BHILWARA INTRANET PROJECT FOR CMM LEVEL 3 AUTOMATION

A DISSERTATION

Submitted in partial fulfilment of the requirements for the award of the degree

of

MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By

AMIT GUPTA

ER & DCI
NOIDA

IIT Roorkee-ER&DCI, Noida
C-56/1, "Anusandhan Bhawan"
Sector 62, Noida-201 307

FEBRUARY, 2003

ENROLLMENT NO:- 019004

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this dissertation titled **"BHILWARA INTRANET PROJECT FOR CMM LEVEL 3 AUTOMATION"**, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Information Technology,** submitted in **IIT, Roorkee – ER&DCI Campus, Noida,** is an authentic record of my own work carried out during the period from August, 2002 to February, 2003, under the guidance of **Mr. Sandeep Santa Kumar** Quality Analyst, Bhilwara Infotech Limited, Noida and **Mr. Munish Kumar**, Project Engineer, Electronics Research and Development Center of India, Noida.

The matter embodied in this dissertation has not been submitted by me for award of any other degree or diploma.

Date : 21-02-03
Place : Noida

Amit Gupta
(Amit Gupta)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date : 21-02-03
Place : Noida

Co-Guide

21-02-03

(Munish Kumar)
Project Engineer,
ER&DCI,
Noida - 201 307.

Guide

(Sandeep Santa Kumar)
Quality Analyst,
Bhilwara Infotech Limited,
Noida - 201301.

ACKNOWLEDGEMENT

It gives me great pleasure to take this opportunity to express my deep sense of gratitude to Prof. Prem Vratt, Director, IIT Roorkee, Sh. R.K.Verma, Executive Director, ER&DCI, Noida, Prof. A.K.Awasthi, Dean PGS&R, IIT Roorkee, Prof. R.P.Agrawal, Course Coordinator, M.Tech.(IT), IIT Roorkee and Sh.V.N.Shukla, Course Coordinator, M.Tech.(IT), ER&DCI, Noida, for their invaluable support and inspiration to me throughout my dissertation work.

I would like to thank my guides Sh. Munish Kumar (ER&DCI, Noida) and Sh. Sandeep Santa Kumar (BIL, Noida), for their valuable suggestions, guidance, support and constant encouragement during the course of my dissertation work. I deemed it my privilege to have carried out this dissertation under their able guidance.

I would like to thank all my friend and well wisher who helped me directly or indirectly in making this report.

Finally, I express my regards to my parents, who have been a constant source of inspiration to me.

Amit Gupta

(Amit Gupta)

Enroll. No. 019004.

CONTENTS

ABSTRACT

Today the Software Industry is driven by the quality of the software product. The quality of a (software) system is largely governed by the quality of the process used to develop and maintain it. So this premise focuses on process as well as product. And this has created a need for process improvement , which requires methods for process analysis and assessment.
One of the most extensive and influential software process improvement and assessment framework is the Capability Maturity Model (CMM) developed by Software Engineering Institute (SEI) at Carnegie Mellon University.

Capability Maturity Model is application of process management and quality improvement concepts to software development and maintenance, it is a guide for evolving toward a culture of engineering excellence. A model for organizational improvement as shown in [1]. In Other words Capability Maturity Model for Software (CMM or SW-CMM) is a model for judging the maturity of the software processes of an organization and for identifying the key practices that are required to increase the maturity of these processes as shown in [2]. It is intended to help software organizations improve the maturity of their software processes in terms of an evolutionary path from ad hoc, chaotic processes to mature, disciplined software processes.

This thesis aims to study the CMM and its various KPA's in detail, and on the basis of it, develop an application which would map the functionality of the various KPA's of CMM up to level 3 or if possible the above level. With its help the organization will be able to automate the whole process of the project. The Application based on the standard process

described in the QMS(Quality Management System) will automate whole part of the SDLC documentation process which will be useful in tracking of the project which can be an important requirement for the management.

The Application will also be useful for the organization to collect metrics to apply for CMM Level 3 or above levels and track status of the various projects. This application will serve an important part in the CMM assessment method.

The application will be used by all the employees of the organization and mainly by the development team, the quality department and the management for collecting the metrics of the process.

The application could be further enhanced to automate more process modules so as to help in attaining still higher levels of CMM. Could be helpful to achieve the organization wide data like Attendance, Time sheet, etc.

INTRODUCTION

1.1 Background

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 that don't work as expected also cause problems for the organization's customers. 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.

To build this process infrastructure, organizations producing software need ways to appraise their ability to perform their software process successfully. They also need guidance to improve their process capability. Customers, such as the Department of Defense (DoD), need ways to evaluate more effectively an organization's capability to perform successfully on software engineering contracts. Prime contractors need ways to evaluate the capability of potential subcontractors.

To help organizations and customers like the DoD and prime contractors, the (SEI) has developed the Capability Maturity Model for Software (CMM), that delineates the characteristics of a mature, capable software process. The progression from an immature, unrepeatable software process to a mature, well-managed software process also is described in terms of maturity levels in the model. The Software Engineering Institute is Federally Funded Research and Development Center (FFRDC), affiliated with Carnegie Mellon University and was established in 1984.

What is a Process ?

The means by which people, procedures, methods, equipment, and tools are integrated to produce a desired end result depicted in Figure 1.1.

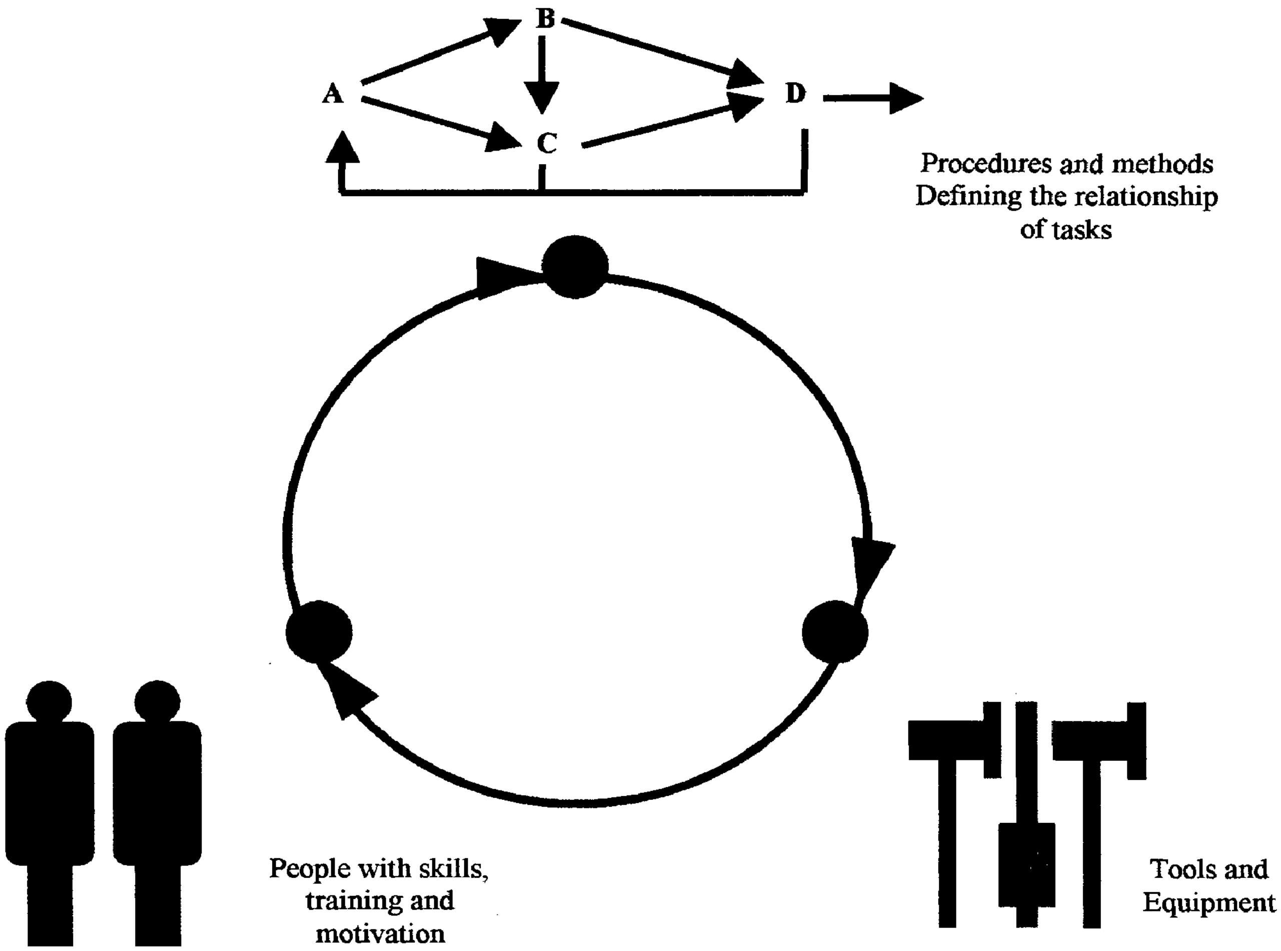


Figure 1.1 Process Model.

What makes Quality System?

The quality of a (software) system is largely governed by the quality of the process used to develop and maintain it. This premise implies focus on process as well as product. The value of this premise is visible world-wide in the Total Quality Management movements in the manufacturing and service industries.

Total Quality Management (TQM) is the application of quantitative methods and human resources to improve :

- The material and services supplied to an organization.
- All the processes within an organization.
- The degree to which the needs of the customer are met, now and in the future.

TQM to Software Organization's Process improvement fits in an overall business context. **Where as CMM applies to software**. Figure 1.2 below depicts the place of the CMM in the Organization.

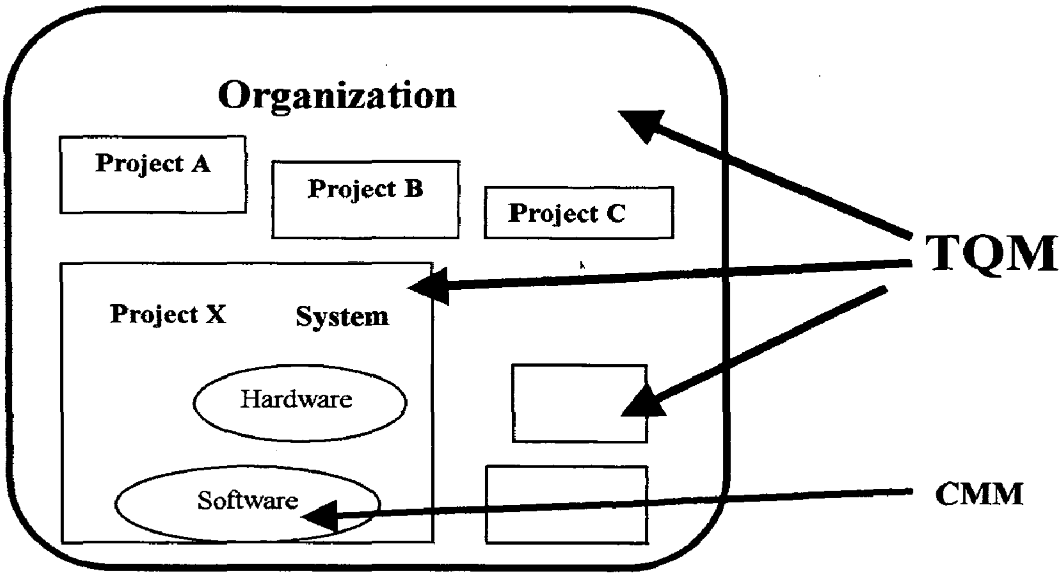


Figure 1.2 Applying TQM to Software Organization.

1.2 Objective Of Dissertation

The objective of the Dissertation is to develop a live application which will be useful for all the Software Organization wanting to improve their product quality and in attaining the CMM Level 3 Certification. The application **Bhilwara Intranet Project For CMM Level-3 Automation** will automate the documentation process followed in the project, will implement the various process involved in SDLC(Software Development Life Cycle) and will map the processes to CMM KPA's in an software Development Organization. Based on the standard process described in the QMS(Quality Management System) the application will automate whole part of the SDLC documentation process which will be useful in tracking of the project and collecting the metrics of the project which will be an important requirement for the management to improve upon the quality of the Processes followed in the company. The project will ease the employees of the organization by filling the various document of the project in a centralized server with the help of this application. And this will help in collecting the data for CMM assessment.

Finally the main objective of this dissertation is to do some work towards improving the Quality of the process followed in the organization which will ultimately lead to Quality Product. And the application developed will surely do that.

1.3 Scope Of Work

The Application made is an important requirement for the Software Development Organization, which wishes to apply for the CMM Level-3 compliance and above and wants to improve upon the Quality of the Process and the Product. The Scope of the work i.e. the application **Bhilwara Intranet Project For CMM Level-3 Automation** is to automate Processes for the various projects which includes, Resource Allocation, Review defect tracking and closure, Test defect tracking and closure, Audit NC tracking and closure, Requirement Tracking and Online availability of the project closure report, customer feedback form and performance reports. Reports for Review Defects, Testing defects, to be made available.

The main areas that are under the scope of this are as follows:

1. Online Review Defect entry and Summarization.
2. Online Testing defect entry and summarization.
3. Online audit NC Entry .
4. Online availability of Performance Report.
5. Online availability of Customer Feedback Form.
6. Requirement tracking to be made online.
7. Online Resource Allocation.
8. SDLC Wise Document List.
9. KPA Wise Document List.
10. ISO Wise Document List.
11. Implement Role based security for the employees.
12. Online Milestones entry.
13. Online availability of MPP.

Accept to this the application will also implement a new type of security other the usual login security, will also maintain the employee information.

1.4 Organization of the Dissertation

This dissertation aims at making an application which will automate the documentation process in the organization as is already described in the chapter 1 of the Dissertation report. In this chapter the Initially the need for quality is described and then the objective and scope of the dissertation is described.

In the chapter 2, the CMM is describe in detail which is the literature needed for the dissertation work, this chapter has described the various levels of the CMM along with the various KPA's of the different levels.

In chapter 3, the requirements for the application which is being built is jotted down, requirement for the application both the technical as well as the functional requirement are noted down along with the assumptions and dependences.

Chapter 4, describes the design of the application showing the lifecycle chosen for making the application, along with the architecture of the application, Class Diagram, Database Diagram, Data Flow Diagram.

In Chapter 5, the Testing part is described, Chapter 6 describes the result and discussion of the dissertation work done. The last Chapter, i.e. Chapter 7 describes the conclusion after which we have the references and the appendixes.

Chapter 2

LITERATURE SURVEY OF CAPABILITY MATURITY MODEL

2.1 Introducing the Capability Maturity Model

The Capability Maturity Model for Software (CMM) is a framework that describes the key elements of an effective software process. The CMM describes an evolutionary improvement path from an ad hoc, immature process to a mature, disciplined process. The CMM covers practices for planning, engineering, and managing software development and maintenance. When followed, these key practices improve the ability of organizations to meet goals for cost, schedule, functionality, and product quality. The CMM establishes a yardstick against which it is possible to judge, in a repeatable way, the maturity of an organization's software process and compare it to the state of the practice of the industry [2]. The CMM can also be used by an organization to plan improvements to its software process.

2.2 Uses of CMM

- Software process improvement, in which an organization plans, develops, and implements changes to its software process;
- Software process assessments, in which a trained team of software professionals determines the state of an organization's current software process, determines the high-priority software process-related issues facing an organization, and obtains the organizational support for software process improvement; and
- Software capability evaluations, in which a trained team of professionals identifies contractors who are qualified to perform the software work or monitors the state of the software process used on an existing software effort.

2.3 Structure of the CMM

The CMM is composed of five maturity levels. With the exception of Level 1, each maturity level is composed of several key process areas. Each key process area is organized into five sections called common features. The common features specify the key practices that, when collectively addressed, accomplish the goals of the key process area. This structure of the CMM is illustrated in Figure 2.1.

