-orientated organizations. The people of this kind tend to be motivated by facts. The goals and improvement actions must be based on facts and figures to support the change.

2.5.4.3 Resources and Roles

Perhaps the most difficult issue, especially in small organizations, is to provide enough resources and time to the people doing the process improvement work.

Because of the lack of resources, many people in process improvement programs are working only part-time, so that it can be easy for the process improvement task to be pushed aside and forgotten. If process improvement activities are not tracked and planned they will not happen. People will always be busy doing their "real" jobs.

Humphrey discusses three different roles of the people who are taking part in process improvement programs: champions, sponsors, and agents [12].

Champions are the ones who initiate process change. They bring the management's attention to the subject, obtain the blessing of a sponsor, and establish the credibility to get the program launched.

The role of the senior management is to act as a sponsor. The sponsor has authority capable of providing the resources and official backing up of the process change. When it is clear that the improvements are needed and a sponsor has been found the next step is to identify and motivate the agents who will lead the planning and implementation of the change. They will find the resources, assign the tasks and request the help of management when needed.

The key qualities of agents are: they should be enthusiastic about leading the change process, they should be technologically and politically capable, have the respect of the people they are to deal with and they must have the confidentiality and support of the management [12].

2.5.4.4 Process Improvement Groups

A good way to start the process improvement program is to form dedicated process improvement groups. The goals set for process improvement are assigned to process improvement groups that carry them out. The process improvement group is a group of people that have the required support and knowledge of the problem area to carry out the process improvement actions.

The manager establishing the process improvement groups should take into consideration the motivational needs of the team. This will enhance team's active participation, communication and will ensure that everybody in the group feels comfortable about taking part to the process improvement action. People are best motivated if they are selected to the teams based on their expertise and familiarity with other team members. The team leader also requires enough authority and expertise in the field to be convincing and able to lead the group of people.

This group identifies the key problems, establishes priorities and develops action plans. When the plans are completed, they are presented to the top management as the process improvement proposal to be approved and verified. Management then verifies that the proposal will achieve the goals set for the group.

The next phase is to assign the planned activities to people and allocate the required resources to carry out the action items. Weekly meetings should be held during the process improvement work to track the progress and resolve any problems encountered. This way everybody keeps track of what is going on and everybody is reminded of the tasks they are to carry out. When all the items have been implemented, a smaller group can test the process improvement work. Then the required changes and improvements can be made before the results are made publicly available to improve the current process.

Chapter 3

Software Process Improvement Model

During recent years the competition between different producers has increased. Despite the demanding market situation it is common that budget are exceeded and products are not delivered on time. Within the software development sector three major areas of improvements have been defined namely People, Process and Technology [13] as shown in the figure. Several different models have been developed to measure and improve these areas

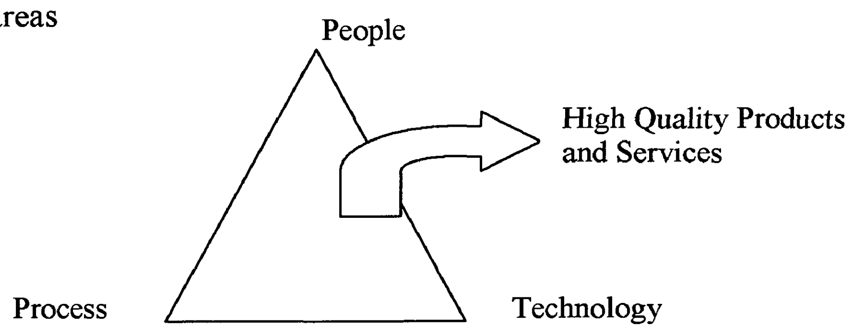


Figure 3.1 Key Components of Software Development

One way to improve productivity and performance is to improve the process used when developing software. Different types of models have been developed to evaluate the working processes used by an organization. The assessment (evaluation) then highlights which areas that need to be improved to achieve higher performance and quality. The benefits of better methods and tools cannot be realized in an undisciplined, chaotic project. In many organizations, projects are often excessively late and double the planned budget, in such instances the organization frequently is not providing the infrastructure and support necessary to help projects avoid these problems.

Even in undisciplined organizations, however, some individual software projects produce excellent results. When such projects succeed, it is generally through the heroic efforts of a dedicated team, rather than through repeating the proven methods of an organization with a mature software process. In the absence of an organization-wide software process, repeating results depends entirely on having the same individuals available for the next project. Success that rests solely on the availability of specific individuals provides no basis for long-term productivity and quality improvement throughout an organization. Continuous improvement can occur only through focused and sustained effort towards building a process infrastructure of effective software engineering and management practices [14].

Software Process Improvement Models provides a basis for orderly exploration as well as a framework for establishing problem priorities. Such a model enables the entire team to work together on the key issues and recommendations. With such a focus, far better conclusion can be reached than would be possible otherwise. An organization with a mature process will take the responsibility of executing its planned commitments.

There are many software process improvement models that exist for software community to improve it process from ad hoc to mature and disciplined one. They are Capability Maturity Model (CMM), Software Process Improvement Capability determination (SPICE), BOOTSTRAP, Trillium, Application of Metrics in Industry (AMI), ISO 9000, and Capability Maturity Model Integration (CMMI). CMM has been widely used in the software industry as compared to other model. Some model are designed to support particular industry for example Trillium was developed for telecommunication oriented software organizations and BOOTSTRAP was mainly developed for software process assessment for use within and among software producing organization in Europe. This section of the thesis discusses in more detail about CMM and CMMI model.

3.1 Capability Maturity Model for Software

What is Capability Maturity Model for Software?

The Capability Maturity Model for Software version 1.1, is developed by the Software Engineering Institute (SEI) for Carnegie Mellon University. The framework of CMM was born in 1986 when the University received a request from the US Department of Defense, DOD. DOD wanted a method for assessing the capability of their software contractors. The original formulation was presented by Watts Humphrey in 1989 and included a framework of process maturity and a questionnaire to aid in appraising maturity [15]. The initial model was further developed through software process assessment, workshop and extensive reviews and resulted in the Capability Maturity Model for Software (SW-CMM) in 1994 [1].

The formal definition of SW-CMM is

"Capability Maturity Model:- A description of the stages through which software organizations evolve as they define, implement, measure, control and improve their software process" [1].

The SW-CMM model provides a roadmap for moving from an ad hoc, chaotic, immature process culture, to a mature organization with a culture of process discipline and continuous process improvement. Practices for the improvement of the ability to meet goals for cost, schedule, functionality and product quality are suggested. The SW-CMM also highlights areas that need to be improved, but not how to do it. The SW-CMM should be seen as a tool to help software organizations improve their software process. The model guides software organizations that want to gain control of their processes for developing and maintaining software and to evolve towards a culture of software engineering and management excellence. The maturity of the organization can then be improved by focusing on a set of activities that impact the software process capability. CMM although used widely has many critiques specially from the small companies.

Brodman and Johnson, [16] conducted research sponsored by the United States Air Force within the Department of Defense software development community to determine the applicability of the CMM to small businesses and small software organizations. They reported critique especially on the issues of separate organizations, specifically for configuration management, software quality assurance, and software engineering process groups. They also pointed out that some KPA practices do not apply to small organizations. CMM addresses practices such as document policies and procedures that large organizations need because of their size and management structure. Small businesses responded that their people communicate verbally on an on-going basis and the required documentation, especially on small two-to-three person projects, would be counterproductive.

Several other models have been developed upon the framework of the CMM. This includes the Software Acquisition Capability Maturity Model (SA-CMM) that focus on the acquirer's role in the software acquisition process, System Engineering Capability Maturity Model (SE-CMM) that describes the essential elements of an organization's system engineering process, and Capability Maturity Model Integration (CMMI) that is currently under development. According to the specification, the CMMI will provide organizations a set of integrated products to support process and product improvement.

3.2 Structure of the SW-CMM

The evolution from an immature to a mature organization is made through continuous improvement. The SW-CMM framework makes the process easier by specifying five different levels of maturity and the practices and processes that should be used to increase the capability of the organization.

At the top level five different maturity levels indicate the process capability of the organization. The process capability indicates the range of results that can be achieved when a certain process is being used and is one way to predict the outcome of a project.

Each maturity level consists of different key process areas that contain a set of goals that need to be satisfied to fulfill the intention of the key process area. The goal summarizes the practices that must be accomplished to meet the key process area and is an indicator of the capability of the organization. Each key process area is organized into five sections called common features. They describe how the implementation of activities should be done and help the organization to institutionalize the practices so that they become effective, repeatable and lasting. The common features contain the key practices. They describe the infrastructure or activities that contribute to the best way of implementation and institutionalization to complete a key process area. The key practices describe what is to be done to improve the process, but not how it should be implemented. This decomposition of CMM makes it possible to identify small areas where improvement can be made. The practical use is that the organization can easily find the key practices for each key process area and each maturity level [1].

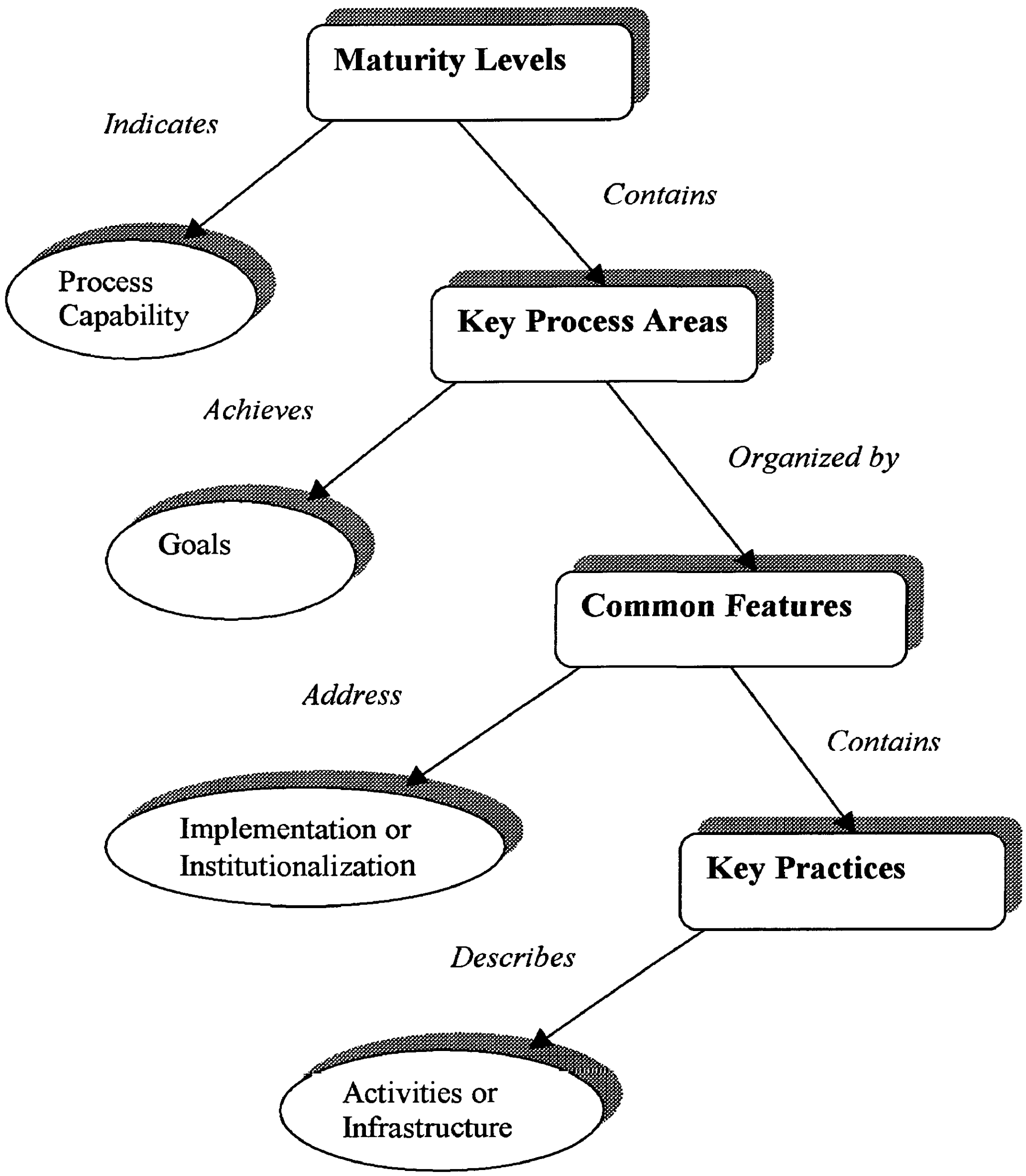


Figure 3.2 Structure of the SW-CMM

3.2.1 The SW-CMM Maturity Levels

A maturity level indicates to which degree an organization has a mature software development process within the organization. At each maturity level a set of process goals are defined that, when satisfied, implement an important part of the software process. At each maturity level different parts and processes are being introduced and established, which lead to continuous improvement of the process capability of the organization. The Maturity level of CMM are as shown in the figure.