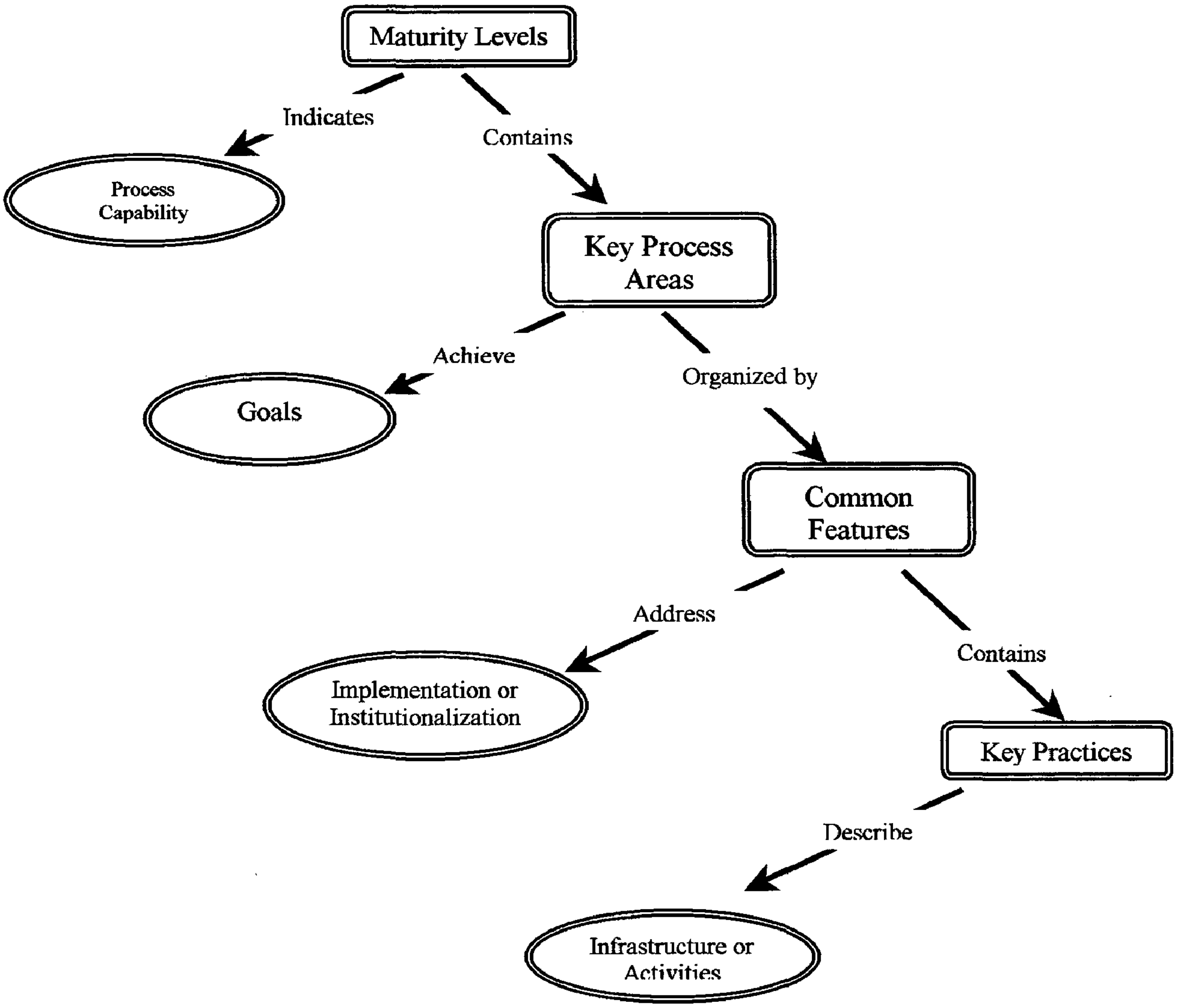


Figure 2.1 The Structure of the Capability Maturity Model.

2.3.1 Maturity Levels

A maturity level is a well-defined evolutionary plateau toward achieving a mature software process. The five maturity levels provide the top-level structure of the CMM.

2.3.2 Process Capability

Software process capability describes the range of expected results that can be achieved by following a software process. The software process capability of an organization provides one means of predicting the most likely outcomes to be expected from the next software project the organization undertakes.

2.3.3 Key Process Areas

Each maturity level is composed of key process areas. Each key process area identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for establishing process capability at that maturity level. The key process areas have been defined to reside at a single maturity level. For example, one of the key process areas for Level 2 is Software Project Planning.

2.3.4 Goals

The goals summarize the key practices of a key process area and can be used to determine whether an organization or project has effectively implemented the key process area. The goals signify the scope, boundaries, and intent of each key process area. In adapting the key practices of a key process area to a specific project situation, the goals can be used to determine whether the adaptation is a reasonable rendering of the practices. Similarly, when assessing or evaluating alternative ways to implement a key process area, the goals can be used to determine if the alternatives satisfy the intent of the key process area. An example of a goal from the Software Project Planning key process area is "Software estimates are documented for use in planning and tracking the software project." See "Capability Maturity Model for Software, Version 1.1" [3] .

2.3.5 Common Features

The key practices in each key process area are organized by a set of common features. The common features are attributes that indicate whether the implementation and institutionalization of a key process area is effective, repeatable, and lasting. The common features also group and order the key practices in a sequence helpful for organizations using them. The five common features are listed below:

Commitment to Perform:-Commitment to Perform describes the actions the organization must take to ensure that the process is established and will endure. Commitment to Perform typically involves establishing organizational policies and senior management sponsorship.

Ability to Perform:-Ability to Perform describes the preconditions that must exist in the project or organization to implement the software process competently. Ability to Perform typically involves resources, organizational structures, and training.

Activities Performed:- Activities Performed describes the roles and procedures necessary to implement a key process area. Activities Performed typically involve establishing plans and procedures, performing the work, tracking it, and taking corrective actions as necessary.

Measurement and Analysis:-Measurement and Analysis describes the need to measure the process and analyze the measurements. Measurement and Analysis typically includes examples of the measurements that could be taken to determine the status and effectiveness of the Activities Performed.

Verifying Implementation:-Verifying Implementation describes the steps to ensure that the activities are performed in compliance with the process that has been established. Verification typically encompasses reviews and audits by management and software quality assurance.

The practices in the common feature Activities Performed describe what must be implemented to establish a process capability. The other practices, taken as a whole, form the basis by which an organization can institutionalize the practices described in the Activities Performed common feature. The Activities Performed by projects or the organization provide the largest category of key practices because they describe the actual implementation of the key process area. Key practices under the other common features are equally important, however, for they address what must be done to support and institutionalize the key process area.

2.3.6 The Key Practices

Each key process area is described in terms of the key practices that contribute to satisfying its goals. The key practices describe the infrastructure and activities that contribute most to the effective implementation and institutionalization of the key process area. Each key practice consists of a single sentence, often followed by a more detailed description, which may include examples and elaboration. These key practices, also referred to as the top-level key practices, state the fundamental policies, procedures, and activities for the key process area. The components of the detailed description are frequently referred to as subpractices. The key practices describe "what" is to be done, but they should not be interpreted as mandating "how" the goals should be achieved.

Alternative practices may accomplish the goals of the key process area. The key practices should be interpreted rationally to judge whether the goals of the key process area are effectively, although perhaps differently, achieved.

2.4 Definition of the CMM Maturity Levels

As organizations establish and improve the software processes by which they develop and maintain their software work products, they progress through levels of maturity. Figure 2.2 shows the five maturity levels of the CMM. Each maturity level provides a layer in the foundation for continuous process improvement. Each key process area comprises a set of goals that, when satisfied, stabilize an important component of the software process. Achieving each level of the maturity model institutionalizes a different component in the software process, resulting in an overall increase in the process capability of the organization.

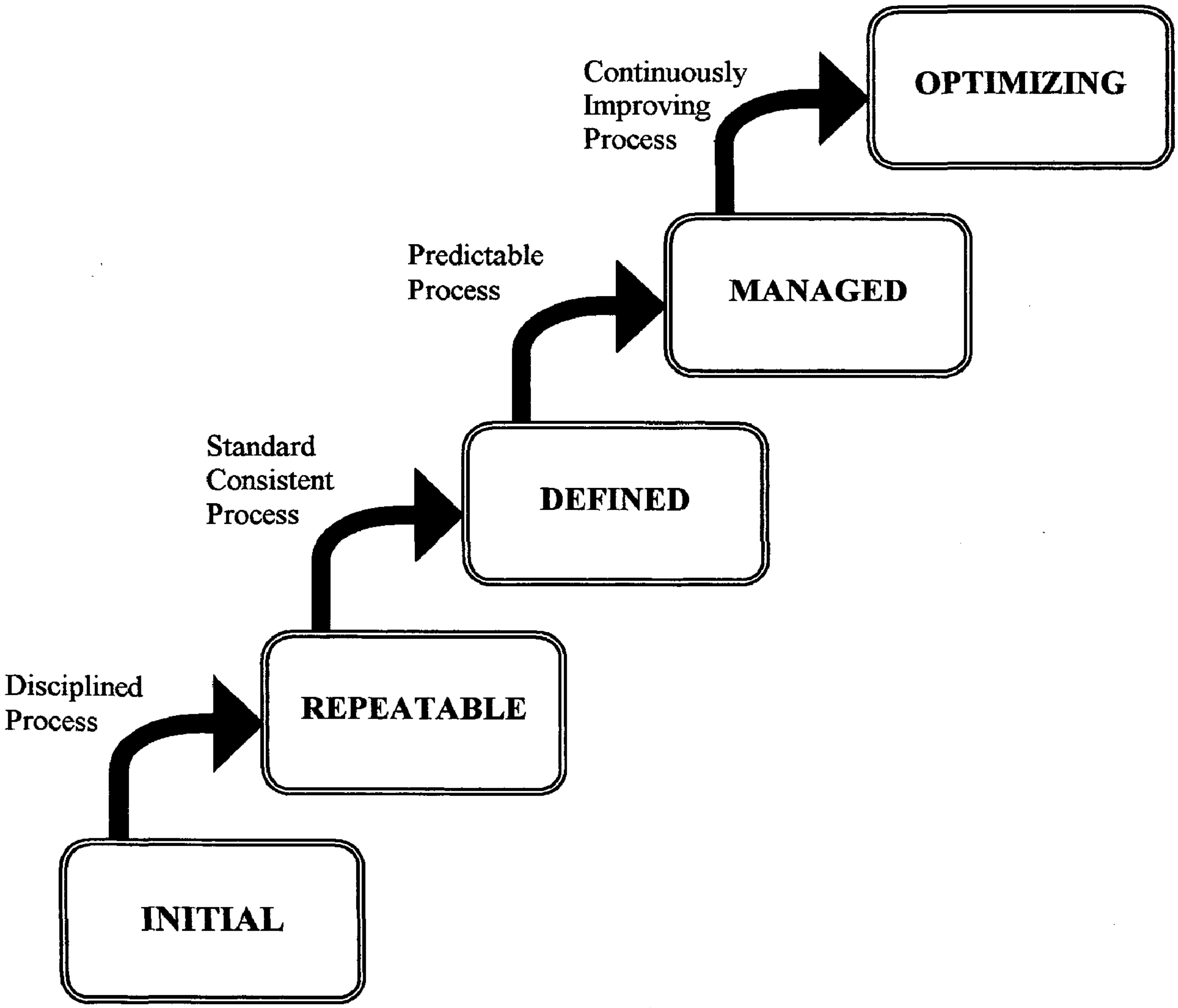


Figure 2.2 The Five Levels of Software Process Maturity.

2.5 Level 1 - The Initial Level

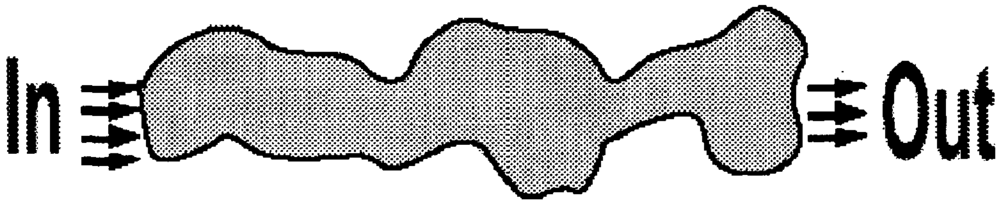


Figure 2.3 Management View of Software Process at Level 1.

At the Initial Level, as depicted in the Figure 2.3 above, the organization typically does not provide a stable environment for developing and maintaining software. When an organization lacks sound management practices, the benefits of good software engineering practices are undermined by ineffective planning and reaction-driven commitment systems. During a crisis, projects typically abandon planned procedures and revert to coding and testing. Success depends entirely on having an exceptional manager and a seasoned and effective software team. Occasionally, capable and forceful software managers can withstand the pressures to take shortcuts in the software process; but when they leave the project, their stabilizing influence leaves with them. Even a strong engineering process cannot overcome the instability created by the absence of sound management practices. The software process capability of Level 1 organizations is unpredictable because the software process is constantly changed or modified as the work progresses (i.e., the process is ad hoc). Schedules, budgets, functionality, and product quality are generally unpredictable. Performance depends on the capabilities of individuals and varies with their innate skills, knowledge, and motivations. There are few stable software processes in evidence, and performance can be predicted only by individual rather than organizational capability.

2.6 Level 2 - The Repeatable Level

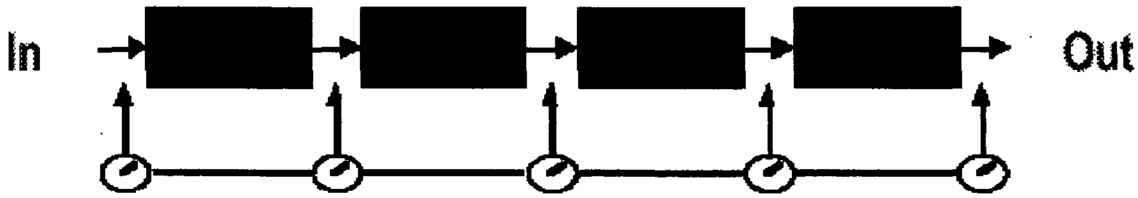


Figure 2.4 Management View of Software Process at Level 2.

At the Repeatable Level, as depicted in the Figure 2.4 above, policies for managing a software project and procedures to implement those policies are established. Planning and managing new projects is based on experience with similar projects. An objective in achieving Level 2 is to institutionalize effective management processes for software projects, which allow organizations to repeat successful practices developed on earlier projects, although the specific processes implemented by the projects may differ. An effective process can be characterized as practiced, documented, enforced, trained, measured, and able to improve. Projects in Level 2 organizations have installed basic software management controls. Realistic project commitments are based on the results observed on previous projects and on the requirements of the current project. The software managers for a project track software costs, schedules, and functionality; problems in meeting commitments are identified when they arise. Software requirements and the work products developed to satisfy them are baselined, and their integrity is controlled. Software project standards are defined, and the organization ensures they are faithfully followed. The software project works with its subcontractors, if any, to establish a strong customer-supplier relationship. The software process capability of Level 2 organizations can be summarized as disciplined because planning and tracking of the software project is stable and earlier successes can be repeated. The project's process is under the effective control of a project management system, following realistic plans based on the performance of previous projects.

2.7 Level 3 - The Defined Level

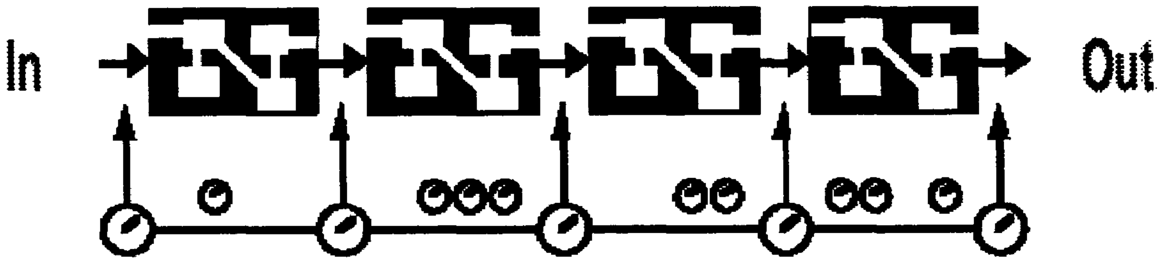


Figure 2.5 Management View of Software Process at Level 3.

At the Defined Level, as depicted in the figure 2.5 above, the standard process for developing and maintaining software across the organization is documented, including both software engineering and management processes, and these processes are integrated into a coherent whole. This standard process is referred to throughout the CMM as the organization's standard software process. Processes established at Level 3 are used (and changed, as appropriate) to help the software managers and technical staff perform more effectively. The organization exploits effective software engineering practices when standardizing its software processes. There is a group that is responsible for the organization's software process activities, e.g., a software engineering process group, or SEPG [4]. An organization-wide training program is implemented to ensure that the staff and managers have the knowledge and skills required to fulfill their assigned roles. Projects tailor the organization's standard software process to develop their own defined software process, which accounts for the unique characteristics of the project. This tailored process is referred to in the CMM as the project's defined software process. A defined software process contains a coherent, integrated set of well-defined software engineering and management processes. A well-defined process can be characterized as including readiness criteria, inputs, standards and procedures for performing the work, verification mechanisms (such as peer reviews), outputs, and completion criteria. Because the software process is well defined, management has good insight into technical progress on all projects. The software process capability of Level 3 organizations can be summarized as standard and consistent because both software engineering and management activities are stable and repeatable. Within established product lines, cost, schedule, and functionality are under control, and software quality is tracked. This process capability is based on a common, organization-wide understanding of the activities, roles, and responsibilities in a defined software process.

2.8 Level 4 - The Managed Level

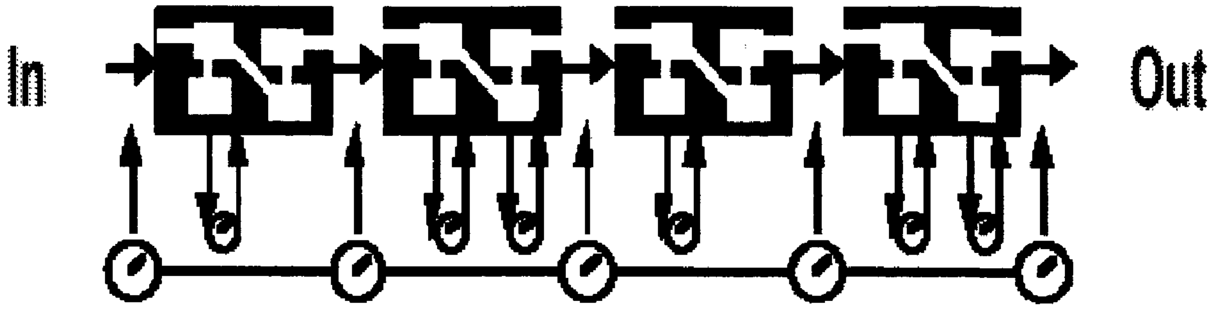


Figure 2.6 Management View of Software Process at Level 4.

At the Managed Level, as depicted in the figure 2.6 above, the organization sets quantitative quality goals for both software products and processes. Productivity and quality are measured for important software process activities across all projects as part of an organizational measurement program. An organization-wide software process database is used to collect and analyze the data available from the projects' defined software processes. Software processes are instrumented with well-defined and consistent measurements at Level 4. These measurements establish the quantitative foundation for evaluating the projects' software processes and products. Projects achieve control over their products and processes by narrowing the variation in their process performance to fall within acceptable quantitative boundaries. Meaningful variations in process performance can be distinguished from random variation (noise), particularly within established product lines. The risks involved in moving up the learning curve of a new application domain are known and carefully managed. The software process capability of Level 4 organizations can be summarized as predictable because the process is measured and operates within measurable limits. This level of process capability allows an organization to predict trends in process and product quality within the quantitative bounds of these limits. When these limits are exceeded, action is taken to correct the situation. Software products are of predictably high quality.

2.9 Level 5 - The Optimizing Level

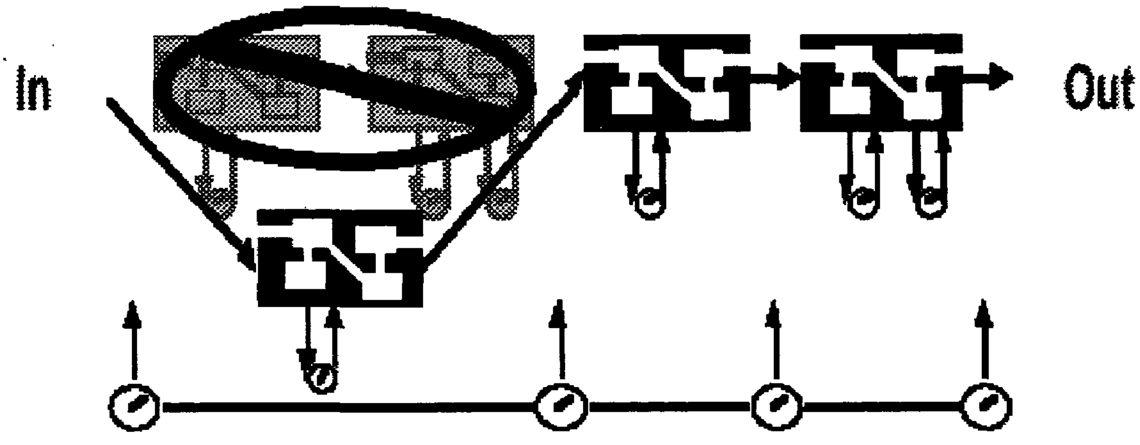
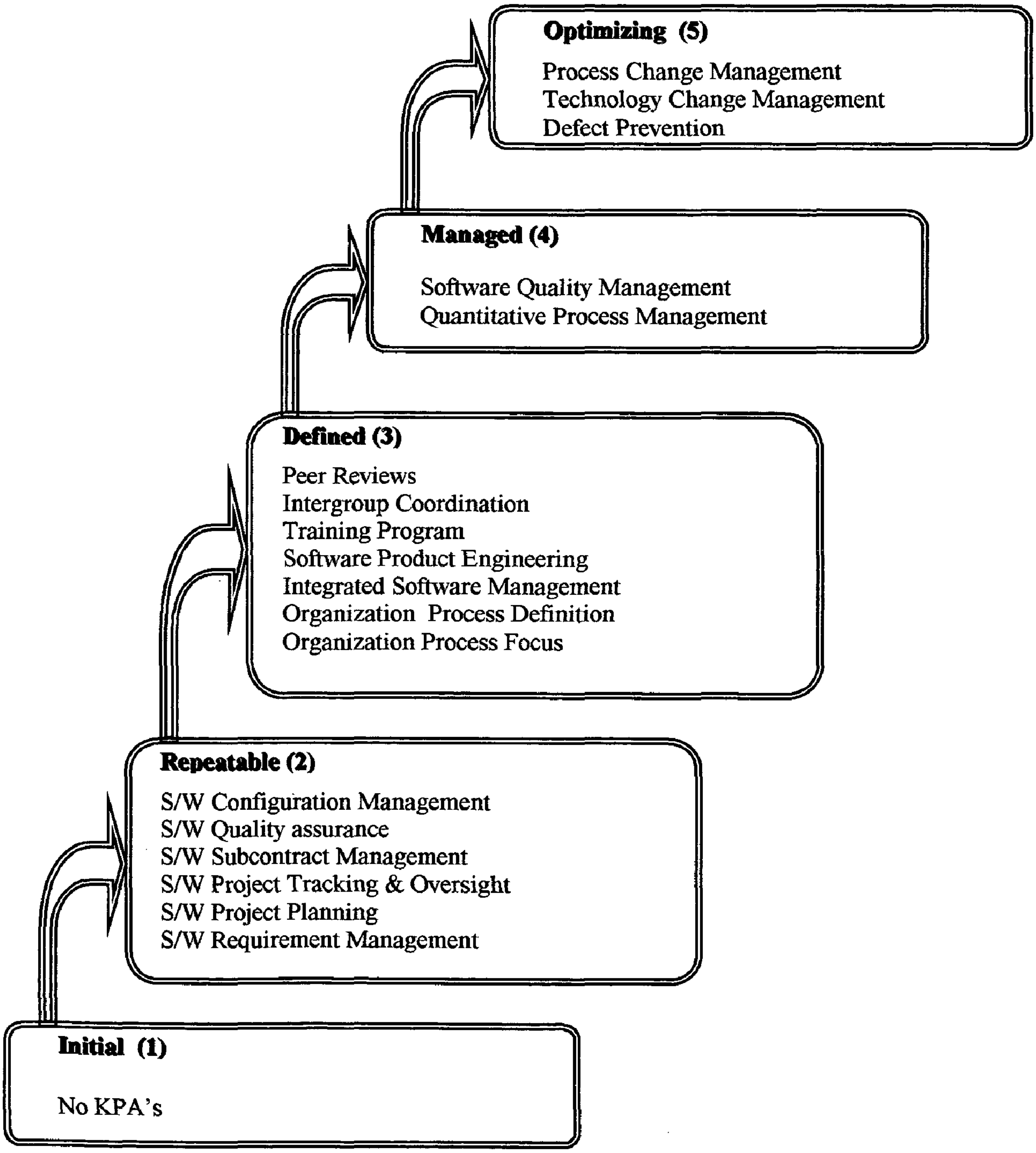


Figure 2.7 Management View of Software Process at Level 5.

At the Optimizing Level, as depicted in the figure 2.7 above, the entire organization is focused on continuous process improvement. The organization has the means to identify weaknesses and strengthen the process proactively, with the goal of preventing the occurrence of defects. Data on the effectiveness of the software process is used to perform cost benefit analyses of new technologies and proposed changes to the organization's software process. Innovations that exploit the best software engineering practices are identified and transferred throughout the organization. Software project teams in Level 5 organizations analyze defects to determine their causes. Software processes are evaluated to prevent known types of defects from recurring, and lessons learned are disseminated to other projects. The software process capability of Level 5 organizations can be characterized as continuously improving because Level 5 organizations are continuously striving to improve the range of their process capability, thereby improving the process performance of their projects. Improvement occurs both by incremental advancements in the existing process and by innovations using new technologies and methods.

2.10 The Key Process Areas in CMM

Figure 2.8 lists the key process areas for each maturity level in the CMM. The key process areas are building blocks that indicate the areas an organization should focus on to improve its software process. Key process areas identify the issues that must be addressed to achieve a maturity level.



Optimizing (5)

Process Change Management
Technology Change Management
Defect Prevention

Managed (4)

Software Quality Management
Quantitative Process Management

Defined (3)

Peer Reviews
Intergroup Coordination
Training Program
Software Product Engineering
Integrated Software Management
Organization Process Definition
Organization Process Focus

Repeatable (2)

S/W Configuration Management
S/W Quality assurance
S/W Subcontract Management
S/W Project Tracking & Oversight
S/W Project Planning
S/W Requirement Management

Initial (1)

No KPA's

Figure 2.8 The Key Process Areas by Maturity Level.

2.11 The Key Process Areas of the CMM Level 2

2.11.1 Requirements Management

The purpose of Requirements Management is to establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project. Requirements Management involves establishing and maintaining an agreement with the customer on the requirements for the software project. This agreement is referred to as the "system requirements allocated to the software." The "customer" may be interpreted as the system engineering group, the marketing group, another internal organization, or an external customer. The agreement covers both the technical and no technical (e.g., delivery dates) requirements. The agreement forms the basis for estimating, planning, performing, and tracking the software project's activities throughout the software life cycle. The allocation of the system requirements to software, hardware, and other system components (e.g., humans) may be performed by a group external to the software engineering group (e.g., the system engineering group), and the software engineering group may have no direct control of this allocation. Within the constraints of the project, the software engineering group takes appropriate steps to ensure that the system requirements allocated to software, which they are responsible for addressing, are documented and controlled. To achieve this control, the software engineering group reviews the initial and revised system requirements allocated to software to resolve issues before they are incorporated into the software project. Whenever the system requirements allocated to software are changed, the affected software plans, work products, and activities are adjusted to remain consistent with the updated requirements.

Goal 1:- System requirements allocated to software are controlled to establish a baseline for software engineering and management use.

Goal 2:-Software plans, products, and activities are kept consistent with the system requirements allocated to software.

2.11.2 Software Project Planning

The purpose of Software Project Planning is to establish reasonable plans for performing the software engineering and for managing the software project. Software Project Planning involves developing estimates for the work to be performed, establishing the necessary commitments, and defining the plan to perform the work. The software planning begins with a statement of the work to be performed and other constraints and goals that define and bound the software project (those established by the practices of the Requirements Management key process area). The software planning process includes steps to estimate the size of the software work products and the resources needed, produce a schedule, identify and assess software risks, and negotiate commitments. Iterating through these steps may be necessary to establish the plan for the software project (i.e., the software development plan). This plan provides the basis for performing and managing the software project's activities and addresses the commitments to the software project's customer according to the resources, constraints, and capabilities of the software project.

Goal 1:- Software estimates are documented for use in planning and tracking the software project.

Goal 2 :- Software project activities and commitments are planned and documented.

Goal 3:- Affected groups and individuals agree to their commitments related to the software project.

2.11.3 Software Project Tracking and Oversight

The purpose of Software Project Tracking and Oversight is to provide adequate visibility into actual progress so that management can take effective actions when the software project's performance deviates significantly from the software plans. Software Project Tracking and Oversight involves tracking and reviewing the software accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results. A documented plan for the software project (i.e., the software development plan, as described in the Software Project Planning key process area) is used as the basis for tracking the software activities, communicating status, and revising plans. Software activities are monitored by the management. Progress is primarily determined by comparing the actual software size, effort, cost, and schedule to the plan when selected software work products are completed and at selected milestones. When it is determined that the software project's plans are not being met, corrective actions are taken. These actions may include revising the software development plan to reflect the actual accomplishments and replanning the remaining work or taking actions to improve the performance.

Goal 1:- Actual results and performances are tracked against the software plans.

Goal 2:- Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans.

Goal 3:- Changes to software commitments are agreed to by the affected groups and individuals.

2.11.4 Software Subcontract Management

The purpose of Software Subcontract Management is to select qualified software subcontractors and manage them effectively. Software Subcontract Management involves selecting a software subcontractor, establishing commitments with the subcontractor, and tracking and reviewing the subcontractor's performance and results. These practices cover the management of a software (only) subcontract, as well as the management of the software component of a subcontract that includes software, hardware, and possibly other system components. The subcontractor is selected based on its ability to perform the work. Many factors contribute to the decision to subcontract a portion of the prime contractor's work. Subcontractors may be selected based on strategic business alliances, as well as technical considerations. The practices of this key process area address the traditional acquisition process associated with subcontracting a defined portion of the work to another organization. When subcontracting, a documented agreement covering the technical and no technical (e.g., delivery dates) requirements is established and is used as the basis for managing the subcontract. The work to be done by the subcontractor and the plans for the work are documented. The standards that are to be followed by the subcontractor are compatible with the prime contractor's standards. The software planning, tracking, and oversight activities for the subcontracted work are performed by the subcontractor. The prime contractor ensures that these planning, tracking, and oversight activities are performed appropriately and that the software products delivered by the subcontractor satisfy their acceptance criteria. The prime contractor works with the subcontractor to manage their product and process interfaces.

Goal 1:- The prime contractor selects qualified software subcontractors.

Goal 2:- The prime contractor and the software subcontractor agree to their commitments to each other.

Goal 3:-The prime contractor and the software subcontractor maintain ongoing communications.

Goal 4:- The prime contractor tracks the software subcontractor's actual results and performance against its commitments.

2.11.5 Software Quality Assurance

The purpose of Software Quality Assurance is to provide management with appropriate visibility into the process being used by the software project and of the products being built. Software Quality Assurance involves reviewing and auditing the software products and activities to verify that they comply with the applicable procedures and standards and providing the software project and other appropriate managers with the results of these reviews and audits. The software quality assurance group works with the software project during its early stages to establish plans, standards, and procedures that will add value to the software project and satisfy the constraints of the project and the organization's policies. By participating in establishing the plans, standards, and procedures, the software quality assurance group helps ensure they fit the project's needs and verifies that they will be usable for performing reviews and audits throughout the software life cycle. The software quality assurance group reviews project activities and audits software work products throughout the life cycle and provides management with visibility as to whether the software project is adhering to its established plans, standards, and procedures. Compliance issues are first addressed within the software project and resolved there if possible. For issues not resolvable within the software project, the software quality assurance group escalates the issue to an appropriate level of management for resolution. This key process area covers the practices for the group performing the software quality assurance function. The practices identifying the specific activities and work products that the software quality assurance group reviews and/or audits are generally contained in the Verifying Implementation common feature of the other key process areas.

Goal 1:- Software quality assurance activities are planned.

Goal 2:- Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.