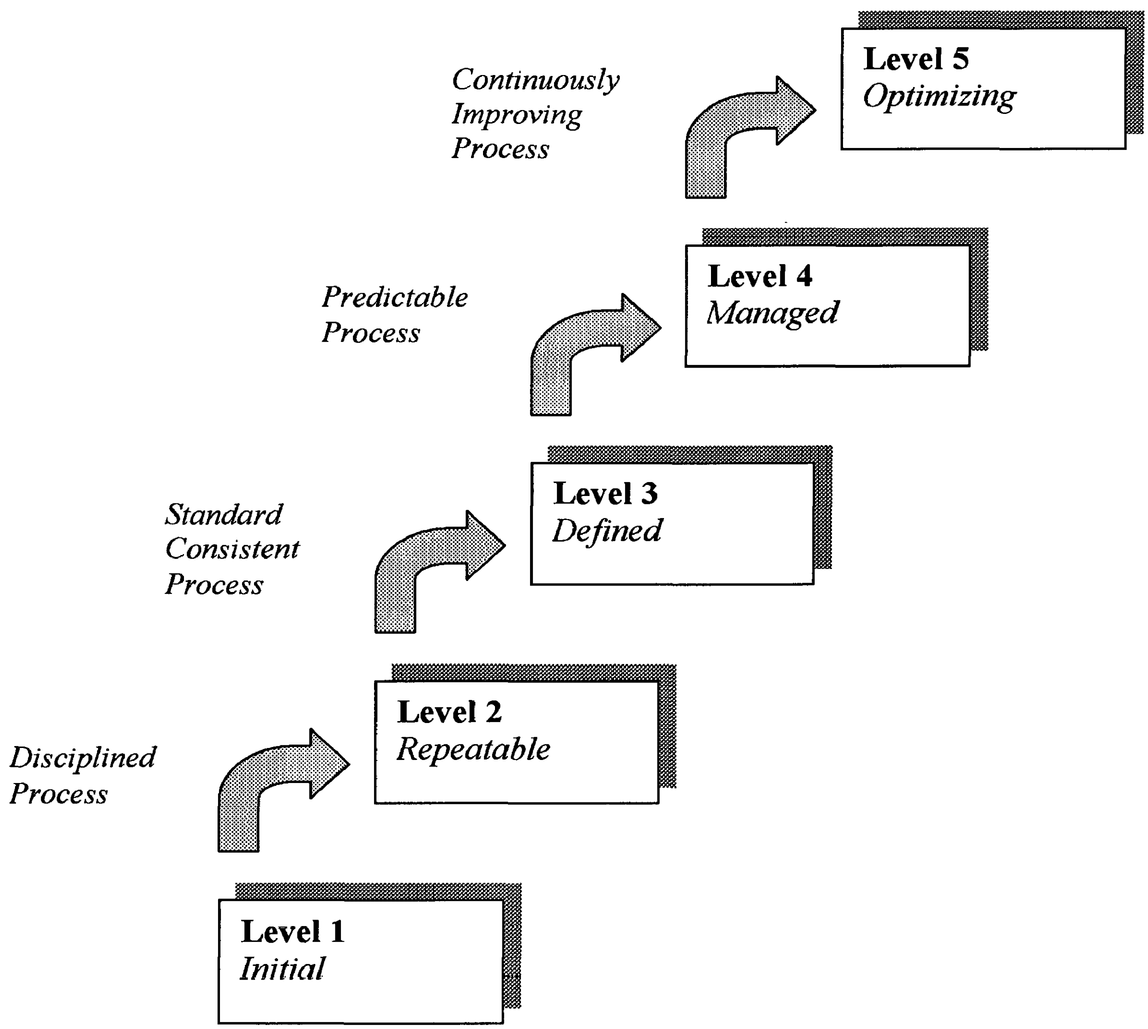


Figure 3.3 Levels of Maturity

The labeled arrows in the figure indicate the type of process capability being institutionalized by the organization at each level of the maturity framework.

With guidance from these levels an organization can prioritize improvement actions to establish and improve their processes. As the maturity level of the organization increases, fundamental changes in processes and quality of the products will occur. The maturity levels are characterized by the process at the present levels.

The maturity levels for SW-CMM as discussed in [1] are:

Level 1- Initial Level:-

The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort and heroics.

Level 2- Repeatable Level:-

Basic processes are established to track cost, schedule and functionality. The necessary process discipline is in place to repeat earlier successes in projects with similar applications.

Level 3- Defined Level:-

The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard process for developing and maintaining software.

Level 4 – Managed Level:-

Detailed measures of the software process and product quality are controlled. Both the software process and products are quantitatively understood and controlled.

Level 5 – Optimizing Level:-

Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

3.2.2 The Sub-Structure of CMM

The maturity levels, with exception of the Initial level, are decomposed into several key process areas the key process areas highlight the areas and the issues that the organization must focus on to reach a higher level of maturity. In order to do this all the key process areas at the desired level and the levels below must be fulfilled and institutionalized.

Each key process areas contain a set of goals and a set of key practices. The goals summarize the key process area and can be used to determine whether the key process area is fulfilled or not. The key practices define a set of activities that need to be accomplished to fulfill the goal of the key process area. However, the key practices only highlight which activities that should be performed, not how they should be carried out.
The key practices of each key process area divided into five common features. The common features indicate whether the implementation of the key process is effective, repeatable and lasting. The common feature activities performed, contains the activities that must be implemented to establish process capability. The other common features include the basic process discipline needed to institutionalize the practices described in activities performed. The common features of the SW-CMM are as follows:

1. Commitment to perform

2. Ability to perform

3. Activities performed

4. Measurement and Analysis

5. Verifying Implementation

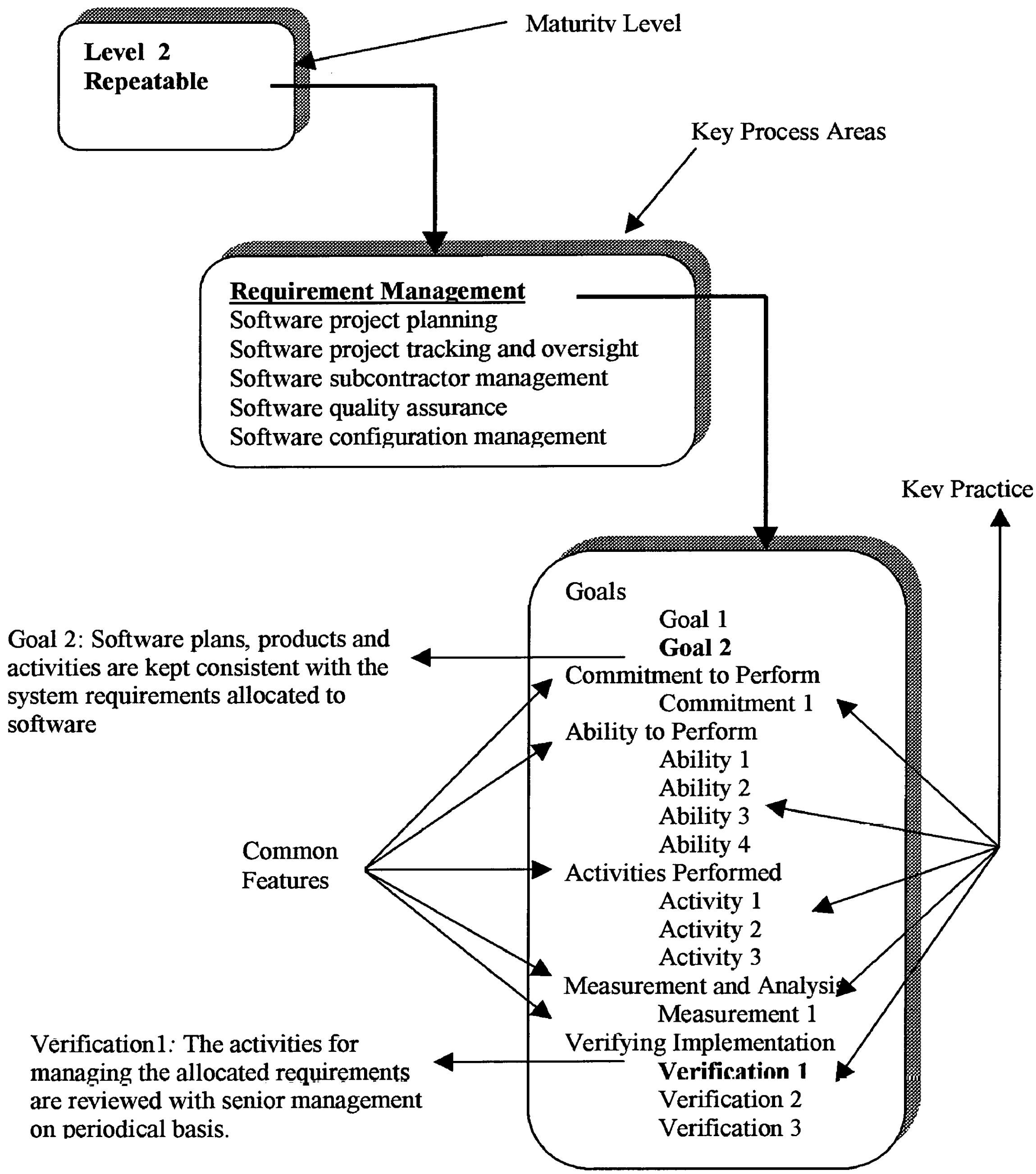


Figure 3.4 Sub-Structure of CMM

3.3 Software Process Appraisal

The CMM uses the process maturity framework as a common frame of reference to find the maturity of an organization. Organization or a project within it can be appraised and found to be at a particular maturity level using the CMM. The CMM describes two different methods, which are to be used for such appraisal. They are Software Process Assessment (SPA), which is also called as CMM Based Appraisal for Internal Process Improvement (CBA-IPI) and Software Capability Evaluation (SCE). These methods differ in who perform the assessment and how it is performed.

3.3.1 CMM based Appraisal for Internal Process Assessment (CBA-IPI)

Software process assessments focus on identifying improvement priorities within an organization's own software process. Assessment teams use the CMM to guide them in identifying and prioritizing findings. These findings, along with guidance provided by the key practices in the CMM, are used (by a software engineering process group, for example) to plan an improvement strategy for the organization.

Software process assessments are performed in an open, collaborative environment. Their success depends on a commitment from both management and the professional staff to improve the organization. The objective is to surface problems and help managers and engineers improve their organization. While the questionnaire is valuable in focusing the assessment team on maturity level issues, the emphasis is on structured and unstructured interviews as tools for understanding the organization's software process. Aside from identifying the software process issues facing the organization, the buy-in to improvement, the organization-wide focus on process, and the motivation and enthusiasm in executing an action plan are the most valuable outcomes of an assessment [14].

3.3.2 Software Capability Evaluation (SCE)

Software capability evaluations are focused on identifying the risks associated with a particular project or contract for building high-quality software on schedule and within budget. During the acquisition process, software capability evaluations may be performed on bidders. The findings of the evaluation, as structured by the CMM, may be used to identify the risks in selecting a particular contractor. Evaluations may also be performed on existing contracts to monitor their process performance, with the intent of identifying potential improvements in the software process of the contractor.

Software capability evaluations, on the other hand, are performed in a more audit-oriented environment. The objective is tied to monetary considerations, since the team's recommendations will help select contractors or set award fees. The emphasis is on a documented audit trail that reveals the software process actually implemented by the organization [14].

3.4 Capability Maturity Model Integration (CMMI)

The purpose of Capability Maturity Model Integration is to provide guidance for improving organizations process and its ability to manage the development, acquisition and maintenance of products and services. CMM integration places proven practices into a structure that helps organization assess its organizational maturity and process area capability, establish priorities for improvement, and guide the implementation of these improvements. The CMMI products suite springs from a framework that generates multiple integrated models, courses, and an assessment method. As new material is added to the framework, more integrated models and supporting material will become available that cover additional disciplines. The CMM integration project was formed to address the problem of having to use multiple capability maturity models. This project work is jointly sponsored by the Office of the Defense, Acquisition, Technology and Logistics (OUSD/AT&L) and the system engineering committee of the National Defense Industrial Association (NDIA) [4,17].

The CMMI effort is intended to support process and product improvement and to reduce redundancy and eliminate inconsistency when using separate stand-alone models. The goal is to improve efficiency, return on investment, and effectiveness by using models that integrate disciplines such as systems engineering and software engineering that are inseparable in a system development endeavor. The concept of the CMMI project was to improve the usability of the CMM technology in a wider set of disciplines beyond its initial success for software engineering alone. The concept called for use of common terminology, common components, and rules for constructing capability maturity model that would be available with a reduction in the amount of training and process improvement effort needed by users of multiple disciplines [4].

3.5 Need for Transition to CMMI

Some of the reasons for transitioning to CMMI are as follows:

1. ***Problems with using current CMMs***

   The SW-CMM is a roadmap that describes evolutionary stages consisting of key practices that guide organization in improving their software capability. Also, several systems engineering maturity models support system engineering process improvement. All of the system engineering models share many of the same principles as the SW-CMM, but were written to address the needs of the system engineering community. This had two consequences. First, the content of SW-CMM and the system engineering models overlaps; for example, all deal with requirements, project management, process definition etc. The different models provide somewhat different guidance in practice where they overlap, but the reason for the difference isn't always clear. Second, the system engineering models are based on a different representation than the SW-CMM. This representation describes the entire process area terrain with less emphasis on exactly how an organization might mature through terrain. Process area span levels rather than being defined within a maturity level as in the SW-CMM. The system engineering models employs what is termed a continuous representation; that is, capability levels for each process area are described independently of others. The SW-CMM employs a staged representation; that is, process areas are grouped into collection and aligned with maturity levels. Finally improvement efforts based on more than one unique CMM would likely result in sub-optimization, confusion, and potentially unnecessary expenditure of process improvements resources.

2. *OSD Perspective on need for change*

The OSD rationale for change was provided by the Office of the Secretary of Defense/Acquisition and Technology (OSD/A&T). With the various models for software engineering and system engineering containing common process, it was recognized that improvements made to one discipline could benefit the other. Also, assessments made for one discipline could be used for other discipline assessments thus eliminating redundant assessments.

3. *Solution to Problem*

Several events in combination made it evident that the time was right to begin developing an integrated CMM framework. These occurrences included the following:

- A first step had already been taken to merge the two existing systems engineering models (the SE-CMM and the SECAM into EIA/IS-731).
- A major update to the CMM (version 2.0) was nearing completion.
- The proliferation of CMM was escalating.