Goal 3:- Affected groups and individuals are informed of software quality assurance activities and results.

Goal 4:- Noncompliance issues that cannot be resolved within the software project are addressed by senior management.

2.11.6 Software Configuration Management

The purpose of Software Configuration Management is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle. Software Configuration Management involves identifying the configuration of the software (i.e., selected software work products and their descriptions) at given points in time, systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration throughout the software life cycle. The work products placed under software configuration management include the software products that are delivered to the customer (e.g., the software requirements document and the code) and the items that are identified with or required to create these software products (e.g., the compiler). A software baseline library is established containing the software baselines as they are developed. Changes to baselines and the release of software products built from the software baseline library are systematically controlled via the change control and configuration auditing functions of software configuration management. This key process area covers the practices for performing the software configuration management function. The practices identifying specific configuration items/units are contained in the key process areas that describe the development and maintenance of each configuration item/unit.

Goal 1:- Software configuration management activities are planned.

Goal 2:- Selected software work products are identified, controlled, and available.

Goal 3:- Changes to identified software work products are controlled.

Goal 4:-Affected groups and individuals are informed of the status and content of software baselines.

2.12 The Key Process Areas of the CMM Level 3

2.12.1 Organization Process Focus

The purpose of Organization Process Focus is to establish the organizational responsibility for software process activities that improve the organization's overall software process capability. Organization Process Focus involves developing and maintaining an understanding of the organization's and projects' software processes and coordinating the activities to assess, develop, maintain, and improve these processes. The organization provides the long-term commitments and resources to coordinate the development and maintenance of the software processes across current and future software projects via a group such as a software engineering process group. This group is responsible for the organization's software process activities. It is specifically responsible for the development and maintenance of the organization's standard software process and related process assets (as described in the Organization Process Definition key process area), and it coordinates the process activities with the software projects.

Goal 1:- Software process development and improvement activities are coordinated across the organization.

Goal 2:- The strengths and weaknesses of the software processes used are identified relative to a process standard.

Goal 3:- Organization-level process development and improvement activities are planned.

2.12.2 Organization Process Definition

The purpose of Organization Process Definition is to develop and maintain a usable set of software process assets that improve process performance across the projects and provide a basis for cumulative, long-term benefits to the organization. Organization Process Definition involves developing and maintaining the organization's standard software process, along with related process assets, such as descriptions of software life cycles, process tailoring guidelines and criteria, the organization's software process database, and a library of software process-related documentation. These assets may be collected in many ways, depending on the organization's implementation of Organization Process Definition. For example, the descriptions of the software life cycles may be an integral part of the organization's standard software process or parts of the library of software process-related documentation may be stored in the organization's software process database. The organization's software process assets are available for use in developing, implementing, and maintaining the projects' defined software processes. (The practices related to the development and maintenance of the project's defined software process are described in the Integrated Software Management key process area.)

Goal 1:- A standard software process for the organization is developed and maintained.

Goal 2:- Information related to the use of the organization's standard software process by the software projects is collected, reviewed, and made available.

2.12.3 Training Program

The purpose of the Training Program key process area is to develop the skills and knowledge of individuals so they can perform their roles effectively and efficiently. Training Program involves first identifying the training needed by the organization, projects, and individuals, then developing or procuring training to address the identified needs. Each software project evaluates its current and future skill needs and determines how these skills will be obtained. Some skills are effectively and efficiently imparted through informal vehicles (e.g., on-the-job training and informal mentoring), whereas other skills need more formal training vehicles (e.g., classroom training and guided self-study) to be effectively and efficiently imparted. The appropriate vehicles are selected and used. This key process area covers the practices for the group performing the training function. The practices identifying the specific training topics (i.e., knowledge or skill needed) are contained in the Ability to Perform common feature of the individual key process areas.

Goal 1:- Training activities are planned.

Goal 2:- Training for developing the skills and knowledge needed to perform software management and technical roles is provided.

Goal 3:-Individuals in the software engineering group and software-related groups receive the training necessary to perform their roles.

2.12.4 Integrated Software Management

The purpose of Integrated Software Management is to integrate the software engineering and management activities into a coherent, defined software process that is tailored from the organization's standard software process and related process assets, which are described in Organization Process Definition. Integrated Software Management involves developing the project's defined software process and managing the software project using this defined software process. The project's defined software process is tailored from the organization's standard software process to address he specific characteristics of the project. The software development plan is based on the project's defined software process and describes how the activities of the project's defined software process will be implemented and managed. The management of the software project's size, effort, cost, schedule, staffing, and other resources is tied to the tasks of the project's defined software process. Since the projects' defined software processes are all tailored from the organization's standard software process, the software projects can share process data and lessons learned. The basic practices for estimating, planning, and tracking a software project are described in the Software Project Planning and Software Project Tracking and Oversight key process areas. They focus on recognizing problems when they occur and adjusting the plans and/or performance to address the problems. The practices of this key process area build on, and are in addition to, the practices of those two key process areas. The emphasis of Integrated Software Management shifts to anticipating problems and acting to prevent or minimize the effects of these problems.

Goal 1:- The project's defined software process is a tailored version of the organization's standard software process.

Goal 2:- The project is planned and managed according to the project's defined software process.

2.12.5 Software Product Engineering

The purpose of Software Product Engineering is to consistently perform a well-defined engineering process that integrates all the software engineering activities to produce correct, consistent software products effectively and efficiently. Software Product Engineering involves performing the engineering tasks to build and maintain the software using the project's defined software process (which is described in the Integrated Software Management key process area) and appropriate methods and tools. The software engineering tasks include analyzing the system requirements allocated to software (these system requirements are described in the Requirements Management key process area), developing the software requirements, developing the software architecture, designing the software, implementing the software in the code, integrating the software components, and testing the software to verify that it satisfies the specified requirements (i.e., the system requirements allocated to software and the software requirements). Documentation needed to perform the software engineering tasks (e.g., software requirements document, software design document, test plan, and test procedures) is developed and reviewed to ensure that each task addresses the results of predecessor tasks and the results produced are appropriate for the subsequent tasks (including the tasks of operating and maintaining the software). When changes are approved, affected software work products, plans, commitments, processes, and activities are revised to reflect the approved changes.

Goal 1:-The software engineering tasks are defined, integrated, and consistently performed to produce the software.

Goal 2:- Software work products are kept consistent with each other.

2.12.6 Inter-group Coordination

The purpose of Intergroup Coordination is to establish a means for the software engineering group to participate actively with the other engineering groups so the project is better able to satisfy the customer's needs effectively and efficiently. Intergroup Coordination involves the software engineering group's participation with other project engineering groups to address system-level requirements, objectives, and issues. Representatives of the project's engineering groups participate in establishing the system-level requirements, objectives, and plans by working with the customer and end users, as appropriate. These requirements, objectives, and plans become the basis for all engineering activities. The technical working interfaces and interactions between groups are planned and managed to ensure the quality and integrity of the entire system. Technical reviews and interchanges are regularly conducted with representatives of the project's engineering groups to ensure that all engineering groups are aware of the status and plans of all the groups, and that system and intergroup issues receive appropriate attention. The software-specific practices related to these engineering tasks are described in the Requirements Management and Software Product Engineering key process areas.

Goal 1:- The customer's requirements are agreed to by all affected groups .

Goal 2:- The commitments between the engineering groups are agreed to by the affected groups.

Goal 3:- The engineering groups identify, track, and resolve intergroup issues.

2.12.7 Peer Reviews

The purpose of Peer Reviews is to remove defects from the software work products early and efficiently. An important corollary effect is to develop a better understanding of the software work products and of defects that might be prevented [5].

Peer Reviews involve a methodical examination of software work products by the producers' peers to identify defects and areas where changes are needed. The specific products that will undergo a peer review are identified in the project's defined software process and scheduled as part of the software project planning activities, as described in Integrated Software Management. This key process area covers the practices for performing peer reviews. The practices identifying the specific software work products that undergo peer review are contained in the key process areas that describe the development and maintenance of each software work product. The peer review is an important and effective engineering method that is called out in Software Product Engineering and that can be implemented via Fagan-style inspections [6], structured walkthroughs, or a number of other collegial review methods [7]

Goal 1:- Peer review activities are planned.

Goal 2:-Defects in the software work products are identified and removed.

2.13 The Key Process Areas of the CMM Level 4

2.13.1 Quantitative Process Management

The purpose of Quantitative Process Management is to control the process performance of the software project quantitatively. Software process performance represents the actual results achieved from following a software process. Quantitative Process Management involves establishing goals for the performance of the project's defined software process, which is described in the Integrated Software Management key process area, taking measurements of the process performance, analyzing these measurements, and making adjustments to maintain process performance within acceptable limits. When the process performance is stabilized within acceptable limits, the project's defined software process, the associated measurements, and the acceptable limits for the measurements are established as a baseline and used to control process performance quantitatively. The organization collects process performance data from the software projects and uses these data to characterize the process capability (i.e., the process performance a new project can expect to attain) of the organization's standard software process, which is described in the Organization Process Definition key process area. Process capability describes the range of expected results from following a software process (i.e., the most likely outcomes that are expected from the next software project the organization undertakes). These process capability data are, in turn, used by the software projects to establish and revise their process performance goals and to analyze the performance of the projects' defined software processes.

Goal 1:- The quantitative process management activities are planned.

Goal 2:- The process performance of the project's defined software process is controlled quantitatively.

Goal 3:- The process capability of the organization's standard software process is known in quantitative terms.

2.13.2 Software Quality Management

The purpose of Software Quality Management is to develop a quantitative understanding of the quality of the project's software products and achieve specific quality goals. Software Quality Management involves defining quality goals for the software products, establishing plans to achieve these goals, and monitoring and adjusting the software plans, software work products, activities, and quality goals to satisfy the needs and desires of the customer and end user for high quality products. The practices of Software Quality Management build on the practices of the Integrated Software Management and Software Product Engineering key process areas, which establish and implement the project's defined software process, and the Quantitative Process Management key process area, which establishes a quantitative understanding of the ability of the project's defined software process to achieve the desired results. Quantitative goals are established for the software products based on the needs of the organization, the customer, and the end users. So that these goals may be achieved, the organization establishes strategies and plans, and the project specifically adjusts its defined software process, to accomplish the quality goals.

Goal 1:- The project's software quality management activities are planned.

Goal 2:- Measurable goals for software product quality and their priorities are defined.

Goal 3:- Actual progress toward achieving the quality goals for the software products is quantified and managed.

2.14 The Key Process Areas of the CMM Level 5

2.14.1 Defect Prevention

The purpose of Defect Prevention is to identify the cause of defects and prevent them from recurring. Defect Prevention involves analyzing defects that were encountered in the past and taking specific actions to prevent the occurrence of those types of defects in the future. The defects may have been identified on other projects as well as in earlier stages or tasks of the current project. Defect prevention activities are also one mechanism for spreading lessons learned between projects. Trends are analyzed to track the types of defects that have been encountered and to identify defects that are likely to recur. Based on an understanding of the project's defined software process and how it is implemented (as described in the Integrated Software Management and Software Product Engineering key process areas), the root causes of the defects and the implications of the defects for future activities are determined. Both the project and the organization take specific actions to prevent recurrence of the defects. Some of the organizational actions may be handled as described in the Process Change Management key process area.

Goal 1:- Defect prevention activities are planned.

Goal 2:- Common causes of defects are sought out and identified.

Goal 3:- Common causes of defects are prioritized and systematically eliminated.

2.14.2 Technology Change Management

The purpose of Technology Change Management is to identify new technologies (i.e., tools, methods, and processes) and track them into the organization in an orderly manner. Technology Change Management involves identifying, selecting, and evaluating new technologies, and incorporating effective technologies into the organization. The objective is to improve software quality, increase productivity, and decrease the cycle time for product development. The organization establishes a group (such as a software engineering process group or a technology support group) that works with the software projects to introduce and evaluate new technologies and manage changes to existing technologies. Particular emphasis is placed on technology changes that are likely to improve the capability of the organization's standard software process (as described in the Organization Process Definition key process area). By maintaining an awareness of software-related technology innovations and systematically evaluating and experimenting with them, the organization selects appropriate technologies to improve the quality of its software and the productivity of its software activities. Pilot efforts are performed to assess new and unproven technologies before they are incorporated into normal practice. With appropriate sponsorship of the organization's management, the selected technologies are incorporated into the organization's standard software process and current projects, as appropriate. Changes to the organization's standard software process (as described in the Organization Process Definition key process area) and the projects' defined software processes (as described in the Integrated Software Management key process area) resulting from these technology changes are handled as described in the Process Change Management key process area.

Goal 1:- Incorporation of technology changes are planned.

Goal 2:-New technologies are evaluated to determine their effect on quality and productivity.

Goal 3:-Appropriate new technologies are transferred into normal practice across the organization.

2.14.3 Process Change Management

The purpose of Process Change Management is to continually improve the software processes used in the organization with the intent of improving software quality, increasing productivity, and decreasing the cycle time for product development. Process Change Management involves defining process improvement goals and, with senior management sponsorship, proactively and systematically identifying, evaluating, and implementing improvements to the organization's standard software process and the projects' defined software processes on a continuous basis. Training and incentive programs are established to enable and encourage everyone in the organization to participate in process improvement activities. Improvement opportunities are identified and evaluated for potential payback to the organization. Pilot efforts are performed to assess process changes before they are incorporated into normal practice. When software process improvements are approved for normal practice, the organization's standard software process and the projects' defined software processes are revised as appropriate.

The practices for revising the organization's standard software process are found in the Organization Process Definition key process area, and the practices for revising the projects' defined software processes are found in the Integrated Software Management key process area.

Goal 1:- Continuous process improvement is planned.

Goal 2:- Participation in the organization's software process improvement activities is organization wide.

Goal 3:- The organization's standard software process and the projects' defined software processes are improved continuously.

Chapter 3

REQUIRMENT SPECIFICATION

3.1 Aim of the Thesis

The aim to make an Intranet software application (Bhilwara Intranet Project For CMM Level 3 Automation) as a dissertation work for the MTech IT course are as follows.

- To study the Capability Maturity Model in detail. To the study and implement the various process involved SDLC(Software Development Life Cycle) and mapping the processes to CMM KPA's in an software Development Organization.
- Based on the standard process described in the QMS(Quality Management System) automate whole part of the SDLC documentation process which will be useful in tracking of the project which can be an important requirement for the management.
- The project will also be useful for the organization to collect metrics to apply for CMM Level 3 and above.
- The project will ease the employees of the organization by filling the various document of the project in a centralized server with the help of this application. And this will help in collecting the data for CMM assessment.

3.2 Scope Of Work

The Application made is an important requirement for the Software Development Organization, which wishes to apply for the CMM Level-3 compliance and above. The Scope of the **Bhilwara Intranet Project For CMM Level 3 Automation** is to automate Processes for the various projects which includes, Resource Allocation, Review defect tracking and closure, Test defect tracking and closure, Audit NC tracking and closure, Requirement Tracking and Online availability of the project closure report, customer feedback form and performance reports. Reports for Review Defects, Testing defects, to be made available.

The main areas that are under the scope of this are as follows :

1. Online Review Defect entry and Summarization
2. Online Testing defect entry and summarization
3. Online audit NC Entry
4. Online availability of Performance Report
5. Online availability of Customer Feedback Form.
6. Requirement tracking to be made online
7. Online Resource Allocation.
8. SDLC Wise Document List.
9. KPA Wise Document List.
10. ISO Wise Document List.
11. Implement Role based security for the employees
12. Online Milestones entry.
13. Online availability of MPP

3.3 Technical Requirements

The technical requirement of the intranet based application are listed below :

1. Web Server	**:-**	**Microsoft IIS-5.0 web server**
2. DBMS	**:-**	**Microsoft SQL Server 2000**
3. Operating System	**:-**	**Windows 2000**
4. IDE	**:-**	**Microsoft's .Net Framework**
5. Language	**:-**	**C#,ASP.NET**
6. Web Browser	**:-**	**DOTNET compliant web browser.**

3.4 Functional Requirement

This subsection provides a summary of the major functions that the software will perform. The function listed below are the functional requirement of the application which is being developed keeping in mind the KPA's of different maturity levels being satisfied.