- Organizations operating in more than one discipline were becoming acutely aware of the problems of trying to improve integrated processes using separate, sometime inconsistent CMMs.
- Sufficient experience in developing CMMs and abundant experience in using CMMs increased the likelihood that an integrated framework for a family of CMMs could be developed .

Thus a requirement was formulated for Capability Maturity Model-Integrated (CMMI) product suite. The suite would include a framework for generating CMMI products. The generated products would be based on CMMI models for specified disciplines and discipline combinations, training products, assessment materials, glossary terms, and tailoring requirements. The disciplines initially specified include System Engineering, Software Engineering, and Integrated Product and Process Development (IPPD).

The CMMI includes a common set of process areas which forms the core of an integrated capability model that integrates process improvement guidance for system engineering, software engineering, and Integrated product and process development. The resulting integrated capability models will be tailorable to an organization's mission and business objectives.

3.6 CMMI Framework

The CMMI framework is one part of what is called the "CMMI product suite". The purpose of the CMMI product suite is to serve the user of capability models (CMs) and capability maturity models (CMMs) better than the independently created CMs and CMMs now available. The framework's integrated approach will simplify the process of understanding and using multiple models and provide integrated and tailorable process improvement tools for the development user community [17]. The vision of the CMMI product suite is that a framework user can generate capability models and their supporting training and assessment materials as needed from the framework's common elements and discipline-specific elements. In other words, an organization could streamline its software engineering, system engineering, and integrated product and process development, capabilities by using the CMMI product suite to merge them into one capability model supported by one set of training material and assessment materials. By eliminating unnecessary duplication of common activities, the job of training people to use the model will be simpler. The common terminology used in the models, training materials, and assessment methods will simplify the introduction of process improvement activities based on these models, thereby reducing the cost of adoption.

3.6.1 The Structure of the Framework

The CMMI framework is designed to provide an internally consistent set of common elements that apply to any discipline and that must be included in any CMMI products. These CMMI products will support process improvement activities, including assessments and training. The CMMI framework currently consists of four parts namely, the input process, repository, control process, and output process.

The repository contains the components of capability models, training materials, as well as construction rules and the conceptual architecture of the outputs. The control process sorts, combines and arranges the inputs, the rules for generating capability models, assessment materials, and the training materials to produce appropriate capability model that can be applied to an organization's process improvement efforts [17].

Figure below is the framework of CMMI.

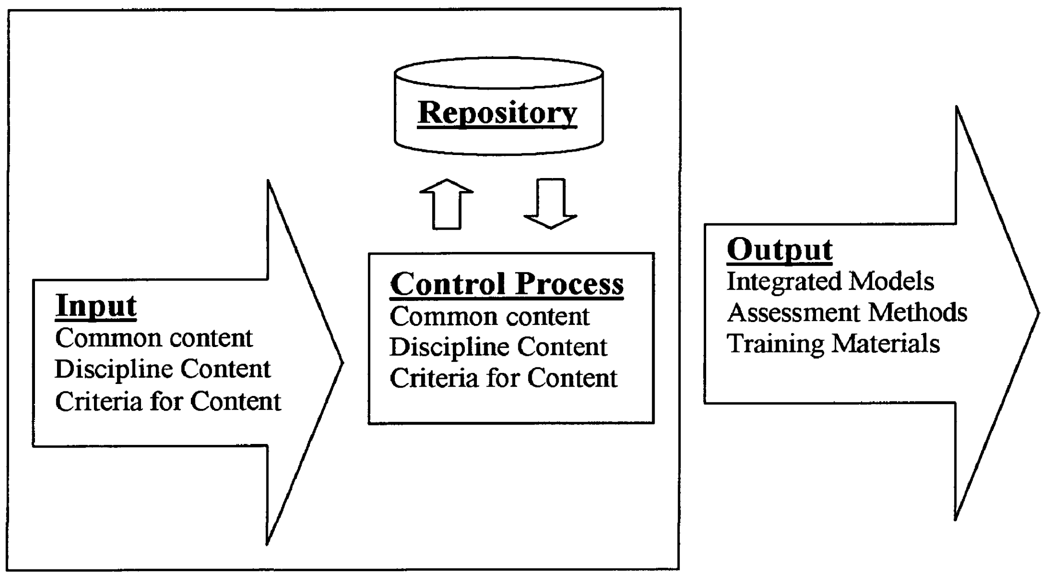


Figure 3.5 CMMI Framework

The framework can be considered as capability model generator. As a user of the CMMI framework, one can specify various options based on the needs of the organization, such as disciplines that need to be covered, and staged vs. continuous representation. The framework then generates the capability model that best meet the organization needs based on the selection of the model. Figure below shows the information that resides in the framework and the processing done to produce tailored capability model [17].

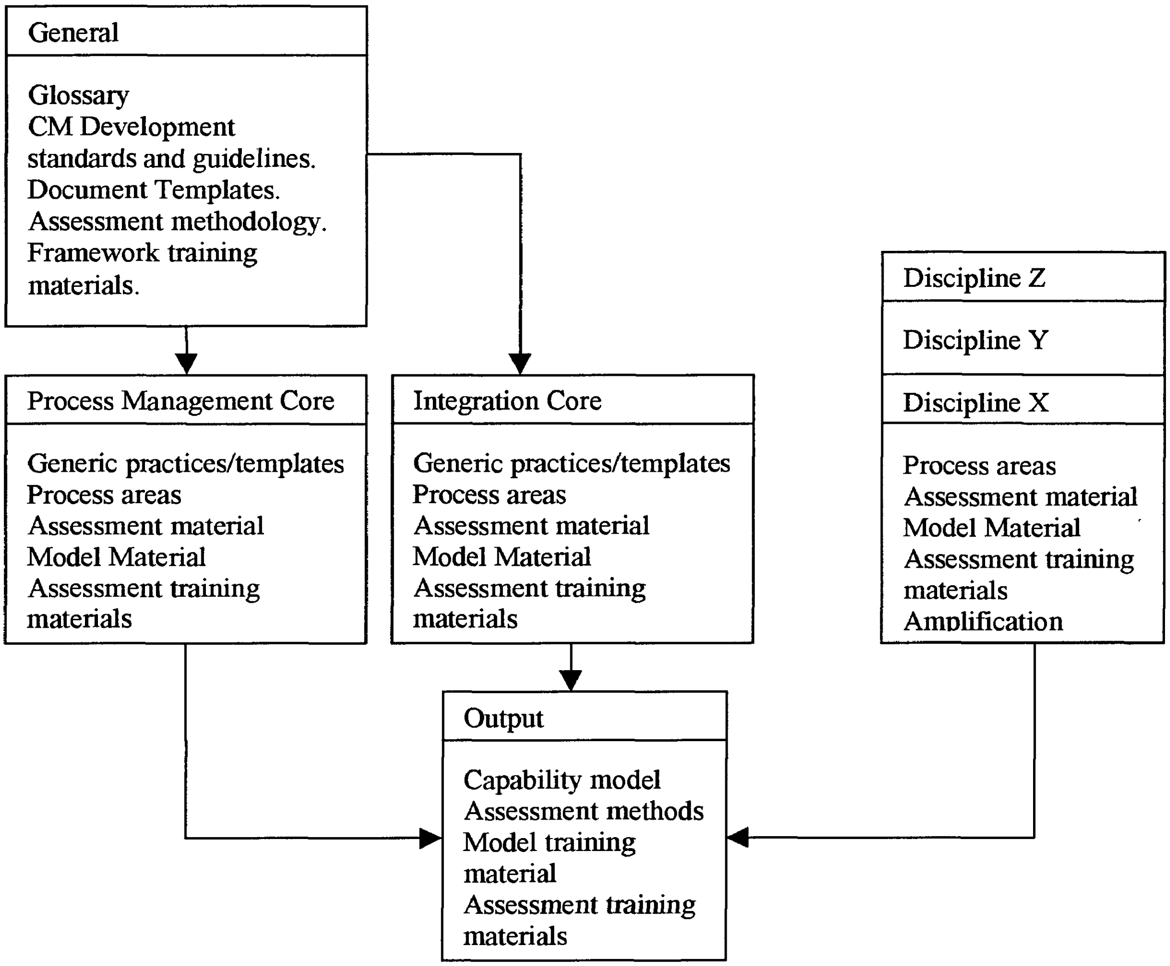


Figure 3.6 Processing of the CMMI Frameworks

The process management core contains process management components that apply to all disciplines and all domains. These components area automatically included in the capability model that is being considered for any particular organization that wishes to develop the model irrespective of the disciplines.

The integration core contains information about Integrated Product and Process Development (IPPD), which can be applied in virtually any discipline or domain. IPPD is a management technique that integrates all development activities ranging from product concept to product support. The IPPD approach uses multifunctional teams, called integrated product teams (IPTs), to improve the product and its development and sustainment processes. The goal of this improvement is to meet the organization's cost and performance objectives. IPPD evolved from concurrent engineering and is sometimes called integrated product development (IPD) [17].

The disciplines represent specific information that can be selected to include in the capability model. The initial CMMI product suite will include two disciplines software engineering, and system engineering, there is also third discipline called Integrated Product and Process Development, which is being worked, but not yet completed. There are plans to include other disciplines but they are waiting for the successful implementation of the current released disciplines namely software and system engineering. The framework is designed in such a way that new disciplines can be added over time as the user community feels the need for it. The framework basically sorts, combines and arranges information to make it more useful to the organization and to tailor the information to the organization needs.

3.7 Selecting CMMI Model

There are multiple CMMI models available, as generated from the CMMI framework, to choose the right model depend on the organization requirement. If an organization is concerned exclusively with software engineering or exclusively with system engineering activities, then the appropriate models would be the CMMI-SW or

CMMI-SE, respectively. However, if an organization were concerned with both systems engineering and software engineering, then using a combined CMMI-SE/SW model would be more appropriate, since it would encourage improvement of integrated practices, reducing the repetition and administrative burden that is common to maintaining separated disciplines. Finally if an organization were employing Integrated Product and Process Development in their practices, then using a model that includes IPPD would be appropriate. Each model that result from the CMMI framework has two representation namely Staged and Continuous representation. The organization has to select the representation that best meet the industry requirement. Usually the industry selects the representation that is most familiar with because it helps to understand the model very easily. In every process improvement model, regardless of representation, the basic building blocks are process areas. How these process areas are presented in the model can be considered its representation. Each representation has its own advantages and disadvantages in the development community.

3.7.1 Staged Representation

In staged representation, process areas are grouped into stages or maturity levels. Each process area contains practices that, when performed, achieves the purpose of the process area. Within a stage, the institutionalization practices for all constituent process areas must be achieved to successfully achieve the entire stage. Once an organization has achieved the entire stage, it has reached a capability maturity level. The software engineering institute's CMM for software

SW-CMM is an example of a staged model [17]. There are five maturity levels, numbered 1 through 5.

3.7.2 Continuous Representation

In a continuous representation, process areas also contain practices that when performed; achieve the purpose of the process area. Generic practices are grouped into capability levels. These practices are added to the practices of each process area to attain a capability level for each process area. Capability levels are achieved process area by process area. Although the order in which process areas are addressed is not required to follow a particular sequence, the order may follow recommended staging [17]. The Electronic Industries Alliance's Interim Standard 731, and system engineering capability model is an example of a continuous model. There area six capability levels, numbered 0 through 5.

Following comparison gives the advantages of using each model [19]

Continuous Representation	**Staged Representation**
Grants explicit freedom to select the order of improvement that best meets the organization's business objectives and mitigates the organization's areas of risk.	Introduce a sequence of improvements, beginning with basic management practices and progressing through a predefined and proven path of successive levels, each serving as a foundation for the next.
Enables increased visibility into the capability achieved within each individual process area.	Visibility is primarily at the maturity level with limited visibility at the process area level.

Because capability levels are measured by process area, comparisons across and among organization can only be made on a process area by process area basis.	Permits easy comparison across and among organization because process improvement results are summarized as a single maturity level number.
Reflects a newer approach that does not yet have the data to demonstrate its ties to return on investment.	Builds on a relatively long history of use that includes case studies and data that demonstrate proven return on investment.
Provide an easy migration from EIA/IS 731 to CMMI	Provide an easy migration from the SW-CMM to CMMI

Table 3.1 Comparisons of Staged and Continuous Representation

As there are different models developed from the CMMI framework, and more discipline will be added; this thesis will discuss the CMMI model for systems engineering/software engineering in detail. In particular the focus will be more on staged representation. The CMMI model for systems engineering/software engineering (CMMI-SE/SW) consists of the same process areas, regardless of representation (continuous or staged).

The systems engineering discipline covers the development of total systems, which may or may not include software. Systems engineers focus on transforming customer needs, expectations, and constraints into product solutions and supporting those product solutions throughout the product life cycle. The software engineering discipline covers the development of software systems. Software engineers focus on applying systematic, disciplined, and quantifiable approaches to the development, operation, and maintenance of software.

3.8 CMMI-SE/SW Model for Continuous Representation

3.8.1 Structure of the model

CMMI models are designed to describe discrete levels of process improvement. In the continuous representation, capability levels provide a recommended order for approaching process improvement within each process area. At the same time, the continuous representation allows some flexibility for the order in which the process areas are addressed. All continuous representations of CMMI models reflect capability levels in their design and content. A capability level consists of related specific and generic practices for a process area that, when performed, increase the capability of the organization in that process area and enhance the organization's overall process capability [20].

Capability levels of the continuous representation focus on maturing the organization's ability to perform, control, and improve its performance in a process area. These levels enables organization to track, evaluate, and demonstrate progress as organization improves processes associated with process areas.

Capability levels are determined by reviewing the organization's implementation of the specific and generic practices and its achievement of the associated goals through that capability level. For example, to achieve capability level 2 for a process area, the organization's activities are reviewed against the specific and generic practices and goals through capability level 2. The specific and generic goals through capability level 2 must be satisfied. A process area that does not satisfy all of the requirements for capability level 1 is said to be at level 0. As organization achieves the generic and specific goals for a process area at a particular capability level, then it is increasing its process capability and reaping the benefits of process improvement. The generic goals and practices define a sequence of capability levels, which represent improvements in the implementation and effectiveness of the processes. Specific goals and specific practices apply to individual process areas. Generic goals and generic practices apply to multiple process areas. There are six capability levels numbered 0 through 5 they are as follows [20].

3.8.2 Capability Levels

Capability Level 0: Incomplete

A process that is considered incomplete does not implement all of the capability level 1 specific and generic practices.

Capability Level 1: Performed

A performed process is a process that is expected to perform all of the capability level 1 specific and generic practices. Performance may not be stable and may not meet specific objectives such as quality, cost, and schedule, but useful work can be done.

Capability Level 2: Managed

A capability level 2 process is a managed process. A managed process is planned, performed, monitored, and controlled for individual projects, groups, or stand-alone processes to achieve a given purpose. Managing the process achieves both the model objectives for the process as well as other objectives, such as cost, schedule, and quality.

Capability Level 3: Defined

A capability level 3 process is a defined process. A defined process is a managed process that is tailored from the organization's set of standard processes. Deviations beyond those allowed by the tailoring guidelines are documented, justified, reviewed, and approved.

Capability Level 4: Quantitatively Managed

A capability level 4 process is a quantitatively managed process. A quantitatively managed process is a defined process that is controlled using statistical and other quantitative techniques. Product quality, service quality, process performance, and other business objectives are understood in statistical terms and are controlled throughout the life cycle.

Capability Level 5: Optimizing

A capability level 5 process is an optimizing process. An optimizing process is a quantitatively managed process that is improved based on an understanding of the common causes of process variation inherent in the process. An optimizing process focuses on continually improving process performance through both incremental and innovative improvements. Both the defined processes and the organization's set of standard processes are targets of the improvement activities.

3.8.3 Components Of CMMI Model

Figure below shows the major components of the continuous representation of each CMMI model [20].

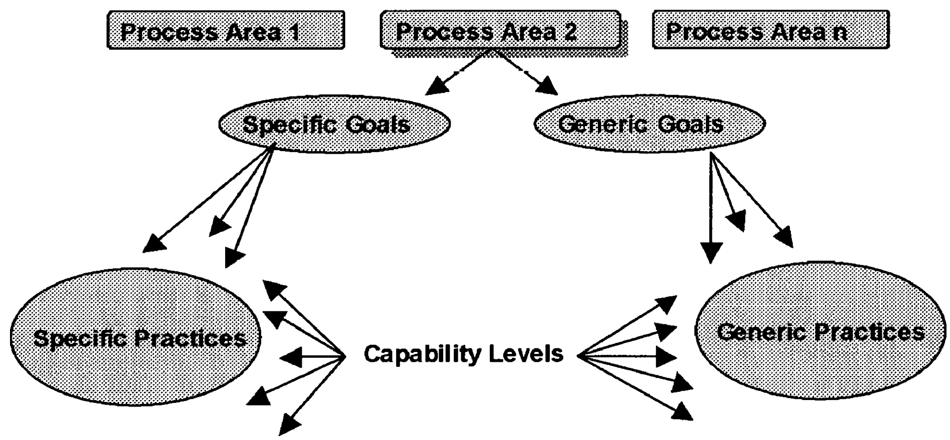


Figure 3.7 Component of the Continuous Representation

Process Areas:

A process area is a group of related practices that are performed collectively to achieve a set of objectives, including what it does (specific practices) and the anticipated behavior (specific goals). All CMMI process areas are common to both continuous and staged representations.

Generic Goals:

Each capability level has only one generic goal that describes what the organization must achieve at that capability level. Achievement of each of these goals relative to a process area signifies improved control in performing the process area.

Generic Practices:

Generic practices are practices that apply to any process area because they can improve the performance and control of any process. Generic practices are categorized by capability level and are expected components in the model. In the continuous representation, each generic practice maps to one generic goal.

Specific Goals:

Specific goals apply to only one process area and address the unique characteristics that describe what must be implemented to satisfy the purpose of the process area. Goals are required model components and are used in assessments to determine whether a process area is satisfied. There can be specific practices at different capability levels mapped to the same goal. However, every goal has at least one capability level practice mapped to it.

Specific Practice:

A specific practice is an activity that is considered important in achieving the specific goal that it is mapped to. The specific practices describe the activities expected to result in achievement of the specific goal of a process area. Every specific practice is associated with a capability level.

Base Practices:

The specific practices in the continuous representation that are at a capability level of 1 are called base practices. These practices are considered essential in achieving the purpose of the process area to which it belongs.

Advanced Practices:

Some specific practices in the continuous representation are at a capability level higher than 1. These practices are called advanced practices.

3.9 CMMI - SE/SW Model for Staged Representation

The staged representation organizes process areas into maturity levels to support and guide process improvement. The staged representation uses four common features to organize the generic practices in the process areas. CMMI models are designed to describe discrete levels of process improvement. In the staged representation, maturity levels provide a recommended order for approaching process improvement in stages so that not all process areas are addressed at the same time. Within the process areas are generic and specific goals as well as generic and specific practices. The maturity level of an organization provides a way to predict the future performance of an organization within a given discipline or set of disciplines. Experience has shown that organizations do their best when they focus on a manageable number of process areas that require increasingly sophisticated effort as the organization improves. Each maturity level stabilizes an important part of the organization's processes. Achieving each maturity level results in an increase in the process capability of the organization [19].

There are five maturity levels they are as follows:

- Initial
- Managed
- Defined
- Quantitatively Managed
- Optimizing

Maturity levels are measured by the achievement of the specific and generic goals that apply to a predefined set of process areas. As the organization achieves the generic and specific goals for the set of process areas in a maturity level, then it increases its process maturity and benefits from process improvement. Maturity level is one means of predicting the most likely outcomes from the next project the organization undertakes [19]. The staged representation identifies the maturity levels through which an organization should evolve to establish a culture of excellence. Because each maturity level forms a necessary foundation on which to build the next level, trying to skip maturity levels is not a good idea and is usually counterproductive.

Organizations can institute specific process improvements at any time they choose, even before they are prepared to advance to the maturity level at which the specific practice is recommended. However, organizations should understand that the stability of these improvements is at a greater risk since the foundation for their successful institutionalization is not been completed. Processes without the proper foundation may fail [19].

3.9.1 Components of the Model

Figure below shows the major components of the staged representation of each CMMI model [19].

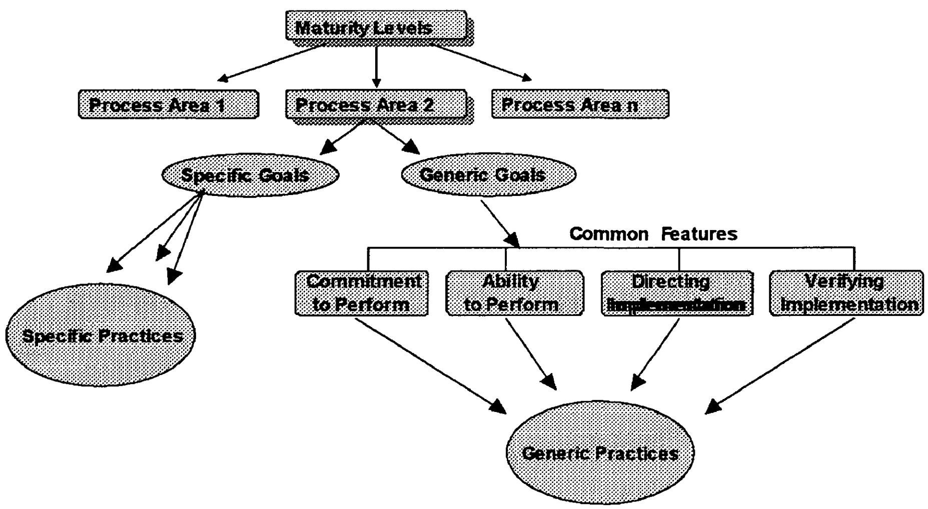


Figure 3.8 Component of Staged Representation

The major components of the staged representation are Maturity level, process areas, specific goals, specific practices, generic goals, generic practices. All the components are similar to the one in continuous representation. Common feature component is different from continuous representation; four common features organize the generic practices of each process area. Common feature names are model components that are informative. They are only groupings that provide a way to present the generic practices. Common features are predefined attributes that group generic goals and generic practices into categories. Common features are model components that are not rated in any way. There are four common features used in CMMI models: Commitment to

Perform, Ability to Perform, Directing Implementation, and Verifying Implementation [19].

- Commitment to Perform groups all generic practices related to creating policies and securing sponsorship for process improvement efforts.
- Ability to Perform groups all generic practices related to ensuring that the project and/or organization has the resources it needs to pursue process improvement.
- Directing Implementation groups the generic practices related to collecting, measuring, and analyzing data related to processes. The purpose of these activities is to provide insight into the performance of processes.
- Verifying Implementation groups all generic practices related to verifying that the projects and/or organization's activities conform to requirements, processes, and procedures.

3.9.2 Maturity Levels

In the staged representation, maturity levels allow organizations to focus improvement efforts on the critical processes that will have the most benefit to the organization. There are five maturity levels as described below [19].

Maturity Level 1 - Initial

A maturity level 1 is typically ad hoc and chaotic. The organization typically does not provide a stable environment. Success in these organizations depends on the competence and heroics of the people in the organization and cannot be repeated unless the same competent and experienced individuals are assigned to the next project. In spite of this ad hoc, chaotic environment, maturity level 1 organizations frequently produce products that work; however, they often greatly exceed the budget and schedule of the project.

Maturity Level 2 – Managed

At maturity level 2, an organization has achieved all of the goals of the maturity level 2 process areas. In other words, the organization has ensured that its processes are planned, documented, performed, monitored, and controlled at the project level; at maturity level 2, objectives established for the process, such as cost, schedule, and quality objectives are also achieved.

Maturity Level 3 – Defined

At maturity level 3, an organization has achieved all of the goals of the maturity level 2 and 3 process areas. Processes are tailored from the organization's set of standard processes and related organizational process assets to suit the circumstances in which they will be performed. At maturity level 3, processes are well characterized and understood, and are described in standards, procedures, tools, and methods. At maturity level 3, processes are described in more detail and more rigorously than at maturity level 2. At maturity level 3, processes are managed more proactively using an understanding of the interrelationships of the process activities and detailed measures of the process, its work products, and its services.

Maturity Level 4 – Quantitatively Managed

At maturity level 4, an organization has achieved all of the goals of the maturity level 2, 3, and 4 process areas. Processes are controlled using statistical and other quantitative techniques. Quantitative objectives for product quality, service quality, and process performance are established and used as criteria in managing processes. Product quality, service quality, and process performance are understood in statistical terms and are managed throughout the life of processes.

Maturity Level 5 – Optimizing

This is the highest level that an organization can achieve. At maturity level 5, an organization has achieved all of the goals of the maturity level 2, 3, 4, and 5 process areas. Processes are continually improved based on an understanding of the common causes of variation inherent in processes. Maturity level 5 focuses on continually improving process performance through both incremental and innovative technological improvements. Quantitative process improvement objectives for the organization are established, continually revised to reflect changing business objectives, and used as criteria in managing process improvement. Selected incremental and innovative technological process improvements are deployed into the organization systematically. The effects of the deployed process improvements are measured and evaluated against the quantitative process improvement objectives.

3.9.3 Process Areas

There are no process areas at maturity level 1since it is ad hoc and chaotic; the process areas for different maturity levels are as follows [19]:

Process Areas at Maturity Level 2 are

- *Requirements Management:* Requirements Management manages the requirements of the project's product and product components and identifies inconsistencies between the project's plans and work products and the requirements.
- *Project Planning:* Project Planning establishes and maintains plans that define project activities.
- *Project Monitoring and Control*: Project Monitoring and Control provides understanding into the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.
- *Supplier Agreement Management*: Supplier Agreement Management manages the acquisition of products and services from suppliers external to the project for which there exists a formal agreement.
- *Measurement and Analysis*: Measurement and Analysis develops and sustains a measurement capability that is used to support management information needs.
- *Process and Product Quality Assurance*: Process and Product Quality Assurance provides staff and management with objective insight into the processes and associated work products.

- *Configuration Management:* Configuration Management establishes and maintains the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.

Process areas at Maturity Level 3 are

- *Requirements Development*: Requirements Development produces customer, product, and product component requirements and analyses required for their development and understanding.
- *Technical Solution:* Technical Solution develops, designs, and implements solutions to requirements. Solutions, designs and implementations encompass products, product components, and product related processes either singly or in combinations as appropriate.
- *Product Integration*: Product Integration assembles the product from the product components, ensures that the product, as integrated, functions properly, and delivers the product.
- *Verification*: Verification assures that selected work products meet their specified requirements.
- *Validation:* Validation demonstrates that a product or product component fulfills its intended use when placed in its intended environment.
- *Organizational Process*: Organizational Process establishes and maintains an understanding of the organization's processes and process assets, and identifies, plans, and implements the organization's process improvement activities.

- *Organizational Process Definition:* Organizational Process Definition establishes and maintains a usable set of organizational process assets.
- *Organizational Training*: Organizational Training develops the skills and knowledge of people so they can perform their roles effectively and efficiently.
- *Integrated Project Management*: Integrated Project Management establishes and manages the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes.
- *Risk Management*: Risk Management identifies potential problems before they occur, so that risk handling activities may be planned and invoked as needed across the lifecycle to mitigate adverse impacts on achieving objectives.
- *Decision Analysis and Resolution:* Decision Analysis and Resolution makes decisions using a structured approach that evaluates identified alternatives against established criteria.