1. Employee Authentication
2. Employee Profile
3. Change Password
4. Master Maintenance
5. Project/ Phases Creation and Approval mechanism.
6. Employee Creation and Maintenance
7. Security
8. Online review defect entry and summarization
9. Online testing defect entry and summarization
10. Online audit NC entry
11. Online availability of Customer Feedback Form
12. Online availability of Performance reports
13. Requirement tracking to be made online for the various projects
14. Online Resource Allocation.
15. Online Availability of MPP
16. Online availability of the Project Reports
17. Online availability of Project Milestone .

1. Employee Authentication

This is the basic functionality, which is required by almost all the application today. In this the user will be asked for his credentials i.e. his UserID and Password .On entering the correct credentials he will be able to Login the application There is also a for some specific time.

2. Employee Profile

The employee logged in will be able to view his profile and update it . the profile is categorized into many parts as :

Personal Information
Qualification
Experience
Skill
Location

3. Change Password

The employee logged in will be able to change his password.

4. Master Maintenance

There are information which will be used by the other modules as master information. So there is a provision to enter the master information in a separate module. This functionality will be assessed by the administrator.

5. Project/ Phases Creation and Maintenance

This is the most important function of the application .The right to access this lies with the System Administrator and some specific Role. The person creating the project for the the first time can not just only fill the information about the project but also, he can select the list of document which he will be requiring in the project. There will be three type of document list which will be available to the person creating the project. They are :-

SDLC wise document list
KPA wise document list
ISO wise document list

Once the person has selected the list of documents. It will be approved by the technical head and then by the SEPG head. After which the document list for the project will be finalized, which will be assessable to the users who have entered in the project detail.

6. Employee Creation and Maintenance

The Application will also be handling the information about all the employees based on their.

Personal Information
Experience
Skill
Location
Qualification

The right of employee creation, modification and renaming lies with the System Administrator.

7. Security

Security is also another important issue for the application as the access rights to the specific module will depend upon the user role. The right for maintaining security will lie only with System Administrator.

There will be two type of security provided in the application :

Role Based Security :- Which will show only the accessible parts of the application to the user based on his role and level. The administrator will be able to change the security permission.

User Based Security :- The user will be able to see the parts in which he is involved i.e. the project in which he is involved.

8. Online review Defect entry and Summarization

The user should be able to enter/update/delete the review defects and summarization of review defects to be made online:

Review defect log to be converted online. Defects to be categorized based on their severity. Also track the closure of the review defect, root cause of the defect and corrective action that was taken to close the review defect.

Review Summary Sheet to be converted online. Defects of all the items reviewed

9. Online testing defect entry and summarization

The user should be able to enter/update/delete the testing defects and results online. Summarization of testing defects to be made online.

Test Incident Report to be converted online. Testing defects to be categorized based on their severity. Also track the closure of the testing defect, root cause of the defect and corrective action that was taken to close the testing defect.

Test Summary Sheet to be converted online.

10. Online audit NC, KPA Wise Analysis and Root Cause Analysis

The user to be able to enter audits NCs online for the various projects and to do a KPA Wise Analysis with regards to the NCs.

Audit Finding report to be converted online. Audit findings to be categorized based on their root cause and KPA wise. Audit findings to be categorized based on their non-conformance of the particular KPA and Procedure. Also track the closure, root cause and corrective action that was taken to close the Audit findings.

11. Online availability of Customer Feedback

The user will be able to upload the Customer feedback form of the project which he has received from the customer.

12. Online availability of reports

The user will be able to upload the Performance report of the project.

13. Requirement tracking to be made online for the various projects.

Tracing of requirement will be able to done online for all the project and there will be an option to tailor the process.

14. Resource Allocation to be made online for all the project.

There will be an option to allocate the resources to the project and deallocate the resources from the project .

15. Online Availability of MPP .

There will be an option to upload the Microsoft Project Plan for the project.

16. Online availability of the Project Reports.

To support metrics group in collecting the metrics more effectively as well as for the organization's continuous improvement the following type of reports are generated:

Milestones
Test Summary
Review Summary
Customer Feedback Form
Performance Report
Project Plan

17. Online availability of Project Milestone.

There will be an option to maintain the milestones of the projects and also update them.

3.5 User Characteristics

The intended users of the application are pre-defined users. All the pages of the application will be secure pages and access will be available through proper validation process like username and password. All the users should know how to surf the Internet using different browsers.

The User of the application will be :

SA –System Administrator

SPM-Senior Project Manager

QP-Quality Person

PM-Project Manager

PL-Project Leader

TM-Team Member

RE-Reviewer

RS-Revolver

The user can be increased depending upon the requirement.

3.6 Performance / Acceptance Requirements

The system database will support 100 users working simultaneously on different consoles. The web server will support 100 concurrent connections / users.

3.7 Assumptions And Dependencies

The application assumes that users will be using Microsoft's Internet Explorer 5 and above. If visitor is using any other browser then system may behave differently.

DESIGN AND ANALYSIS

4.1 Life Cycle Chosen

The life cycle chosen for the application is the Linear Sequential Model which is also termed as classical life cycle or waterfall model. Below Figure 4.1 depicts the Linear Sequential Model.

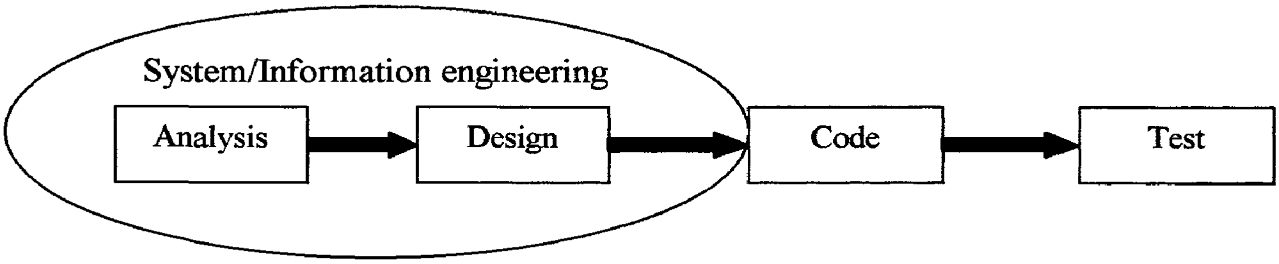


Figure 4.1 Linear Sequential Model.

The various Phases of the Model chosen above are as follows.

REQUIRMENT:- The Requirement gathering process is intensified and focused specifically on software. To understand the nature of the program to be built , the software engineer must understand the information domain for the software as well as required function, behavior, performance and interface.

DESIGN:- Software design is actually a multistep process that focuses on four distinct attributes of a program: data structures, software architecture, interface representation and procedural detail. The design process translates requirement into a representation of the software that can be accessed for quality before coding begins.

CODE GENERATION:-The design must be translated into a machine readable form. The code generation step performs this task. The Design is performed in a detail manner, code manner , code generation can be accomplished mechanistically.

TESTING:- Once code has been generated program testing begins. The testing process focuses on the logical internal of the software, ensuing that all statements have been tested, and on the functional externals i.e. conducting test to uncover error and ensures that defied input will produce actual results that agree required results.

SUPPORT:- software will undoubtedly undergo change after it is delivered to the customer. Change will occur because error have been encountered, because the software must be adapted to accommodate changes in its external envoirment, or because the customer requires functional or performance enhancement. Software support/maintenance reapplies each of the preceding phases to an existing program rather than a new one.

4.2 Architecture of BIP

This architecture chosen is a layered architecture with different functionality divided in different layer. The Figure 4.2 below shows the layered architecture of the application. The detail description of each layer is described below.

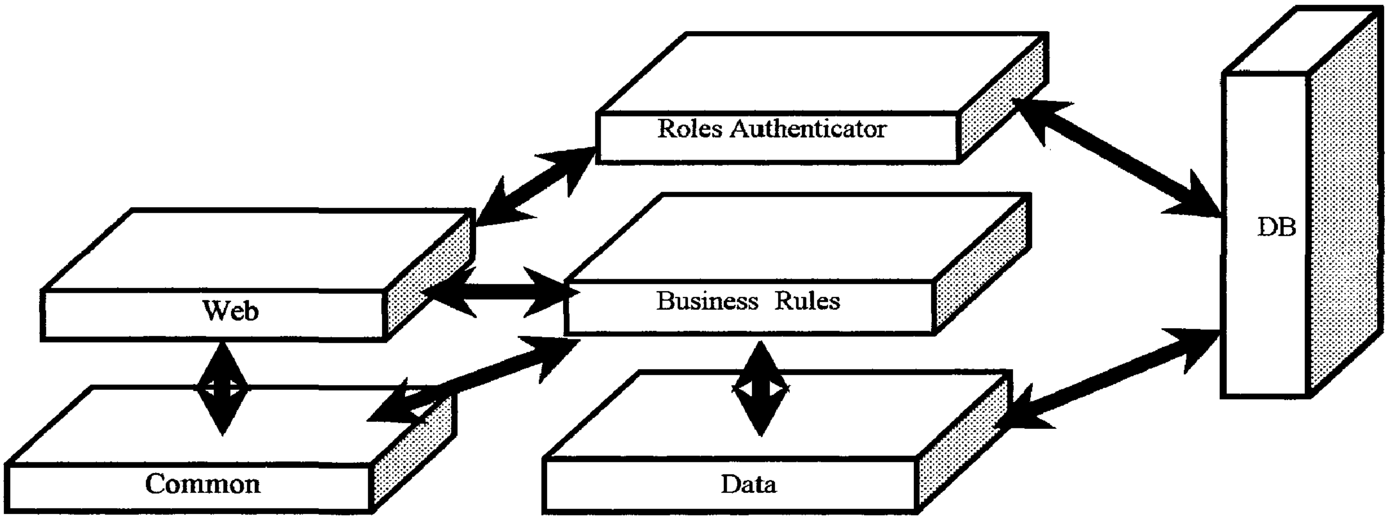


Figure 4.2 Architecture Of BIP.

Web Layer :- This layer communicates with the BusinessRules layer and never accesses the Data or Database layer directly. The Web Layer will have all pages (HTML, ASPX),with the code behind(CS).The function of the Web layer is basically the GUI, and related issues like page_load, page_init, clicks (url and button) and some of the front end validation like mandatory data checks, data attributes (type, length). Web layer depends upon the Common layer for accessing the common methods, on the Business layer for the business logic and the Roles Authenticator for implementing the role based security.

Roles Authenticator Layer :- The RolesAuthenticator layer is for providing the security, which is both role wise and Level wise. The User whole is logging in will be logging in with some role and the visibility of the application will be dependent on that role which can be configured by the administrator. Similarly the component can be allocated with a level no. along with the user so the user will be only able to see the component whose level no. is equal or less than his level.

Business Rules Layer :- The Business Rules layer represents the application's functionality, all the business logic and the validation are done in this layer. The BusinessRules layer is always required to go through the Data layer component to access the DataBase. All kind of database related validation (duplication, parent-child relation, edit rules) is done here before calling the data layer. In data updating cases attributes (type, size) are checked here and then appropriate data layer is called. Business layer depends upon the Common layer and the Data Layer.

Common Layer :- This is the layer which is used by the other layers for accessing the common methods. It is used by the Web layer , the BuisnessRules layer for the common methods.

Data Layer :- BIP uses SQL Server database, ADO.NET (to interface with SQL Server). Thru this component all data related access is handled.

Database :- MRMS uses SQL Server database for storing data.

4.3 Flow Diagram

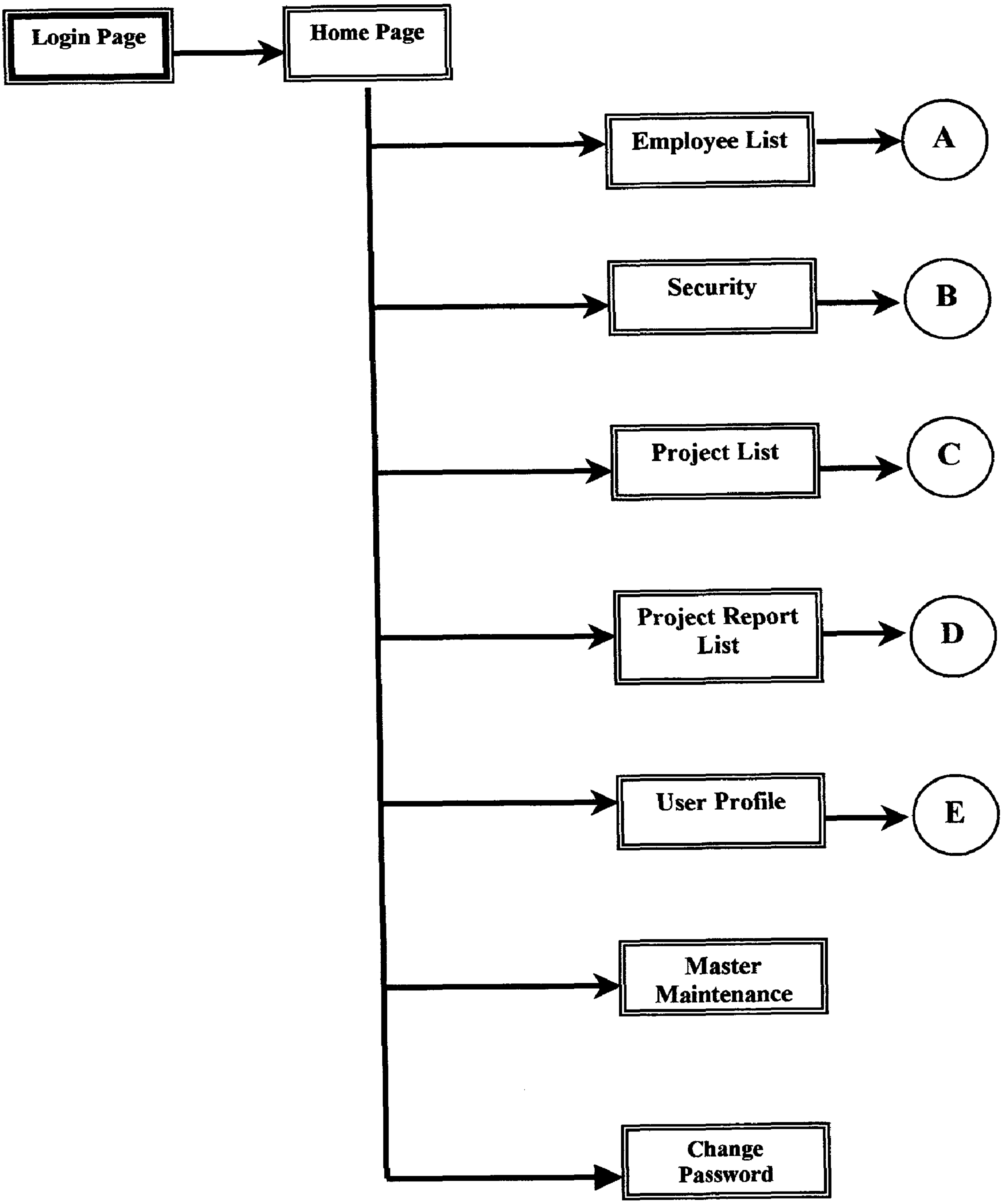


Figure 4.3 Flow Diagram of the Home Page.

Employee List

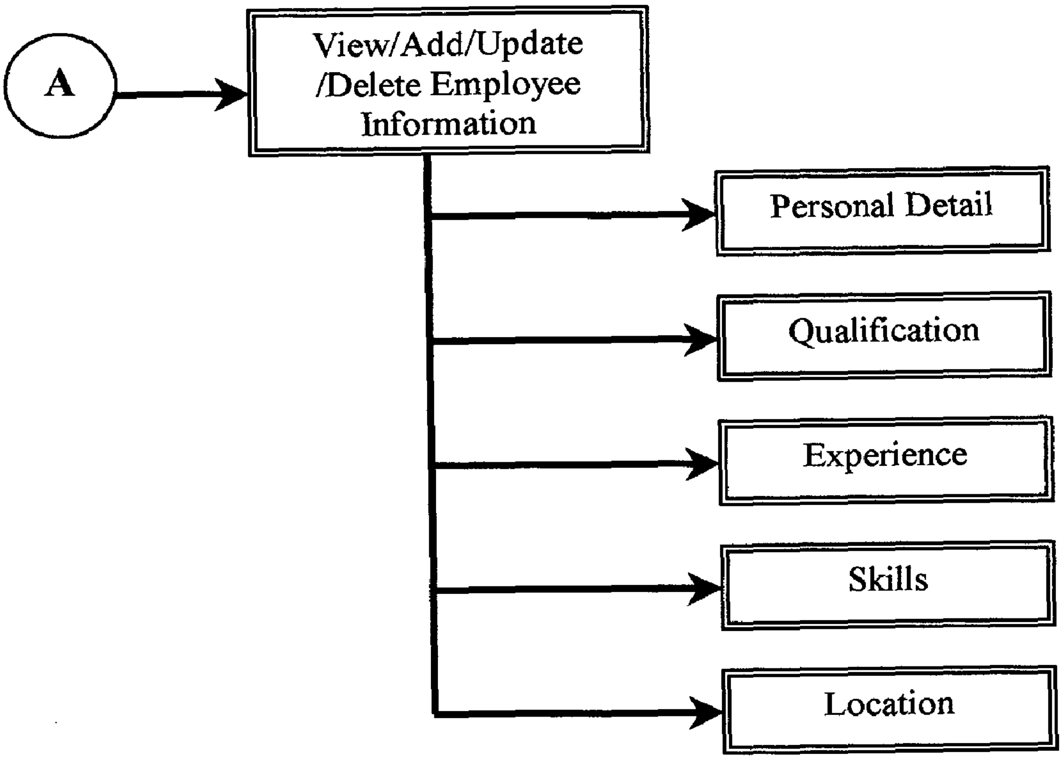


Figure 4.4 Flow Diagram of the Employee Module.

Security

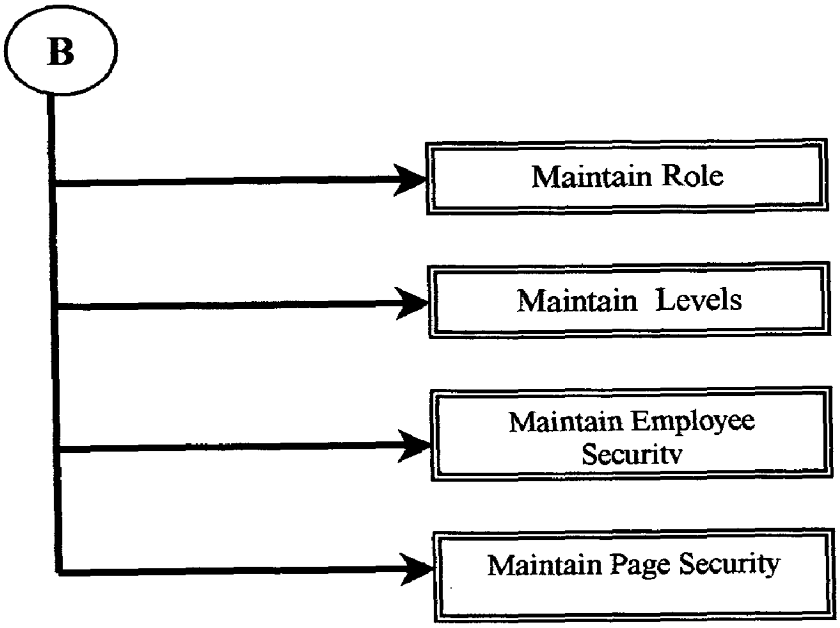


Figure 4.5 Flow Diagram of the Security Module.

Project List

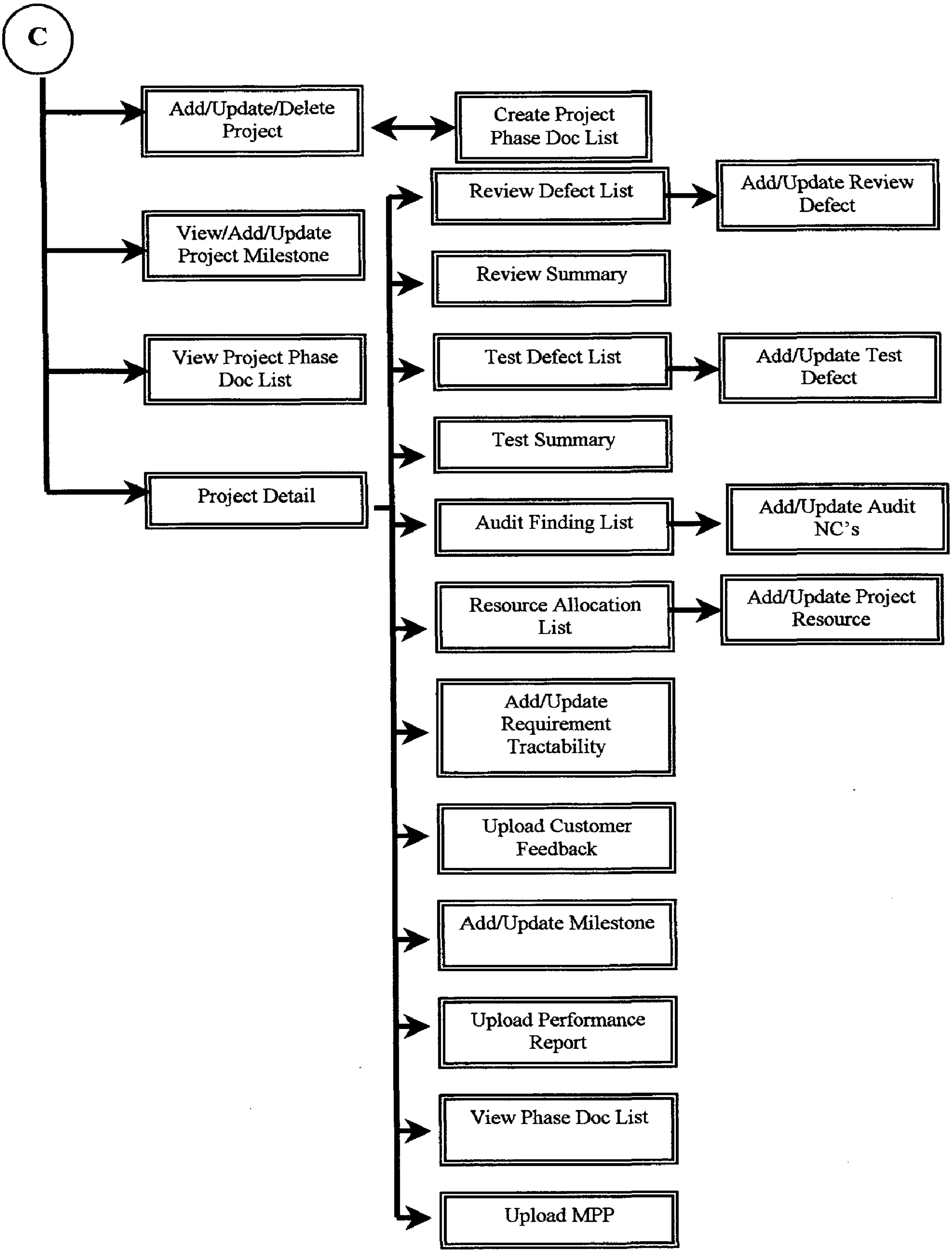


Figure 4.6 Flow Diagram of the Project Module.

Project Report List

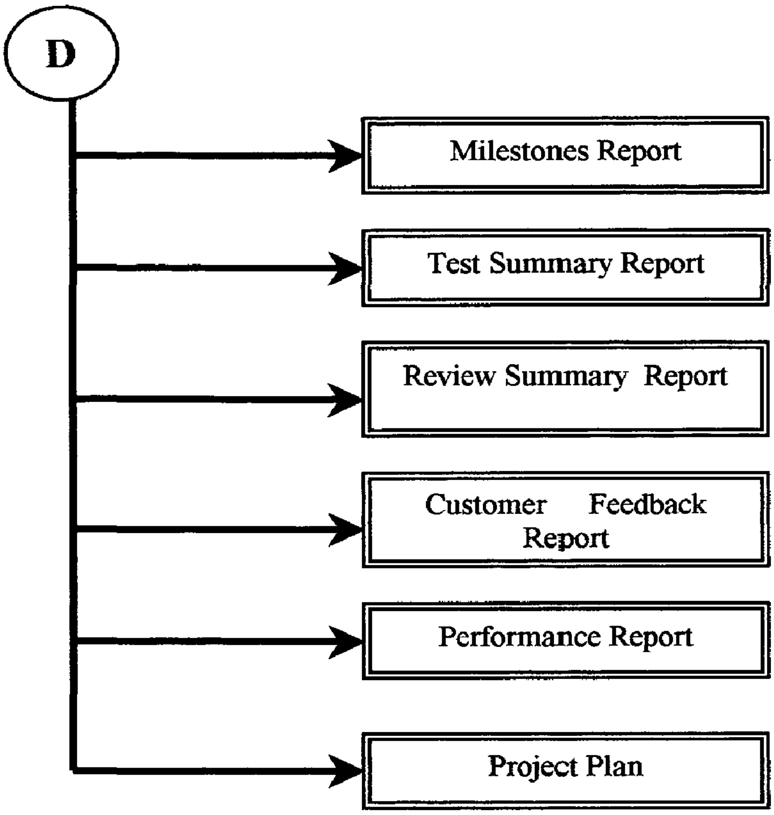


Figure 4.7 Flow Diagram of the Project Report Module.

User Profile

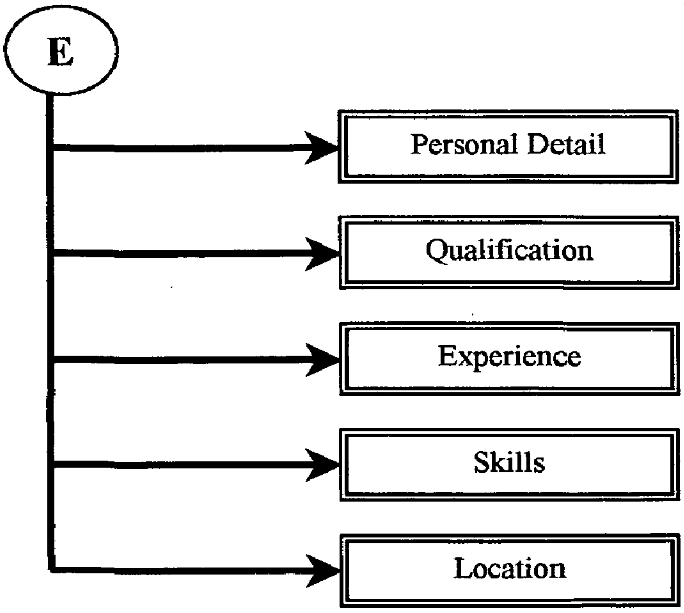


Figure 4.8 Flow Diagram of the User Profile Module.

4.4 Assembly Structure

Dlls	Web	Web.dll
	Business Rules	Business Rules.dll
	Data	Data.dll
	Common	Common.dll
	RolesAuthenticator	Roles Authenticator.dll

4.5 Name Spaces

BIL-Intranet Application.Web

BIL-Intranet Application.BusinessRules

BIL-Intranet Application.Data

BIL-Intranet Application.Common

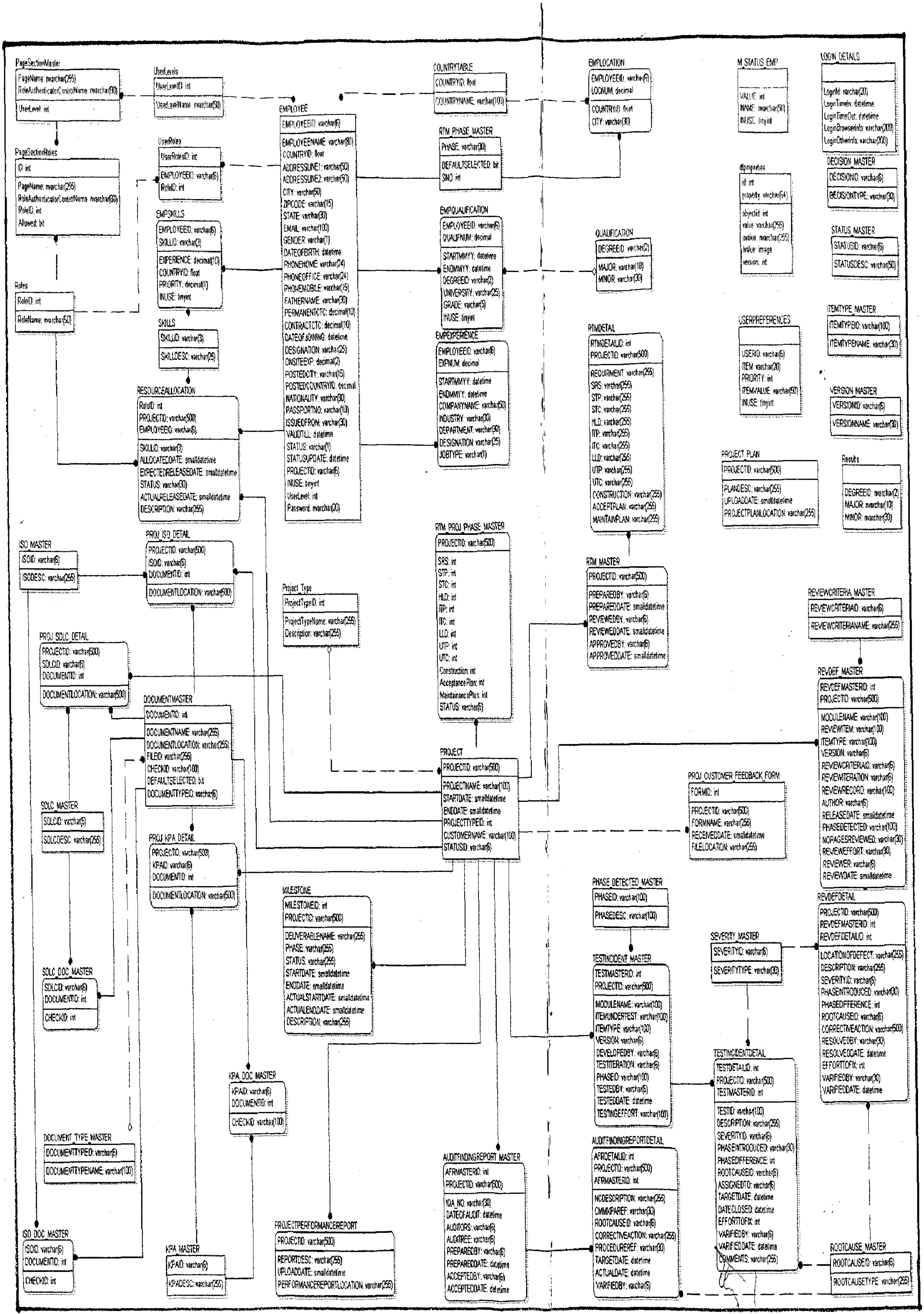
BIL-Intranet Application.RolesAuthenticator

4.6 Classes Structure

Web	The Web Layer contains the .aspx Pages which are the interface of the application , the cod behind .cs file. The naming convention of the files are the name of the page's functionality
BuisnessRules.	The layer have the classes ending with Rule suffix and there are separate classes for separate module.
Data	The layer have the classes ending with Mgr suffix and there are separate classes for separate module.
Common	There is only one class named Common for the common methods.

4.7 Database Diagram

The following is the database diagram of the application with the table structure and the relationship between the tables.