The process Areas at Maturity Level 4 are:

- *Organizational Process Performance*: Organizational Process Performance establishes and maintains a quantitative understanding of the performance of the organization's set of standard processes and provides the process performance data, baselines, and models to quantitatively manage the organization's projects.
- *Quantitative Project Management*: Quantitative Project Management quantitatively manages the project's defined process to achieve the project's established quality and process performance objectives.

The process Areas at Maturity Level 5 are:

- *Organizational Innovation and Deployment*: Organizational Innovation and Deployment selects and deploys incremental and innovative improvements that measurably improve the organization's processes and technologies. The improvements support the organization's quality and process performance objectives as derived from the organization's business objectives.
- *Causal Analysis and Resolution:* Causal Analysis and Resolution identifies causes of defects and other problems and takes action to prevent them from occurring in the future.

3.10 CMMI Assessment Methods

Process assessments focus on identifying improvement opportunities within an organization. Assessment teams use CMMI models, including the Capability Maturity Model –Integrated for Systems Engineering and Software Engineering (CMMI-SE/SW) model, to guide them in identifying and prioritizing findings. These findings, combined with guidance provided by the practices in the CMMI model, are used (by an engineering process group, for example) to plan an improvement strategy for the organization.

For organizations that wish to assess against multiple disciplines (e.g., software engineering and system engineering), the unified CMMI approach permits some economy of scale in model and assessment training. One assessment method can provide separate or combined results for multiple disciplines, while assessment of a single discipline is also supported. CMMI assessment products provide consistent findings for both staged and continuous representations through the use of equivalent staging [21].

The Assessment Requirements for CMMI (ARC) comprise a set of high-level design criteria for developing, defining, and using assessment methods based on CMMI models. The Assessment Requirements for CMMI defines the requirements considered essential to assessment methods intended for use with CMMI models. In addition, a set of assessment classes is defined based on assessment usage scenarios. These classes are intended primarily for developers of assessment methods to use with CMMI capability models in the context of the CMMI Product Suite. The approach employed to provide guidance to assessment method developers is to define a class of assessment method usage scenarios (which are based on years of experience in the process improvement community) called assessment classes. Requirements are then allocated to each class as appropriate based on the attributes associated with that class. Thus, a particular assessment method may declare itself to be an ARC class A, B, or C assessment method. This designation implies the sets of ARC requirements, which the method developer has considered when designing the method. Assessment methods, which satisfy all of the ARC requirements are called, class A methods [22]. The ARC requirements are designed to help improve consistency across multiple disciplines and assessment methods and to help assessment method developers, sponsors, and users understand the tradeoffs associated with various methods.

3.10.1 Standard CMMI Assessment Method for Process Improvement (SCAMPI)

The SCAMPI method is a Class A method that complies with all of the ARC requirements. The method enables an organization to do the following [21]:

- Gain insight into its development capability by identifying the strengths and Weaknesses of its current processes.
- Relate these strengths and weaknesses to the CMMI model.
- Prioritize improvement plans.
- Focus on improvements that are most beneficial, given its current level of organizational maturity or process capabilities and considering its business goals.
- Derive capability level ratings as well as a maturity level rating.

The approach of the SCAMPI method is to assemble and train a competent assessment team under the leadership of a Lead Assessor and to conduct a structured series of data gathering and analysis activities with people in the organization to understand their problems, concerns, and ideas for improvement. The method is enacted by a trained group of professionals who work as a team to generate findings and, optionally, ratings relative to the CMMI model process areas within the assessment scope. The findings are generated from data collected from questionnaires, document review, and in-depth interviews with organization members involved in the enactment or management of the processes being examined. The SCAMPI method has two primary objectives:

1. To support, enable, and encourage an organization's commitment to process improvement
2. To provide an accurate picture of the strengths and weaknesses of the organization's current process, using the CMMI model as a reference model, and to identify areas for improvement.

A primary objective of the SCAMPI method is to build on an organization's commitment, which was established during previous phases of the process improvement cycle. The other primary objective of the SCAMPI method is to provide an accurate picture of existing processes relative to the assessment reference model [21].

The SCAMPI method has three phases. The first phase includes the activities necessary to plan and prepare for the assessment. The second phase consists of onsite activities for conducting the assessment, including techniques for gathering, organizing, and consolidating data. The final phase is to report the results.

3.11 Method of Adding new Discipline to CMMI

The CMMI product suite has been developed to provide a framework that contains process areas, common across disciplines, and is structured to readily add new disciplines in the future. Products for new disciplines will require a discipline sponsor or recognized authority for that discipline who represents the knowledge base and desires of the community associated with the discipline. This discipline sponsor or recognized authority will be responsible for bringing forward the product need to the OSD or NDIA sponsors or to the Steering Group. Assuming the need can be established and it fits within the charter of the CMMI project, effort will be initiated to incorporate the new discipline into the CMMI product suite.

3.12 Failure of CMM among Small organizations

The existing CMM models are not directly applicable for small organizations for various reasons. CMM, for example, proposes more than 25 organizational roles, with various tasks and responsibilities. In a small organization, there are not enough people to fill the roles proposed, neither is there any need of many of those roles. Further, the models are usually described on a huge number of text pages, being cumbersome and time consuming to get oversight, comprehend, and apply. Therefore, the models need to be scaled down to the needs and possibilities of small organizations. A usual restriction in a small organization is that there are not enough resources for appointing external competence for a long term SPI, but local competence is the only realistic possibility. Therefore, the models should include guidelines for internal assessment and application of the model [35]. In addition, process appraisal methods such as CBA-IPI and SCE may seem too expensive or time-consuming for a smaller organization seeking to kick off a process improvement program. Small Organizations and small projects typically have the following issues with CMM:

- It is too complex for small organizations and projects.
- It requires steep initial investment of resources (people, time and money).
- It is difficult to tailor as per the organization need.
- It results in too much documentation.

- It is too rigid and structured.
- No immediate payoff.
- CMM describe what an organization should have or what organization need to do, but does not say how to get there or how to do it.
- DoD, which is a major investor in the software, requires at least level 3 compliance for contractors. Private industries are likely to follow the model, which may lead to political problems by creating competition among the contractors.
- All the lead assessors and the team members of the model are based in the United States. So small organization outside United Stated may be reluctant to make transition, as they have to incur additional expenses for assessment. This is not a major problem for big organization, but might be for small organizations.

3.13 CMMI for small organizations

The application of the CMMI to a small team must be carried out sensibly by permanently adapting to the environment of the team, using just the key practices that are really important to the process, and keeping in mind that team management is just as important as process management. Tailoring the model is a process whereby only a subset of the model is used to make it suitable for a specific application. Different types of projects or different applications appear to require different interpretations and or tailoring of the practices, even the similar types of projects or similar applications do. The cause of this is that some organizational structures seem to be more compatible with the key practices of the SPI model they are using than other. Almost every organization or project will have to perform some tailoring or interpretation in order to apply the key practices in their specific environment. The prerequisite to using the key practices of the CMMI as process requirement is that an organization must determine the similarities and difference between the environment expressed in the terminology of the CMMI and the organizations environment. This analysis is a very important input into the organizations process activity [36]. The CMMI models were written for use by all types of organizations; however, for small organizations a CMMI model must be interpreted.

In small projects, meetings take place more frequently, take less time, and cover more details. The schedule may contain daily activities, and may be monitored in weekly meetings. The schedule may change weekly. A configuration management function keeps every version of the schedule in the project library.

In a small team, the customer usually knows the entire team and feels comfortable calling any member of the team to propose or discuss a change. The team must decide up front how to handle these informal calls from the customer. Once they have decided on an approach, it should be captured in the project plan details, and communicated to the customer.

The work of a small team may be highly collaborative; thus, a formal peer review may not provide a high return on investment. The checklist for the review by a peer is just as comprehensive in this small team approach as it would be for a larger team. All of the standards are enforced by all of the members of the team.

Periodically, reviews of the project plans and lessons learned may be funneled to a higher-level of the organization. This review ensures that the higher-level documentation and direction is continually improved. Best business practices are identified and fed back into the organization's process asset library, and the organizational processes, plans, and templates are modified to reflect the improvements used by the project. The next time the project begins work with a new set of requirements, it tailors the updated organizational assets [19].

Small organizations and projects can satisfy the goals of CMMI by using the following guidelines.

- ✓ Using shortcuts such as templates and checklist for documentation.
- ✓ Using alternative methods such as spot-check, and resource sharing for activities.
- ✓ Using combined roles for agents of activities.
- ✓ Using manual methods or basics tools, such as estimation techniques and spreadsheets in place of higher priced automated tools.

The CMMI model is mostly based on the CMM model and has additional process areas and functionality. As a matter of fact, CMMI model will also face the same criticism from the industry as CMM did. These drawbacks will definitely pull the attentions among the organization's SPI expert. This might slow down the popularity of integrated model at earlier stages until the proven data are available from the industry.

Chapter 4

Comparison and Evaluation of the Integrated Model

The CMMI model for process improvement was released in August 2000 as discussed in the previous sections. The model being fairly new no data are available for the organizations to compare its strength and stability. This section provides a brief comparison between the currently used model CMM and the new integrated model CMMI, based on the different criteria. This section provides some useful recommendations that have to be considered in making the transition and the conditions that are favorable for making the transitions and the conditions that are not favorable for transition. Every organization that wish to make the transition have to consider these issues, as they are vital for the organizations process improvement, further growth, competition among the organizations and success.

4.1 Comparison between CMM and CMMI

1. Evolution of the Model

<u>CMM:-</u>

The CMM as a model for software process improvement got its roots in November 1986 when SEI with assistance from MITRE Corporation, began developing a process maturity framework in order to help organization to improve on their software process. This effort was initiated in response to a request to provide the federal government with a method for assessing the capability of its software contractor. In September 1987, the SEI released a brief description of the software process maturity framework. After four years of experience with the software process maturity framework and the 1987 maturity questionnaire, the SEI evolved the maturity framework into the Capability Maturity Model for software. The CMM is based on knowledge acquired from software process assessments and extensive feedback from both industry and government.[1]

CMMI:-

In the fall of 1997, a review of SEI activities was conducted by the Office of the Under Secretary of Defense for acquisition and Technology (OSD). An OSD led team comprised of government, industry, and the SEI decided to focus on developing an integrated framework for maturity models and associated products [4]. SEI had already initiated an effort to develop a framework to integrate existing maturity models as a result of interest expressed by the model user community. The CMMI project is a collaborative effort among industry, government, and the SEI. It is sponsored by OSD and the National Defense Industrial Association (NDIA) System Engineering Committee.

2. Goals of the Model

CMM:-

The CMM as a model for process improvement defines a maturity framework, which can be used to characterize the capability of a software producing organization. This maturity framework can then be used by an organization to improve its process. The CMM was developed to serve as an assessment tool for Department of Defense (DOD), to identify areas for improvement, and to help process improvement. The CMM describes an evolutionary improvement path for software organization from an ad hoc, immature process to a mature disciplined one. CMM is only applicable to software projects within an organization.

CMMI:-

The CMMI project is a collaborative effort intended to support process and product improvement and to reduce redundancy and eliminate inconsistency when using separate stand-alone models. The goal is to improve efficiency, return on investment, and effectiveness by using models that integrate discipline such as system engineering and software engineering that are inseparable in a system development endeavor [4]. The primary focus of the project is to build tools to support improvement of processes used to develop and sustain systems and products. The output of the CMMI project is a suite of products, which provides an integrated approach across the enterprise for improving processes. Further goals are to ensure that the products of the framework are easy to understand and use because they use common terminology, have consistent style, follow uniform construction rules, and share common components among products. Finally the development team is trying to minimize the impact on those who are using existing models, assessment materials, and training materials [17].

3. Models Used

CMM:-

The structure of the CMM is based on principle of product quality by Shewart, Deming, Juran, and Crosby. The product quality principles have been in existence since the 1930's. The principles of statistical quality control were promulgated by Walter Shewhart in the 1930's. These principles were expanded by W.E. Deming, Joseph Juran, and Crosby. Crosby's quality management maturity grid describes five evolutionary stages done on these principles to develop a maturity framework under the direction of Watts Humphrey at IBM. Humphery was responsible for refining the concept of maturity levels of this maturity framework. The SEI adopted this framework to develop its maturity framework that establishes a project management and engineering foundation for quantitative control of the software process [7].

CMMI:-

The CMMI product suite springs from a framework that generates multiple integrated models, courses, and an assessment method. As new material is added to the framework, more integrated models and supporting materials will become available that cover additional disciplines. The initial mission of the project was to combine the following three source models into a single model for use by organizations pursuing enterprise-wide process improvement.

1. Capability Maturity Model for Software (SW-CMM) v2.0 draft C
2. Electronic Industries Alliance Interim Standard (EIA/IS) 731, System Engineering Capability Model (SECM)
3. Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98

The resulting CMMI models are referred to as Capability Maturity Model Integrated-Discipline, where Discipline is the name of the discipline covered by the model [6] For example, the CMMI for system engineering is designated CMMI-SE; for system engineering and software engineering, the designation is CMMI-SE/SW. These models have the same structure and similar content as previous CMMs and Capability Models, and are tailorable to an organizations mission/business objective.

4. Scope of the Model

CMM:-

CMM as a model for SPI was developed with DoD in mind. The model has been successfully implemented by the DoD to evaluate its subcontractors and this success has percolated into the other software producing organization in the US. The CMM was initially used by DoD Federal contractors and Military or Federal organizations. Air force has used CMM since its inception, Air force looks for contractors with mature software development processes and use SCE during source selection as best practice. There has been a remarkable increase in the number of commercial/in-house organization that uses CMM model [7]. The main reason is that more and more commercial organization want a greater share of the software market and are hence willing to do anything that shows a promise of increased productivity or profitability. CMM as a model for SPI has spread its roots beyond the US and many organizations outside the US have been assessed by this model. Though CMM has been used for assessment and SPI outside the US, it has not really picked up the way it has in the US. The main reason for this is that the assessors have to be from the US. This increase the cost of assessment and organizations in countries other than the US may not be able to afford the cost of the assessment.

CMMI:-

The concept of the CMMI project was to improve the usability of the CMM technology in a wider set of disciplines beyond its initial success for software engineering alone. The concept is to use common terminology, common components, and rules for constructing Capability Maturity Model that would be available with a reduction in the amount of training and process improvement effort needed by user of multiple disciplines. As the concept developed, it was advisable to restrict the initial scope of the CMMI project to a few of the most needed disciplines until the concept was proven. The selection of software engineering, system engineering, and integrated product development CMMs was made by industry and government participants for the initial proof-of-concept phase. The CMMI product suite was designed with the capability to expand in both disciplines and life-cycle coverage. Work has begun on an expansion for acquisition, and coverage of additional disciplines such as security system engineering. The expansion decision will be made based on the success of the initial release, user community needs and support, and availability of participants for development [4].

5. Model Representation

CMM:-

CMM is a staged representation; the structure of the CMM allows the addressing of the capability of an organization's process in terms of maturity levels. CMM provides five maturity levels, which can be used for rating an organization's process. This maturity is a strong indicator of the software process capability of the organization. A maturity level consists of key skill sets, called key process areas. These key process areas are the means by which an organization is successful in achieving its goals. Each contains specific policies, procedures and practices called key practices. The key practices are how the organizational wisdom is institutionalized. In addition, these key practices specify or imply key indicators that may be monitored to measure effectiveness. An organization can be only at one level at a given period of time, as an organization moves up in the maturity level its process is stable and more mature.

CMMI:-

There are multiple CMMI models available, as generated from the CMMI Framework. Consequently, one needs to be prepared to decide which CMMI model best fits its organization's process improvement needs. Basically three are two representation for each model that exist namely staged and continuous. Each representation has its own advantage and disadvantage, so selecting the proper representation depend upon the requirements of the organization and its current practices. In the continuous representation of a CMMI model, the summary components are process area. Within each process area there are specific goals that are implemented by specific practices. Also contained in the continuous representation of a model are generic goals that are implemented by generic practices. Specific goals and practices are unique to individual process areas, whereas generic goals and practices apply to multiple process area. Each practice belongs to only one capability level. In the staged representation, the summary components are maturity levels. Within each maturity level there are process area that contain goals, common features, and practices.

6. Common Feature Comparison

The key practices in each key process area are organized by a set of common features. The common features are the attributes that indicate whether the implementation and institutionalization of a key process area are effective, repeatable, and lasting. There are basically five common features namely:

1. Commitment to Perform:-

Commitment to perform describes the actions the organization must take to ensure that the process is established and will endure.

2. Ability to Perform:-

Ability to perform describes the preconditions that must exist in the project or organization to implement the software process competently.

3. Activities Performed:-

Activities Performed describes the roles and procedures necessary to implement a key process area.

4. Measurement and Analysis:-

Measurement and Analysis describes the need to measure the process and analyze the measurements.

5. Verifying Implementation:-

Verifying Implementation describes the steps to ensure that the activities are performed in compliance with the process that has been established.

SW-CMM v1.1 common features	CMMI common features
Commitment to Perform	**Commitment to Perform**
Establish an organizational Policy	Establish an organizational Policy
Ability to Perform	**Ability to Perform**
	Plan the Process
Provide Resources	Provide Resources
Assign Responsibility	Assign Responsibility
Train People	Train People
Activities Performed	**Activities Performed**
Plan the Process	
Perform the process	Perform the Process
Monitor and control the Process	
	Directing Implementation
	Manage configuration
	Monitor and control the process
Measurement and Analysis	Handled by the measurement and analysis process area
Measure the process	
Analyze the measurements	
Verifying Implementation	**Verifying Implementation**
Review with Org. Management	Review with management
Review with project management	
Objectively verify adherence	Objectively verify adherence

Table 4.1 Common Feature Comparison [25]

7. Key Process Areas

CMMI has some significant differences and improvement from CMM. The major differences between these two models are in the following three areas:

1. Process areas have been added.
2. A capability level goal has been added to each process area.
3. Practices have been added in process areas where necessary.

The table below gives the mapping of the process between these two models [25].

SW-CMM	CMMI
LEVEL 2 REPEATABLE	
Requirement management	Requirement management
Software project planning	Project planning
Software project tracking & oversight	Project monitoring and control
Software subcontract management	Supplier agreement management
Software quality assurance	Process & product quality assurance
Software configuration management	Configuration management
	Measurement and analysis
LEVEL 3 DEFINED	
Organization process focus	Organization process focus
Organization process definition	Organization process definition
Training program	Organizational training
Integrated software management	Integrated project management
Software product engineering	Risk management
Intergroup coordination	Customer & product requirement
Peer reviews	Technical solution
	Product integration
	Verification
	Validation
	Decision analysis and resolution

LEVEL 4 MANAGED	
Quantitative process management	Organization process performance
Software quality management	Quantitative Project management
LEVEL 5 OPTIMIZING	
Defect prevention	Casual analysis and resolution
Technology change management	Org. innovation and deployment
Process change management	

Table 4.2 Mapping of the Process Areas

8. Software Process Appraisal

CMM:-

The CMM uses the process maturity framework as a common frame of reference to find the maturity of an organization. Organization or project can be appraised and found to be at particular maturity level using CMM. The CMM describes two different methods, which can be used for such an appraisal. Software Process Assessment (SPA) called CMM-Based Appraisal for Internal Process Improvement (CBA-IPI) and Software Capability Evaluation (SCE) are the two methods used with CMM. Both the methods use same principles but differs depending on who is performing assessment.

CMMI:-

The assessment method used in CMMI is Standard CMMI Assessment Method for Process Improvement (SCAMPI). The SCAMPI method is based on the CMM Based Appraisal for Internal Process Improvement (CBA-IPI) v1.1 method and the Electronic Industries Alliance/Interim Standard (EIA/IS) 731.2 Appraisal method. SCAMPI satisfies the Assessment Requirement for CMMI (ARC) and is a Class A assessment method. The SCAMPI method is a diagnostic tool that supports, enables, and encourages an organization's commitment to process improvement. The method helps an organization gain insight into its process capability or organizational maturity by identifying strengths and weaknesses of its current processes relative to one or more of the CMMI models.

4.2 Recommendations to be considered during Transition

Organizations that are currently using one or more CMMs should not suffer any improvement setbacks by migrating to CMMI; rather they will gain the benefits of working with the guidance of an updated integrated reference model. Since CMM concepts have not changed, training on CMMI models and assessment will be analogous to current training with courses that have specific focus on structural changes in model representation, as well as new practices and process areas.

This section presents some of the recommendations that could be considered while adopting the new integrated model.

1. Senior Management Involvement

One of the most critical factors for all process improvement is the strength of executive support. Usually there is a major interest in process improvement by senior management. Senior management must be willing to commit funds to support advancement. Executive support is critical to obtain the resources for process improvement activities and to ensure the rewards for the innovation and additional hard work that are fundamental parts of any changes strategy. Leadership must achieve agreement among both middle management and the practitioners; otherwise, process improvement will become a paper exercise [28]. For this reasons, it is essential to get commitment from executives as early as possible. This could be achieved by informing the upper management of the possible benefits that an organization could make after making the transition.

2. Identify relevant business objectives

Identify the business objectives of the organization. What are the future goals and accomplishment? Determine if CMMI meets the organizations needs by reviewing the model and other relevant CMMI information. It is very important that the process improvement is in compliance with the organization needs. Align process improvement and integration efforts with the business goals. In doing so, understand how the cost of process improvement and integration can be reduced as much as possible while continuing to enhance overall competitiveness. In the long run, if process improvement will not affect the bottom line, then it will not be seen as valuable by the management.

3. Evaluate current process

Map the existing process to models and standards of interest or customer requirements. An assessment could be performed to identify the scope and nature of the changes required to adopt a CMMI model. Organization using legacy models could conduct an informal assessment against a CMMI model, potentially with outside support, or may decide to perform a simple analysis of the changes required. The CMMI project is producing mappings from legacy models to/from the CMMI models as an aid in doing this. These mappings can be used for an initial gap analysis. This assessment could be used in following ways:

1. To develop an improvement plan
2. Focus on key infrastructure and culture change issues.
3. Determine the current process effectiveness.

4. Align the Process

Align the current organizational processes so that they are more compatible and usable by both software and systems functions. It is better to develop common processes across organization that have applicability to functional components, and yet can be tailored for specific applications. Continue effort should be done to maintain maturity level credentials in systems and software through continuous management and quality assurance efforts.

5. Process Infrastructure Improvement

As the organization aligns the system and software process towards its business goals it should identify areas of process infrastructure that overlap and can support integration efforts. Through the use of a Software Engineering Process

Group (SEPG), which can be formed from a system or software engineering group, establish the organizational focus on integrated process efforts and create a management steering committee through additional involvement of other functional discipline representatives. Enhance the organizations Process Asset Library to include additional artifacts resulting from and supporting process integration efforts, and expand the organization process database to include new measurement to help define integrated process capability descriptions [26].

6. Common Process Integration

Identify common process that are used by both systems and software engineering, such as quality assurance, configuration management, inspections or peer reviews, risk management, and project management. Improve them for better integration and application to common organizational objectives, so that the various disciplines can be effectively applied to business solutions where appropriate in the near future [26].

7. Picking the Target Carefully

Many process improvement programs fail because they make poor estimates of what they can reasonably accomplish in a given amount of time. Process improvement can be difficult work, especially across organizations with strong personalities and well-defined cultures. For this reason, it is necessary to consider the initial process improvement goals very carefully [28]. One good point would be to select the achievable objectives that can immediately show a benefit to the organization. It is a good idea to consider an initial goal of improving in at least one cross-discipline or cross-organizational process this will help not only to bring more of the stakeholders into the early planning, but often find significant reengineering gains that can pay for some of the later process improvement work. Care must be taken to use the measurement program to establish a baseline against which to compare the organization process improvement activities. This is important because the case should not be that the organization ends up with just the raw data in hand to defend its success against group who oppose the new process improvement. It is very essential to have a convincing data to show that process improvement is affecting the organizations bottom line or at least some of the bottom line.

8. Forming Responsible Group

It is critical to establish roles and responsibilities for those individuals who will lead, carry out training, and perform the process improvement activities. This is even more critical when the organization is approaching process improvement in an integrated fashion. Generally a group of process champions will provide the core experience and the initial start up to get process improvement off the ground. The steering group should not be limited to practitioners and process improvcmcnt spccialists. It will prove much more effective if middle management is included and if executive management is represented or chairs the group. In this way, the process group can adequately deal with resource and other management issues. A process group that comprises mostly managers probably will not get the most of the work done. To achieve the goals, it will need subgroups, committees, or work teams to coordinate the training, build better processes, maintain process assets, and perform all the other crafts involved in process improvement [28].

9. Training

Training is a key element in the ability of organizations to adopt CMMI and is therefore a key part of the Product Suite. Training is probably the most expensive and most important aspect of the infrastructure. Software and system engineers may need to be cross-trained. They certainly need to understand the fundamentals of one another's discipline to define effective processes that involve both. Although most organizations have internal resources that can provide much of this training, others may need to look into local university courses or professional training resources [28]. While the SEI and its transition partners will provide an initial set of courses, organization may wish to supplement these courses with internal instruction. This approach allows the focus of organizational attention to be placed on the areas marked for greater attention due to the linkage to the product development value chain. Initial training will be available for both representations of CMMI models, with additional training provided to assist those who will need to guide improvement on the EPG, or those seeking to become lead assessors.

4.3 Conditions favorable for moving from CMM to CMMI

1. Organizations that have already achieved a high level of maturity may wish to make the transition more quickly to take advantage of the additional organizational coverage described in the CMMI model. These organizations will find strong commonality between this and the heritage model. There is also significant improvement in coverage of the engineering dimension, more detailed coverage of risk management and measurement and analysis that was less specific in the Software CMM.

2. If the organization is serving for the government organization like DOD then it should make the transition or plan for transition as early as possible. The reason is because the DOD is strictly following the model generated from SEI and it had earlier issued a notice to its contractor that they should be at least at CMM level 3 in order to continue with them. If the organization anticipates future customer requirement for CMMI compliance e.g. DOD then it is a good point to make transition. Most of the organizations are still reluctant to make the transition, as no sufficient data are available to defend them for a transition. In this situation any organization that takes the benefit of the new integrated model might flourish in the future.

3. If the current customer have indicated an interest in the new integrated model and the competitors are moving to the CMMI model then it is important for the organization to make a quicker move for the transition or think for an alternative. If the decision is not made at the earlier stage chances might be of losing the customer, which might jeopardize the future transition. This is because once the customer are gone it might put the small organization in financial Problem.

4. Pure software organizations will certainly have less incentive to migrate as compared to multi-disciplinary organization; a pure software organization should consider migrating to CMMI if

    a. They have just started on their process improvement journey

    b. Their review of the CMMI model has revealed that some of the newer components will alleviate some of the pain points not addressed by the CMM.

5. If the scope of the organization includes more than one discipline like software, system engineering then that type of organization should make a transition as the new integrated model is more robust and support multi-discipline. This can reduce cost, improve customer satisfaction, increased productivity and many other benefits that integrated model provide.