PageSectionMaster
PageName: nvarchar(255)
RoleAuthenticationControlName: nvarchar(100)
UserLevel: int
UserLevels
UserLevelID: int
UserLevelName: nvarchar(50)
PageSectionRoles
ID: int
PageName: nvarchar(255)
RoleAuthenticationControlName: nvarchar(100)
RoleID: int
Allowed: bit
UserRoles
UserRolesID: int
EMPLOYEEID: varchar(6)
RoleID: int
EMPSKILLS
EMPLOYEEID: varchar(6)
SKILLID: varchar(3)
EXPERIENCE: decimal(10)
COUNTRYID: float
PRIORITY: decimal(1)
INUSE: tinyint
Roles
RoleID: int
RoleName: nvarchar(50)
SKILLS
SKILLID: varchar(3)
SKILLDESC: varchar(25)
RESOURCEALLOCATION
RoleID: int
PROJECTID: varchar(500)
EMPLOYEEID: varchar(6)
SKILLID: varchar(3)
ALLOCATEDDATE: smalldatetime
EXPECTEDRELEASEDATE: smalldatetime
STATUS: varchar(30)
ACTUALRELEASEDATE: smalldatetime
DESCRIPTION: varchar(255)
EMPLOYEE
EMPLOYEEID: varchar(6)
EMPLOYEENAME: varchar(80)
COUNTRYID: float
ADDRESSLINE1: varchar(50)
ADDRESSLINE2: varchar(50)
CITY: varchar(50)
ZIPCODE: varchar(15)
STATE: varchar(30)
EMAIL: varchar(100)
GENDER: varchar(1)
DATEOFBIRTH: datetime
PHONEHOME: varchar(24)
PHONEOFFICE: varchar(24)
PHONEMOBILE: varchar(15)
FATHERNAME: varchar(30)
PERMANENTCTC: decimal(10)
CONTRACTCTC: decimal(10)
DATEOFJOINING: datetime
DESIGNATION: varchar(25)
ONSITEEXP: decimal(2)
POSTEDCITY: varchar(15)
POSTEDCOUNTRYID: decimal
NATIONALITY: varchar(30)
PASSPORTNO: varchar(10)
ISSUEDFROM: varchar(30)
VALIDTILL: datetime
STATUS: varchar(1)
STATUSUPDATE: datetime
PROJECTID: varchar(6)
INUSE: tinyint
UserLevel: int
Password: nvarchar(20)
COUNTRYTABLE
COUNTRYID: float
COUNTRYNAME: varchar(100)
RTM_PHASE_MASTER
PHASE: varchar(30)
DEFAULTSELECTED: bit
SNO: int
EMPQUALIFICATION
EMPLOYEEID: varchar(6)
QUALIFNUM: decimal
STARTMMYY: datetime
ENDMMYY: datetime
DEGREEID: varchar(2)
UNIVERSITY: varchar(25)
GRADE: varchar(5)
INUSE: tinyint
EMPEXPERIENCE
EMPLOYEEID: varchar(6)
EXPNUM: decimal
STARTMMYY: datetime
ENDMMYY: datetime
COMPANYNAME: varchar(50)
INDUSTRY: varchar(30)
DEPARTMENT: varchar(30)
DESIGNATION: varchar(25)
JOBTYPE: varchar(1)
EMPLOCATION
EMPLOYEEID: varchar(6)
LOCNUM: decimal
COUNTRYID: float
CITY: varchar(30)
QUALIFICATION
DEGREEID: varchar(2)
MAJOR: varchar(10)
MINOR: varchar(30)
M_STATUS_EMP
VALUE: int
NAME: nvarchar(50)
INUSE: tinyint
dtproperties
id: int
property: varchar(64)
objectid: int
value: varchar(255)
uvalue: nvarchar(255)
lvalue: image
version: int
LOGIN_DETAILS
LoginId: varchar(20)
LoginTimeIn: datetime
LoginTimeOut: datetime
LoginBrowserInfo: varchar(200)
LoginOtherInfo: varchar(200)
DECISION_MASTER
DECISIONID: varchar(6)
DECISIONTYPE: varchar(30)
STATUS_MASTER
STATUSID: varchar(6)
STATUSDESC: varchar(50)
RTMDETAIL
RTMDETAILID: int
PROJECTID: varchar(500)
REQUIRMENT: varchar(255)
SRS: varchar(255)
STP: varchar(255)
STC: varchar(255)
HLD: varchar(255)
ITP: varchar(255)
ITC: varchar(255)
LLD: varchar(255)
UTP: varchar(255)
UTC: varchar(255)
CONSTRUCTION: varchar(255)
ACCEPTPLAN: varchar(255)
MAINTAINPLAN: varchar(255)
USERPREFERENCES
USERID: varchar(6)
ITEM: varchar(20)
PRIORITY: int
ITEMVALUE: varchar(50)
INUSE: tinyint
ITEMTYPE_MASTER
ITEMTYPEID: varchar(100)
ITEMTYPENAME: varchar(30)
VERSION_MASTER
VERSIONID: varchar(6)
VERSIONNAME: varchar(30)
PROJECT_PLAN
PROJECTID: varchar(500)
PLANDESC: varchar(255)
UPLOADDATE: smalldatetime
PROJECTPLANLOCATION: varchar(255)
Results
DEGREEID: nvarchar(2)
MAJOR: nvarchar(10)
MINOR: nvarchar(30)
ISO_MASTER
ISOID: varchar(6)
ISODESC: varchar(255)
PROJ_ISO_DETAIL
PROJECTID: varchar(500)
ISOID: varchar(6)
DOCUMENTID: int
DOCUMENTLOCATION: varchar(500)
Project_Type
ProjectTypeID: int
ProjectTypeName: varchar(255)
Description: varchar(255)
RTM_PROJ_PHASE_MASTER
PROJECTID: varchar(500)
SRS: int
STP: int
STC: int
HLD: int
ITP: int
ITC: int
LLD: int
UTP: int
UTC: int
Construction: int
AcceptancePlan: int
MaintainancePlan: int
STATUS: varchar(6)
RTM_MASTER
PROJECTID: varchar(500)
PREPAREDBY: varchar(6)
PREPAREDDATE: smalldatetime
REVIEWEDBY: varchar(6)
REVIEWEDDATE: smalldatetime
APPROVEDBY: varchar(6)
APPROVEDDATE: smalldatetime
REVIEWCRITERIA_MASTER
REVIEWCRITERIAID: varchar(6)
REVIEWCRITERIANAME: varchar(255)
PROJ_SDLC_DETAIL
PROJECTID: varchar(500)
SDLCID: varchar(6)
DOCUMENTID: int
DOCUMENTLOCATION: varchar(500)
DOCUMENTMASTER
DOCUMENTID: int
DOCUMENTNAME: varchar(255)
DOCUMENTLOCATION: varchar(255)
FILEID: varchar(256)
CHECKID: varchar(100)
DEFAULTSELECTED: bit
DOCUMENTTYPEID: varchar(6)
PROJECT
PROJECTID: varchar(500)
PROJECTNAME: varchar(100)
STARTDATE: smalldatetime
ENDDATE: smalldatetime
PROJECTTYPEID: int
CUSTOMERNAME: varchar(100)
STATUSID: varchar(6)
REVDEF_MASTER
REVDEFMASTERID: int
PROJECTID: varchar(500)
MODULENAME: varchar(100)
REVIEWITEM: varchar(100)
ITEMTYPE: varchar(100)
VERSION: varchar(6)
REVIEWCRITERIAID: varchar(6)
REVIEWITERATION: varchar(6)
REVIEWRECORD: varchar(100)
AUTHOR: varchar(6)
RELEASEDATE: smalldatetime
PHASEDETECTED: varchar(100)
NOPAGESREVIEWED: varchar(30)
REVIEWEFFORT: varchar(30)
REVIEWER: varchar(6)
REVIEWDATE: smalldatetime
PROJ_CUSTOMER_FEEDBACK_FORM
FORMID: int
PROJECTID: varchar(500)
FORMNAME: varchar(255)
RECEIVEDDATE: smalldatetime
FILELOCATION: varchar(255)
SDLC_MASTER
SDLCID: varchar(6)
SDLCDESC: varchar(255)
PROJ_KPA_DETAIL
PROJECTID: varchar(500)
KPAID: varchar(6)
DOCUMENTID: int
DOCUMENTLOCATION: varchar(500)
MILESTONE
MILESTONEID: int
PROJECTID: varchar(500)
DELIVERABLENAME: varchar(255)
PHASE: varchar(255)
STATUS: varchar(255)
STARTDATE: smalldatetime
ENDDATE: smalldatetime
ACTUALSTARTDATE: smalldatetime
ACTUALENDDATE: smalldatetime
DESCRIPTION: varchar(255)
PHASE_DETECTED_MASTER
PHASEID: varchar(100)
PHASEDESC: varchar(100)
TESTINCIDENT_MASTER
TESTMASTERID: int
PROJECTID: varchar(500)
MODULENAME: varchar(100)
ITEMUNDERTEST: varchar(100)
ITEMTYPE: varchar(100)
VERSION: varchar(6)
DEVELOPEDBY: varchar(6)
TESTITERATION: varchar(6)
PHASEID: varchar(100)
TESTEDBY: varchar(6)
TESTEDDATE: datetime
TESTINGEFFORT: varchar(100)
SEVERITY_MASTER
SEVERITYID: varchar(6)
SEVERITYTYPE: varchar(30)
REVDEFDETAIL
PROJECTID: varchar(500)
REVDEFMASTERID: int
REVDEFDETAILID: int
LOCATIONOFDEFECT: varchar(255)
DESCRIPTION: varchar(255)
SEVERITYID: varchar(6)
PHASEINTRODUCED: varchar(30)
PHASEDIFFERENCE: int
ROOTCAUSEID: varchar(6)
CORRECTIVEACTION: varchar(500)
RESOLVEDBY: varchar(30)
RESOLVEDDATE: datetime
EFFORTTOFIX: int
VARIFIEDBY: varchar(30)
VARIFIEDDATE: datetime
SDLC_DOC_MASTER
SDLCID: varchar(6)
DOCUMENTID: int
CHECKID: int
KPA_DOC_MASTER
KPAID: varchar(6)
DOCUMENTID: int
CHECKID: varchar(100)
TESTINCIDENTDETAIL
TESTDETAILID: int
PROJECTID: varchar(500)
TESTMASTERID: int
TESTID: varchar(100)
DESCRIPTION: varchar(255)
SEVERITYID: varchar(6)
PHASEINTRODUCED: varchar(30)
PHASEDIFFERENCE: int
ROOTCAUSEID: varchar(6)
ASSIGNEDTO: varchar(6)
TARGETDATE: datetime
DATECLOSED: datetime
EFFORTTOFIX: int
VARIFIEDBY: varchar(6)
VARIFIEDDATE: datetime
COMMENTS: varchar(255)
DOCUMENT_TYPE_MASTER
DOCUMENTTYPEID: varchar(6)
DOCUMENTTYPENAME: varchar(100)
AUDITFINDINGREPORT_MASTER
AFRMASTERID: int
PROJECTID: varchar(500)
IQA_NO: varchar(30)
DATEOFAUDIT: datetime
AUDITORS: varchar(6)
AUDITEE: varchar(6)
PREPAREDBY: varchar(6)
PREPAREDDATE: datetime
ACCEPTEDBY: varchar(6)
ACCEPTEDDATE: datetime
AUDITFINDINGREPORTDETAIL
AFRDETAILID: int
PROJECTID: varchar(500)
AFRMASTERID: int
NCDESCRIPTION: varchar(255)
CMMKPAREF: varchar(30)
ROOTCAUSEID: varchar(6)
CORRECTIVEACTION: varchar(255)
PROCEDUREREF: varchar(30)
TARGETDATE: datetime
ACTUALDATE: datetime
VARFIEDBY: varchar(6)
ISO_DOC_MASTER
ISOID: varchar(6)
DOCUMENTID: int
CHECKID: int
KPA_MASTER
KPAID: varchar(6)
KPADESC: varchar(255)
PROJECTPERFORMANCEREPORT
PROJECTID: varchar(500)
REPORTDESC: varchar(255)
UPLOADDATE: smalldatetime
PERFORMANCEREPORTLOCATION: varchar(255)
ROOTCAUSE_MASTER
ROOTCAUSEID: varchar(6)
ROOTCAUSETYPE: varchar(255)

4.8 Guidelines and Naming conventions

Variables Names

- Use descriptive names instead.
- Do not use BASIC data type suffixes (%, $, &,!, and so on).
- Prefix names with data types
- No special notation should prefix variables passed by reference or value.

Method Names

- Use mixed-case formatting for method names: e.g. ListEmployee.

Class/Module Description

Each class should be preceded by a class description of the following format.

```
'-------------------------------------------------------
'    Namespace: Web
'    Class: login
'    Description:
'    Special considerations:
'    This class requires read/write session state support.
'-------------------------------------------------------
```

Procedure/Method Description

Each procedure (Sub, Function, Method, or Property) should include the following description. If the fields are blank, then tags can be removed (e.g. a Sub does not need a return code).

```
'**********************************************************
'    Function -Name Of Function-
'---Description of the function---
'    Returns:
'---Description of return value---
'    Parameters:
'    [in]  Flags: Attributes
'    [out] SomeStuff: Returns valid stuff
'**********************************************************
```

The parameters are listed in order and tagged with [in] [in,out] [out] as appropriate. The data returned in the output parameters should be shown here.

In-Function Comments

In-function comments should be one of three types:

- Comments on variable declarations are appropriate on the same line as the declaration.
- Comments describing a large block of code.
- Individual lines or groups of lines.

TESTING

The most major phase in making a quality software is the testing phase. And today high focus is given on the software testing in the development organization. Software testing has become the most critical element in Software Quality Assurance. Usually in a project there are following type of testing done.

Unit Testing
Integration Testing
System Testing
Acceptance Testing

For all the above testing there are test cases on the basis of which the testing is done. The Unit testing, Integration Testing, and the System Testing is done by the developer's side and the Acceptance testing is done by the customer's side.
Below Table 5.1 contains the list of the Test Cases which has to be satisfied before the application is installed.

Table 5.1 List of the Test Cases.

Test ID	Input Specification	Expected Output
1.	On clicking Login Button on the Login Page.	User will be able to login the application using his/her user Sl. No.ID and password. Which is granted to him by the system administrator.
2.	On clicking the "Master" link	The User will shown the Master Information page where he will be adding the master information.
3.	On clicking the "Save" button in the Master Module.	The Master Information will be saved in the database.
4.	On clicking the "Delete" link in the Master List in the Master module.	The Master Information will be deleted from the Dateable.
5.	On Clicking the "Security" link.	The user will be taken to the Security Page where the security for the application can be implemented.
6.	On clicking the "Maintain Roles" on the Security Page.	The Maintain Roles page is opened where the user can enter new roles and update the old ones.

Test ID	Input Specification	Expected Output
7.	On clicking the "Maintain Security Levels" on the Security Page.	The Maintain Security Levels page is opened where the user can enter new levels and update the old ones.
8.	On clicking the "Maintain Employee Security " on the Security Page.	The Maintain Employee Security page is opened where the user can assign level and role to the employee.
9.	On clicking the "Maintain Page Security " on the Security Page.	The Maintain Page Security page is opened where the user can assign level and role to the page to be accessed.
10.	On Clicking the "Employee " on the main page.	The Employee List page is shown where the user shall be able to see the list of all the employees.
11.	On Clicking the "Add" link on the "Employee List" page.	The Employee Detail page is opened where the user will be able to add the employee information.
12.	On clicking the "Report" link on the Home page.	The Project Report page is opened where the user can see the respected reports of the project.
13.	On clicking the "Milestone" link on the Project Report page.	The Milestone Report of the respected project is shown.
14.	On clicking the "Test Summary" link on the Project Report page.	The Test Summary Report of the respected project is shown.
15.	On clicking the "Review Summary" link on the Project Report page.	The Review Summary Report of the respected project is shown.
16.	On clicking the "Customer Feedback" link on the Project Report page.	The Customer Feedback Forms page is shown.
17.	On clicking the "Performance Report" link on the Project Report page.	The Performance Report page is opened of the respected project.
18.	On clicking the "Project Plan" link on the Project Report page.	The Project Plan of the Project is shown.
19.	On Clicking the link "Project"	The User will see the list of all the Project.
20.	On clicking the "Add Project" link	The permitted user will be able to add the new project.
21.	On clicking the "Update "link in the project list	The permitted user will be able to update the existing project
22.	On clicking the "Delete" link in the project list	The permitted user will be able to delete the project

Test ID	Input Specification	Expected Output
23.	On clicking on the "Project" name link in the project list	The project detail page displayed.
24.	On clicking the "Phase "link in the project List Page	The user will be able to see the list of the document allocated to the project SDLC wise, KPA wise and ISO wise.
25.	On clicking the "Milestone "link on the Project Link	The user will be able to View/Add/Update/Delete the Milestone of the Project.
26.	On clicking the "Resource Allocation" link on the Project Detail page.	The user will be shown the Resource Allocation page where the user will be able to allocate and deallocate the employees to the project.
27.	On clicking the "Requirement Tracibility Matrices" link Project Detail page.	The Requirement Tracking page on the project will be shown where the requirement of the project can be tracked.
28.	On clicking the "Review Defect Entry" link on the Project Detail page.	The user will be able to add the Review defect of the project .
29.	On clicking the "Review Summary" link on the Project Detail page.	The user will be able to view the Review defect of the project .
30.	On clicking the "Test Defect Entry" link on the Project Detail page.	The user will be able to add the Test defect of the project.
31.	On clicking the "Test Defect Summary" link on the Project Detail page.	The user will be able to view the Test defect of the project.
32.	On clicking the "Audit Finding Report" link on the project detail page	The permitted user will be able to enter the audit report for the project.
33.	On clicking the "Customer Feedback" link on the project detail page.	The user will be able to upload the Customer Feedback Forms for the project.
34.	On clicking the Performance Report link on the Project Detail page	The user will be able to upload the Performance Report of the Project.
35.	On clicking the "Phase" link on the project detail page.	The user will be able to view the phase document list chosen for the project.
36.	On clicking the "Project Plan" link on the project detail page	The user will be able to upload the Project Plan of the respected project.