6. If a particular organization is just starting the process improvement, then it would be much better to start with CMMI model. Why bother investing in the CMM and then be forced to migrate to CMMI?

4.4 Conditions Not favorable for moving from CMM to CMMI

1. If there are no driving reasons for the transition and the goal is to achieve the benefits of process improvement then it would be best to wait for further model improvement. The reason is there are extra costs involved.

2. If an organization has already made a large investment in their CMM based process improvement initiative then it would be a good idea to wait for the result of the new CMMI model. Implementing new model involves extra cost, which will add extra cost in the present process improvement.

3. If an organization is close to achieving the next CMM level for example, suppose it is 6-12 months away from achieving maturity level 3 using the software CMM; then it might be wiser to continue the process improvement with CMM. Once the target is achieved then the organization might consider of making a transition. Organizations that have begun significant movement toward a maturity level 2, 3, or 4 assessment must weigh the costs of making the transition against the benefits of the improved coverage the integrated model offers.

4. Organizations that are more conservative may want to delay their transition until the CMMI model has been fully established.

5. An organization has to be very cautious; it might be probably at CMMI level 1 at first. This may be difficult for an organization to accept and this should not destroy the spirit of the organization just to follow a new, unproven model. Care must be taken to consider this fact and if it really happens to hamper the spirit of the organization then organization should defer the plan of transition.

4.5 Profile of the organization for CMMI Model

CMM for process improvement was initially adopted by the organizations that supported DoD and were the driving reason for the success of the CMM. The successes of the organizations capability were shared and this success spread among other software developing organization. CMMI is adopted by very few organization and major adopters are the government and DoD contractors. The current situation of CMMI is same as the initial stage of CMM. The following suggestions are based on the survey of the organizations that adopted CMM at earlier stage, as CMMI is mostly inherited from CMM and faces same problems.

Following are the profile or the criteria that an organization should have in order to adopt the CMMI for continuous improvement.

1. Financial Resources

Financial resources are one of the most important factors that have to be considered before starting the process improvement. Most organizations that get involved or wish to get involved with process improvement are in the business to make a profit. Process improvement is laborious, long and expensive process; it takes money, computer resources, human resources, tools and techniques, training and consulting support. It is important to determine how much process improvement an organization can support and then plan for adopting the model for improvement. CMMI model is more complex and has more process areas; so the organizations should have more financial resources if planning to adopt the model.

2. Process Improvement Experience

An organization should have some previous experience in the process improvement field by using any process improvement model or principles. CMMI model is new and more complicated as compared to CMM so directly trying to adopt the model for process improvement without previous process improvement experience might be counterproductive and misleading. A survey of the organizations revealed that almost all the organizations had some sort of process improvement within the organization before adopting the CMM. Implementation success, as always, will depend on the sophistication of capability model that an organization has already reached, and the experience level of its configuration manager. So organizations should exhibit some basic process improvement initiatives within the organization.

3. Responsible Software Engineering Process Group

An organization should have its own experienced Software Engineering Process Group (SEPG). Having SEPG membership of both full-time personnel and part-time personnel drawn from various projects is very important. The full-time members provide continuity for the process improvement efforts, while the part-time members act as advisors, advocates, and communication liaison. SEPG team must be trained, mentored, and coached so that they are familiar with organizational changes.

4. Technology Support

An organization should be competent with the change in technology. Appropriate technology should be used to support the process improvement initiative and that both process and technology are needed to allow workers to be as creative and productive as they can be. Technology is necessary to support the managers and developers working on today complex systems. Overview of CMM or CMMI might give one the impression that technology is not thought about until level 5, but that is not true always. Technology is required to support the managers and developers at every level; however, the technology must complement the process, not drive it. Software process improvement must take past, present, and future technology into consideration.

5. Organization Scope

CMMI has some great improvements over the CMM and clarified a lot of the issues, which were poorly described or absent from CMM, CMMI also added a lot of system engineering criteria. So this model is more suited for large, system engineering organizations or the organizations engaged in integrated system development. A survey of the organizations that adopted the CMMI model revealed that almost all the organizations were DoD contractors and most of them were engaged in multiple disciplines with different models for process improvement. This model is very good for organizations performing multiple functions in different areas like Information technology, system engineering, space systems, Telecommunication software engineering and others.

4.6 Success of CMM

In software development it is logical to assume that a discipline process is likely to lead to successful projects and higher product quality. Solid quantified results are needed to convince people that initiating process improvement will result in terms of better quality, better productivity, shorter schedules, and higher user satisfaction. CMM has been used for almost 10 years and there are concrete data available to show that it has turned out to be the most effective process improvement model.

Following are the data of the benefits of process improvement using CMM in 13 different organizations that represent a variety of maturity levels [40].

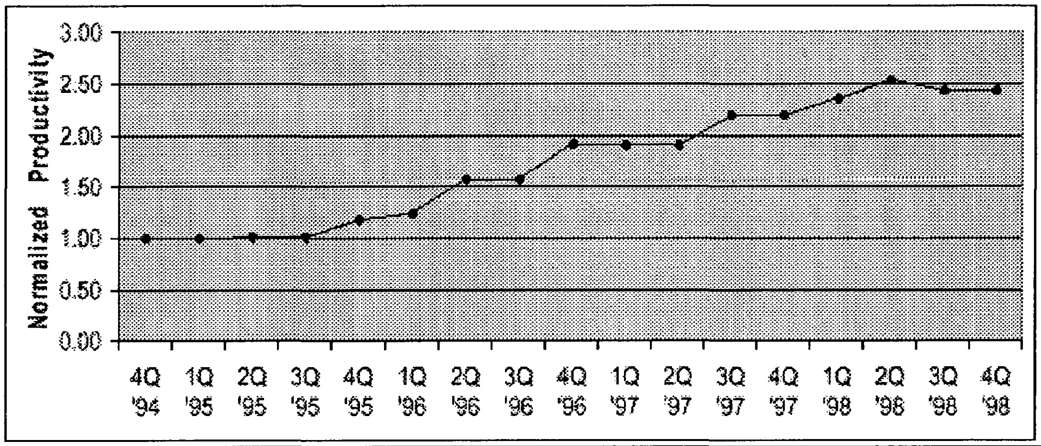
Category	Range	Median
Total yearly cost of SPI activities	$49,000 - $1,202,000	$245,000
Years engaged in SPI	1 - 9	3.5
Cost of SPI per software engineer	$490 - $2004	$1375
Productivity gain per Year	9% - 67%	35%
Early detection gain per year (defects discovered pre-test)	6% - 25%	22%
Yearly reduction in time to market	15% - 23%	19%
Yearly reduction in post-release defect reports	10% - 94%	39%
Business value of investment in SPI (value returned on each dollar invested)	4.0 - 8.8	5.0

Table 4.3 Summary of the overall Result for CMM

Looking at how performance changed over time within each organization as improvement efforts were implemented, it was identified that there was substantial gains in productivity, early defect detection, time to market, and quality.

Following are the data of process improvement at Raytheon

The data in Figures 1 and 2 show that Raytheon organization's productivity has improved 144 percent from 1995 when they were a Level 2 to 1998 when they were a Level 4. During the period that productivity was increasing by 144 percent, they expended 6 percent of the annual budget on process improvement. This yields a 6-to-1 return on investment. Raytheon overall defect containment during the same period increased from 32 percent in phase to 72 percent in phase [37].

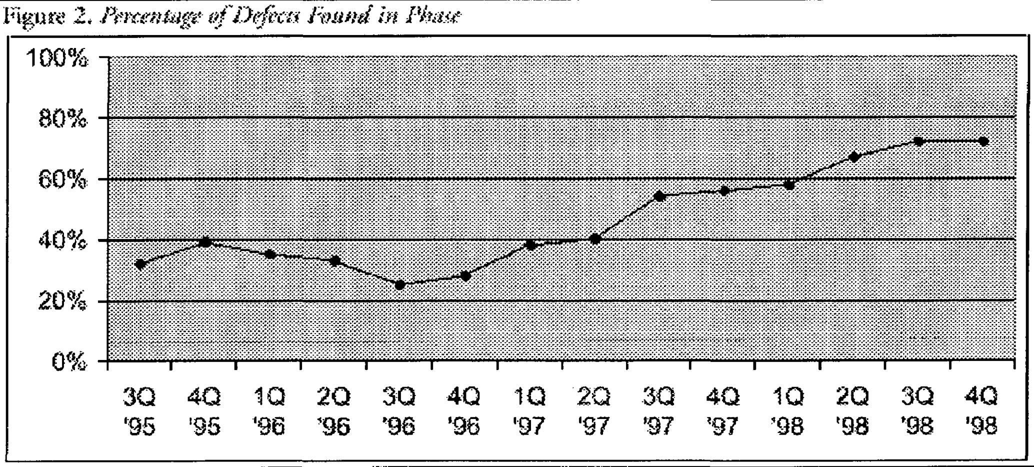


Date	4Q '94	1Q '95	2Q '95	3Q '95	4Q '95	1Q '96	2Q '96	3Q '96	4Q '96	1Q '97	2Q '97	3Q '97	4Q '97	1Q '98	2Q '98	3Q '98	4Q '98
Productivity	1.00	1.00	1.01	1.01	1.18	1.24	1.57	1.57	1.91	1.90	1.90	2.20	2.20	2.35	2.54	2.44	2.44

Figure 1. *Organizational Productivity*

Figure 4.1 Organizational Productivity

Figure 2. *Percentage of Defects Found in Phase*



Date	3Q '95	4Q '95	1Q '96	2Q '96	3Q '96	4Q '96	1Q '97	2Q '97	3Q '97	4Q '97	1Q '98	2Q '98	3Q '98	4Q '98
Defects in Phase	32%	39%	35%	33%	25%	28%	38%	40%	54%	56%	58%	67%	72%	72%

Figure 4.2 Percentage of Defects found in phase

4.7 Success of CMMI at Sverdrup

Sverdrup Advanced Systems Group needed to develop and institutionalize an integrated engineering process for systems and software engineering. After mini-assessment with CMMI SE/SW it was realized that the integrated model would affect nearly all functions of the organization and Sverdrup choose to have CMMI as process improvement model. Minimizing the cost of process and product engineering was critical to maintaining Sverdrup's corporate profit goals and directly supported the generation of new business. To reduce process engineering costs, the company decided to implement both system and software engineering in a single integrated process, saving an estimated 40% over the cost involved in adopting each type of engineering separately [28].

Other cost reduction activities included the following:

- Minimizing the size of the full-time staff.
- Relying on the contributions of some of the best engineers in the organization as process designers and domain experts.
- Implementing the distributed work environment.
- Using pilot projects to finalize the details of the standards process and testing its veracity.
- Adopting a knowledge management approach to training.

There are no data available about the success of CMMI model as it is new and the organizations are still in the beginning phase of transition. However some of the lessons learned and guidance from the early adopter are available and these are presented below.

4.8 CMMI Transition guidance and Lessons learned at Harris Corporation

Following are the guidance and lessons learned at Harris [38]:

- Start Early
  - Understand the difference from legacy models and methods.
  - Identify gaps and establish action plans.
  - Do not underestimate the organization learning curve.
- Continue current improvement based on legacy models
  - Do what makes sense for your organization.
  - Investment will migrate naturally to CMMI.
- Consider opportunities to implement integrated engineering assets
  - Policies, processes, training, metrics
  - Reinforce with templates, checklists, etc.
- Use a variety of assessment methods (class A, B, C methods)
  - Mini-assessments, quick looks, etc.
  - Class A (SCAMPI) assessment not always the appropriate choice.

4.9 CMMI Transition guidance and Lessons learned at Raytheon

Following are the guidance and lessons learned at Raytheon [39]:

- Scope and detail of model
  - Larger, not as specific and more generic than SW-CMM
  - Addresses broad scope of organization and project roles.

- SCAMPI assessment
  - Requires more knowledgeable leadership and broad team understanding.
  - Includes roles not previously included in typical software assessments.
  - May take more time and effort; care in selecting appropriate method (may use class B or C for intermediate assessments)
- Organizational impact
  - Expands prior involvement in deploying processes
  - More integration of functions, greater executive and business-focused sponsor involvement.
  - Implies more integration of functions and processes; more collaborative emphasis.
- Coordinating and planning
  - More effective with broad, multidisciplinary programs experience, management knowledge.

4.10 Advantages of CMMI Model

1. There is always a cost to integration, but the savings from process improvement integration can be significant. By applying single model, organizations that would otherwise use multiple models can reduce the cost of following [28]:
    a. Training in multiple models and assessment methods.
    b. Performing multiple assessments within the same organizations.
    c. Maintaining redundant process assets in a repository.
    d. Maintaining or purchasing expertise in multiple models.

2. The success that arise from integrated process improvement also make it more likely that the organization will achieve cost savings resulting from higher quality, better predictability, and increased customer satisfaction.

3. An integrated process improvement program can clarify the goals and business objectives of the various initiatives. By integrating process improvement activities across a wider range of disciplines, it becomes easier to bring together both practitioners and executive to the process improvement. Having a single process improvement focus can unify and reinforce vision, efficiency and apply scare resources, and provide a common language for improvement across various disciplines [28]. In particular, a single model with common terminology and common assessment methods provides this kind of focus.

4. One of the less obvious benefits of integrated process improvement is the integration effect it has on organizations. When processes are defined across organizational and discipline boundaries, new understanding and mutual education often occur, resulting in the streamlining of the critical work and the elimination of redundant or unneeded activities [28].

5. One of the best benefits provided by integration is the ability to add disciplines as the business or engineering environment changes. Adding a new individual model results in a great deal of redundancy and often conflicting representations in the common process improvement practices. Adding a discipline within an integrated program simply means a few more process areas and perhaps the reinterpretation of other areas, but the fundamental process improvement structure and terminology remain the same.

4.11 Disadvantages of CMMI Model

1. The costs associated with implementing the CMMI model are higher as compared to that of the CMM model.

2. CMMI model added many practices that may not be in the best interest of all organization. Some of the practices are impractical for small companies or projects.

3. The model is very new and no hard data are available for comparison. There should be some kind of data/information from the industry who have used this model and their experience. If such information is available it makes easier for the industry to adopt the model, as was the case with CMM. Since it is not with CMMI model, it makes difficult for the organizations to decide about the immediate transition.

4. All the lead assessors and the team members of the CMMI model are based in the United States. So small organization outside United Stated may be reluctant to make transition, as they have to incur additional expenses for assessment. This is not a major problem for big organization, but might be for small organizations.

Chapter 5

Contributions and Further Research

Generally, model based process improvement begins with management commitment and an assessment. The findings from this assessment, in turn, feed action plans. When these plans have been completed, further assessments are performed and the cycle continues. The goals is for the organization to mature so that it continuously monitors and improves its processes, consistently produces high quality products, is agile within its marketplace, and adjusts quickly to customer needs. There is, and always will be, tension between the desire to improve, change, and grow, and the desire for stability and an avoidance of moving targets. This choice should not apply to CMMI, however both change and stability are fundamental to process improvement. The CMMI models must have sufficient stability so that an investment in process improvement is not undercut by radical and pervasive changes to the models. At the same time, the CMMI model must change to incorporate new knowledge, proven experience, and the continuous creativity of those who practice and preach process improvement. Organizations that are currently using one or more CMMs should not suffer any improvement setback by migrating to CMMI. Rather they will gain the benefits of working with the guidance of an updated integrated reference model.

Many of the research have been done in different model like CMM, SPICE, ISO 9000, BOOTSTRAP etc. Comparison has been done among this model and rich value of information is available for the user of this model. However less research and information is available in the new integrated model as it is very new and has been into practice recently. In this thesis I have done research in the integrated model with the intent of providing brief information and the scope of the model. It is very difficult to adept the change in environment in any field and nobody wants to do unless forced to or have to, for some or the other reasons. The same case is with the user of the CMM model. SEI has set the sunset period for CMM in the year 2003, after this period there will be no further development, support, assessment or training in the model from the SEI as per the current news. In this situation some of the organization might be forced to move due to the nature of the business. As no hard data are available about the integrated model the organizations are reluctant to move. This thesis compares the two model SW-CMM and CMMI to give the difference between the model and its scope. Next, the steps necessary to be considered before the process improvement begins are suggested. There are some situations under which organizations are forced to make the transition; I have recommended some of the conditions under which it is favorable for making the transition and conditions under which it is not favorable for making the transition. This research will provide valuable information and guideline to the user and the organization that wish to gain the knowledge of the integrated model and its applicability.

5.1 Contributions

The main contribution of the thesis are presented below in brief:

- A comparison between the two model namely CMM and CMMI has been done taking into consideration various important aspect of the model. This gives a clear picture about the model and its scope. This could be helpful in understanding the difference between the model and the applicability of the model as appropriate by the organizations. Some of the points discussed in the comparison are Evolution of the model, which gives detail about how and from where the model came into practice. Goals of the model describe why the model was created and its intended purpose. Models used describe the previous model or theory used to make the CMM and CMMI model. Scope of the model describes the applicability of the model to different organizations and its widespread use throughout the software industry. Model representation describes whether it is a staged or a continuous representation; as integrated model is derived from CMM for software and system models it has both representation for each model derived from the framework. Software process appraisal focuses on the assessment methods used to identify the weakness or gap in the current process compared to the integrated model. These findings will help the organizations to utilize the resources in right place, make improvement in their software process and move towards the maturity levels.

- A set of possible recommendations that might have to be considered while making a transition is provided. One of the most critical factors for all process improvement is the strength of executive support. Senior management must be willing to commit funds to support advancement. Executive support is critical to obtain the resources for process improvement activities and to ensure the rewards for the innovation and additional hard work that are fundamental parts of any changes strategy. Another important consideration would be to identify the business objectives of the organization. Determine if the integrated model meets the organizations needs by reviewing the model and other relevant CMMI information. Next would be to evaluate current process; an assessment could be performed to identify the scope and nature of the changes required to adopt a CMMI model. Some of the other recommendations mentioned in the thesis are Aligning the current organizational process so that they are more compatible and usable by both software and systems functions. Identifying areas of process infrastructure that overlap and can support integration efforts. Identifying common process that are used by both system and software engineering, such as quality assurance, configuration management, inspections or peer reviews, risk management and project management; and improve them for better integration. Establishing roles and responsibilities for those individual who will lead, carry out training and perform process improvement activities. This is even more critical when the organization is approaching process improvement in an integrated fashion. These are very important factors that have to be considered while making the transition and the organization will definitely gain an advantage if these recommendations are considered.

- This thesis recommends some of the circumstances under which it is favorable for making transition from CMM to CMMI and the circumstances under which it is not favorable. Recommendations are provided considering the organization at any particular situation like for example organization supporting DOD, organization using multiple disciplines for process improvement etc. If the organization is serving for the government organization then it should make the transition or plan for transition as early as possible due to various reason cited in detail in chapter 4. If the scope of the organization includes more than one discipline like software, system engineering then that type of organization should make the transition. If there are no driving reasons for the transition then that type of organization should not make the transition immediately, instead they should wait and see the improvement made by the integrated model. There are some more criteria and conditions that have to be considered while making the transition and this are discussed in detail in chapter 4. These recommendations will help the organization to make a quicker decision about making the transition.

- A Profile or criteria that an organization should have for continuous process improvement are recommended. These suggestions are based on the survey of the organizations who adopted CMM in the early stages and their success. CMMI is also evolving through the same phase as that of CMM and this recommendation will be of potential use for the organizations making the transition.

5.2 Directions for Further Research

During my research in this topic I found some of the interesting areas for further research. These areas are presented in the following section.

Making the Transition cost effective

- ✓ This thesis gives a brief idea about the two models CMM and CMMI and recommendations to be considered in making the transition. An interesting research topic would be "How to make the transition cost effective"? Definitely there is more cost involved while making the transition as new model have extra process areas added and the organizations does not want extra cost involved. This will provide great help and resources to the organization that wish to make the transition.

Combining the two representations

- ✓ Many people, in discussing CMMI model and its two architectural representations (continuous and staged), have expressed the viewpoint that in the future the CMMI models should move toward a single representation. After all, a close study of the two representations shows that while they have differences, they are much more alike than different. With two representations, an organization must endure confusion and perhaps wasted effort before deciding which one to use. Having two representations also affects the choice of assessment method. So why not merge the two representations? This is one of the most interesting and challenging research topic that I came across during my research.

Making CMMI applicable to small organizations

- ✓ CMM has been successfully used in the US as well as around the world. One of the main drawbacks, which have been identified with the model, is that it is not appropriate for small organizations. CMMI is derived from the CMM model and has more process areas added to it; so this model is also not applicable for small organizations. An interesting area of further research would be to make the integrated model applicable to small organizations, so that they could reap the benefit of the integrated model and perform continue process improvement.

Bibliography

[1] Paulk M.C., Weber C.V., Curtis B., Chrissis M.B. [1995]. "The capability Maturity Model: Guidelines for improving the software process". Reading, MA: Addison-Wesley.

[2] Caputo K. [1998]. "CMM Implementation guide". Reading, MA: Addison-Wesley

[3] Jalote P. [2000]. "CMM in practice". Reading, MA: Addison-Wesley.

[4] "CMMI frequently asked questions (FAQ)". http://www.sei.cmu.edu/cmmi.

[5] Schach S.R. [1999]. "Classical and Object-oriented software engineering". Boston, MA: McGraw-Hill.

[6] "CMMI General Information". http://www.sei.cmu.edu/cmmi/general.

[7] Chennupati K. J. [1996]. "A Framework for evaluating the software process improvement models". Thesis, University of Nebraska at Omaha.

[8] Briand L.; Emam K. "Cost and Benefits of Software Process Improvement". Technical report ISERN-97-12, 1997. Institute for Experimental Software engineering.

[9] Arent J. "Transforming Software Organizations with the Capability Maturity Model". A paper delivered at the International Conference on Product Focused Software Process Improvement (PROFES) June 2000.

[10] Arent J., Iversen J. H., Andersen C.V., Bang S. "Project Assessment: Supporting Commitment, Participation, and Learning in Software Process Improvement". Department of Computer science, Aalborg University.

[11] Arent J. "Making Software Process Improvement Happen: A learning perspective". Department of Computer science, Aalborg University.

[12] Humphrey W. S. [1990]. "Managing the Software Process". Reading, MA: Addison-Wesley.

[13] Curtis B., Hefley W.E., Miller S. "People Capability Maturity Model, Carnegie Mellon University 1995.

[14] Paulk M. C., Curtis B., Chrissis M.B., Weber C.W. "Capability Maturity Model for Software". Technical Report CMU/SEI-93-TR-024, ESC-TR-93-177, February 1993.

[15] Humphery W.S. [1995]. "A Discipline for Software Engineering". Reading, MA: Addison-Wesley.

[16] Broadman J.G., Johnson D.L. [1994]. "What small business and small organization say about the CMM? In Proceeding of the 16th International Conference on Software Engineering. Sorrento, Italy.

[17] Bate R., Shrum S. "CMM Integration Framework". Spotlight vol 1, Issue 2, September 1998.

[18] Shrum S. " Choosing a CMMI model Representation". Crosstalk vol. 13, No 7, July 2000.

[19] "CMMI for Systems Engineering/Software Engineering version 1.02" staged representation, Carnegie Mellon Software Engineering Institute, Pittsburgh, PA. http://www.sei.cmu.edu/cmmi/

[20] "CMMI for Systems Engineering/Software Engineering version 1.02" continuous representation, Carnegie Mellon Software Engineering Institute, Pittsburgh, PA. http://www.sei.cmu.edu/cmmi/

[21] "Standard CMMI Assessment Method for Process Improvement: Method Description version 1.0". Technical Report CMU/SEI-2000-TR-009 ESC-TR-2000-009; October 2000.

[22] "Assessment Requirement for CMMI, version 1.0". Technical Report CMU/SEI-2000-TR-011 ESC-TR-2000-011; August 2000.

[23] Jarvis A.; Hayes L. [1999]. " Dare To be Excellent case studies of Software Engineering Practices that worked". Upper Saddle River, NJ: Prentice Hall.

[24] Moller K. H.; Paulish D.J. [1993]. "Software Metrics A Practitioner's guide to Improved Product Development". Boundary Row, London: Chapman and Hall.

[25] Rassa B.; [1999]. "Capability Maturity Model Integration". Presentation to the Omaha SPIN.

[26] "CMMI: Making the transition" The Software Productivity Consortium Quarterly. Vol. 15 No.1 March 2001.

[27] Humphrey W.S. "Characterizing the software Process: A maturity framework". IEEE Software, Vol 5 March 1988.

[28] Ahern D.M.; Clouse A.; Turner R. [2001]. "CMMI Distilled: A practical Introduction to Integrated Process Improvement". Upper Saddle River, NJ: Addison-Wesley.

[29] Schulmeyer G. G.; McManus J. I. [1999]. "Handbook of Software Quality Assurance". Upper Saddle River, NJ: Prentice Hall.

[30] Yeh H. [1993]. "Software Process Quality". New York, NY: McGraw-Hill Inc.

[31] Grady R. B. [1997]. "Successful Software Process Improvement". Upper Saddle River, NJ: Prentice Hall.

[32] McGuire E. [1999]. " Software Process Improvement: Concepts and Practices". Hershey, PA: Idea Group Publishing.

[33] Donaldson S. E.; Siegel S. G. [1997]. "Cultivating Successful Software Development A practitioner's view". Upper Saddle River, NJ: Prentice Hall.

[34] Emam K. H.; Drouin J.; Melo W. [1998]. "SPICE The Theory and Practice of Software Process Improvement and Capability Determination". Los Alamitos, CA: IEEE Computer Society Press.

[35] Laryd A.; Orci T. "Dynamic CMM for small organizations". Department of Computer Science, Umea University; Umea Sweden.

[36] Zhang C. [2000]. "Devising A Software Process Improvement Models for small-scale software development". University of Nebraska at Omaha.

[37] "Raytheon Stands firm on Benefits of Process Improvement" http://www.stsc.hill.af.mil/crosstalk/2001/mar/bowers.asp

[38] "CMMI Transition at Harris Corporation" SEPG 2001 Conference March 13th 2001 http://seir.sei.cmu.edu/

[39] "CMMI Transition Approach at Raytheon" SEPG 2001 Conference March 13th 2001 http://seir.sei.cmu.edu/

[40] Herbsel 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-013-ESC-TR-94-013.

Appendix

The organizations listed below were some of the first to implement CMMI-based process improvement initiatives.

- ABB
- Accenture
- ARINC
- BAE Systems
- The Boeing Company
- Computer Sciences Corporation (CSC) Defense group
- Concurrent Technology Corporation (CTC) National Security Division
- General Dynamics
- Goddard Space Flight Center NASA
- Harris Corporation
- Jacobs/Sverdrup Technology Inc., Advanced Systems Group Engineering Performance Improvement Center (EPIC)
- Lockheed Martin
- Lockheed Martin management and systems
- The MITRE Corporation
- Motorola Inc.
- Northrop Grumman Information Technology Sector

- Northrop Grumman Integrated Systems Sector Airborne Early Warning/Electronic Warfare Systems
- Process Assessment, Consulting & Training
- Q-Labs
- Raytheon Company
- Synchro PP&T
- THALES
- TRW
- United Space Alliance
- U.S. Army TACOM-ARDEC Software Enterprise