RESULTS AND DISCUSSION

With the help of this application we have tried to automate the documentation process involved in an organization striving to achieve CMM levels (3 and above). As we know the documents are the hard proofs required at the time of assessment so now we have an automated process i.e. an application which will help us in achieving the CMM level 3 and above.

Below is the list of the KPA's of CMM till level 4 and list of the corresponding project module i.e. application module through which the application have mapped the functionality of the KPA.

Table 6.1 List of the Modules Mapping the KPA

Sl. No.	KPA	PROJET MODULE
1.	Requirement Management	i. This KPA is being satisfied by the Phase Document List module where the PM while creating a new project can select the document which will be able the track down the Customer Requirement like the Contract, Proposal , SRS or the Estimation Document.
2.	Software Project Planning	i. This KPA is being satisfied by the Resource Allocation module. Which allocates the employee to the project and deallocates them , this module looks after the team management. ii. The KPA is being satisfied by the Phase Document List module by the project kick off related document, and also by the Project Plan document in the same module. iii. The KPA is also satisfied by the Microsoft Project Plan module where the MPP of the specific project can be uploaded.

Sl. No.	KPA	PROJET MODULE
3.	Software Project Tracking and Oversight	i. This KPA is satisfied by continuously tracking and updating the MPP in the MPP Module , also it can be done in the Project Plan in the Phase Document List. ii. This KPA is also satisfied by the Phase document List module which help in maintaining the document of the project of different phases. iii. The Milestones module also help in achieving this KPA. iv. The Project Report Module and RTM module also as helping to satisfy the KPA.
4.	Software Subcontract Management	i. This KPA is needed by the companies which outsource their project to the other companies, and as BIL does not outsource their project so this KPA is not needed at all.
5.	Software Quality Assurance	i. This KPA is satisfied by the planning of the Audit in the Audit Module. Also the Test Defect and Review Defect Module i.e. both the Defect entry and summary module helps in achieving this KPA. ii. The PP(Project Plan) in the Phase Document List Module and the MPP module also helps in it.
6.	Software Configuration Management	i. This KPA is satisfied by the planning of the Audit in the Audit Module. Also the Test Defect and Review Defect Module i.e. both the Defect entry and summary module helps in achieving this KPA. ii. The PP(Project Plan) in the Phase Document List Module and the MPP module also helps in it.
7.	Organization Process Focus	i. This KPA is satisfied by the Phase document list where the overall document list is listed which are tailored according to need, the document is list in all the three form i.e. KPA wise , ISO wise and SDLC wise.

Sl. No.	KPA	PROJET MODULE
8.	Organization Process Definition	i. In the Phase Document list module after the phase documents are chosen by the Project Manager for a new project a auto generated mail is sent for approval by the technical head, after the technical head approval a mail is sends to the SEPG head for the final approval and after that the list is finally accepted to work upon.
9.	Training Program	i. This KPA is satisfied in including the training schedule in the PP and MPP in the Phase document module and the MPP Module.
10.	Integrated Software Management	i. In the Phase Document list module after the phase documents are chosen by the Project Manager for a new project a auto generated mail is sent for approval by the technical head, after the technical head approval a mail is sends to the SEPG head for the final approval and after that the list is finally accepted to work upon.
11.	Software Product Engineering	i. The Phase Document Module helps in satisfying this KPA as in this module the SDLC is chosen for the project.
12.	Inter-group Coordination	i. This KPA is satisfied by the PP and MPP where the inter group activities are planned. ii. The resource allocation module helps in satisfying this module , The employee module where the HR department maintain the employee information also helps in allocation of the resources to the project. iii. The RTM module also helps in this KPA.
13.	Peer Reviews	i. The KPA is satisfied by the Review Defect Entry Module, The Review Summary , Test Defect Entry and the Test Defect Summary module.

Sl. No..	KPA	PROJET MODULE
14.	Software Quality Management	i. All the report of the application like the Review Summary, test summary, Customer Feedback form, the audit finding report, the Performance Report help this KPA.
15.	Qualitative Process Management	i. All the report of the application like the Review Summary, test summary, Customer Feedback form, the audit finding report, the Performance Report help this KPA.

Seeing the above chart we can see that the application has fulfilled the KPA's till level 4 , the other three KPA's can also be satisfied by making some enhancement in the application.

Chapter 7

CONCLUSION

Through the study of the CMM in detail and the functionality of the various KPA's we have come up with the application which will help us in achieving the CMM level 3 and above levels. The application will serve as a central repository for the projects document which will be required at the time of the assessment in the CMM assessment process. The application will also be helpful to the organization's management in collecting the data for analyzing the project performance and improve upon the process. The application will ease the employees who will be accessing the application to not wonder here and there in the system's directory to find the document they want for the Project they are working on. This application is not organization specific it can be used by any software organization which wishes to attain CMM levels, or want to streamline their process.

Future Enhancement

The proposed future enhancement of this application are as follows : Sl. No.

- The attendance recording software can be added in this application which will be useful by the HR department.
- Daily Time sheet filling can be done through this application.
- There can be a Project wise chat room where all the employees who are in the same project can communicate with each other for the project activities.
- The module which will be required for achieving the level 5 KPA's can be added to this application.

REFERENCES

1 The Capability Maturity Model: A Tutorial April 1993 Software Engineering Institute Carnegie Mellon University Pittsburgh, PA Page No. 18

2 D.H. Kitson and S. Masters, An Analysis of SEI Software Process Assessment Results: 1987-1991, Software Engineering Institute, CMU/SEI-92-TR-24, July 1992.

3 M.C. Paulk, B. Curtis, M.B. Chrissis, and Charles V. Weber, Capability Maturity Model for Software, Version 1.1, Software Engineering Institute, CMU/SEI-93-TR-24,February 1993.

4 P. Fowler and S. Rifkin, Software Engineering Process Group Guide, Software Engineering Institute, CMU/SEI-90-TR-24, ADA235784, September, 1990.

5 P.H. Feiler and W.S. Humphrey, Software Process Development and Enactment: Concepts and Definitions, CMU/SEI-92-TR-4, ADA258465, March 1992.

6 M.E. Fagan, "Advances in Software Inspections," IEEE Transactions on Software Engineering, Vol. 12, No. 7, July, 1986, pp. 744-751.

7 D.P. Freedman and G.M. Weinberg, Handbook of Walkthroughs, Inspections, and Technical Reviews, Third Edition, Dorset House, New York, NY, 1990.

8 ISO in Simple English by PRAXIOM RESEARCH GROUP LIMITED HTTP://WWW.PRAXIOM.COM/INDEX.HTM

9 TickIT official web-site http://www.tickit.org/

10 MARK C. PAULK How ISO 9001 compares with the CMM http://www.sei.cmu.edu/cmm/papers/9001-cmm.pdf

11 S Mater .CMM Based Appraisal for Internal Process Improvement(CBA-IPI): Method Description. Technical Report, Software Engineering Institute, CMU/SEI-96-TR-007,1996.

APPENDIX A

CMM Assessment

We are about to give a brief description of the assessment method, the guidelines for the assessment, and detailed instruction for conducting an assessment appears in the SEI's handbook for assessors.

There are two type of assessment

1. CMM based appraisal for internal process improvement (CBA-IPI). It is intended to help the organization being assessed to improve its processes, in other words it evaluates the strength and weaknesses of the software process in addition to evaluating the satisfaction of the different goals of the different KPA's.
2. Software Capability Evaluation, takes place at the request of someone (typically, a customer) outside the organization being evaluated.

Assessment Team: - The team is lead by the SEI authorized lead assessor, and consist of 6 to 10 experienced people from the organization under scrutiny. The team members should be conversant with the CMM and should get assessment training from the lead assessor. During the assessment the team members collects the information about the software process of the organization. There are three main source of information.

1. Maturity Questionnaires
2. Documentation
3. Interviews

1) Maturity Questionnaires:- is the instrument, which is used to get some feedback regarding the process being used in the organization. It contains a set of questions for each KPA. The questionnaire is given to project leader, their supervisors (middle manager), and some project team members. If the answer to the question is overwhelmingly 'yes' the statement in the question could be treated as an observation about that practice Maturity Questionnaire is used as an aid for further exploration during interviews and document examination. Usually four to six projects are selected for the assessment but these project should be representative of the project profile of the organization.

2) Documentation Document from all the project are made available to the assessment team, and their project leader are interviewed individually to obtain more observation and seek ant necessary clarifications. Beside these document of the selected project the document describing the whole process are examined for the purpose of making observation.

3) Interview A group of personals from different functions are interviewed and these group are called Functional Area Representative (FAR), usually 4 to 6 FAR groups are interviewed and each group contains 4 to 10 people.

Some of the possible FAR groups are

1. Project Leader(of the project other than the selected ones)
2. Middle manager to whom the middle manager report
3. Configuration controller
4. Software Engineering Process Group member
5. Training personnel
6. Developers
7. Testers
8. Analysts

The interview attempt to obtain evidence regarding the usage of the key practices of the different KPA's. Generally the maturity questionnaire and the examination of the project document and the document regarding the process and policies will yield some evidence for most key practices. Interviews therefore becomes the mean to clarify the doubts and as additional source of information. Between the information gathering session, the team consolidates whatever information it has gathered and reaches agreement on which key practices can be considered as satisfied, Agreement requires consensus of the entire team. Figure below shows the overall assessment process

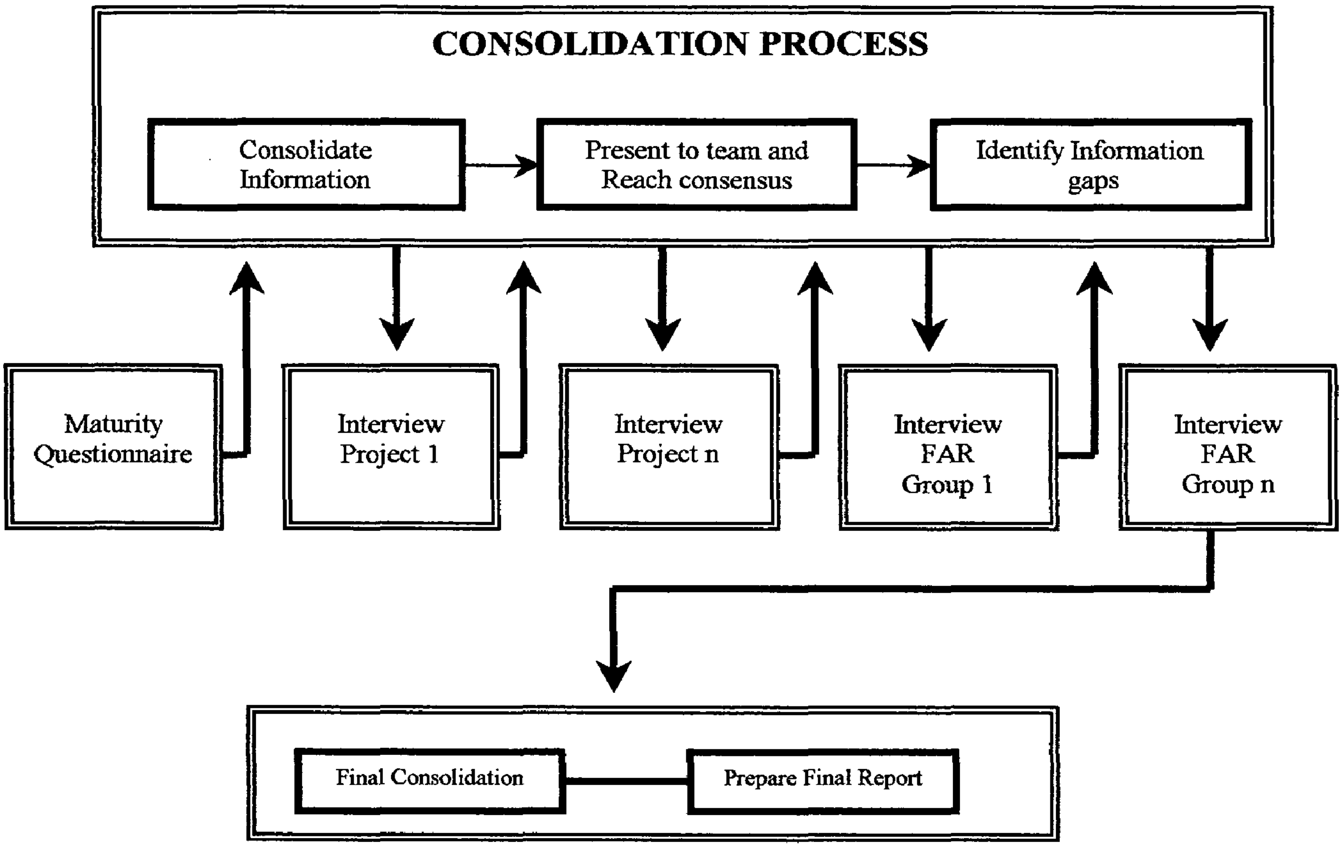


Figure A.1 The Assessment Process

Lets look in the consolidation process more closely, the assessment teams might be organized into multiple mini team, with each mini team responsible for a KPA presents results to the entire assessment team. As noted earlier, a key practice is considered as "Satisfied" only when all team members agree. If doubt persist, then these "gaps" are identified and clarified during the interview session through appropriate questions and requests for documents. If all the key practices are satisfied, then the goals for those KPA's are satisfied. Otherwise, goals satisfaction requires the team to determine weather an alternative practice exists that satisfied the goal. If all the goals of the KPA are satisfied then the KPA is satisfied. In the presentation of the final findings, the goal satisfaction for all KPAs is presented, along with the strengths and weaknesses information. In this manner the final findings not only report the maturity level of an organization, but also delineate the areas in which improvement is possible.

TickIT

TickIT initiative came about as a result of a report commissioned by the British Department of Trade and Industry (DTI) to review the state of software quality and development in industry. This report showed that there was a reluctance on the part of the software producers to adopt ISO9000 as it was pitched at a high level of generality, the terminology was difficult to interpret for software and the guidance documentation was confusing. As a result of the findings of this report, the British Government decided to appoint the British Computer Society (BCS) to lead an initiative called TickIT. The aim of this initiative was to create a detailed method for organization, procedures and rules for a Software Sector Certification Scheme (SSCS) which would cover the assessment and certification of an organization's software quality management scheme to ISO9000/BS5750. A successful audit by a TickIT-accredited certification body results in the award of a certificate of compliance to ISO 9001:2000, endorsed with a TickIT logo[9]. One may consider TickIT as a British guide to using ISO 9001 and ISO 9000-3[10]

So how does ISO/TickIT compare to CMM. Both these standards have a common concern for quality. While ISO identifies the minimal requirements for a quality system, the CMM underlies the need for a continuous improvement. The ISO members maintain that if you read ISO 9001 in depth then you would realize that it does address the continuous process improvement. e.g. Corrective Action clause in ISO may be considered to address continuous improvement. Both ISO and CMM have been accepted world wide. Some organizations, e.g. NASA, have adopted ISO, while other organizations e.g. Department Of Defense, have opted for CMM Mark C Paulk has given an insightful view of comparison between ISO 9001 and CMM [15].

Given below is a diagram taken from Paulk's comparison of ISO 9001, ISO 9000-3 and CMM. The dark shading indicates practices that are directly addressed by ISO 9001 or ISO 9000-3; the light shading indicates practices that may be addressed depending on an interpretation of ISO 9001; and the unshaded areas indicate practices not addressed by ISO 9001. Key process areas may be, therefore, partially or fully satisfied, satisfied under some interpretations, or not satisfied. The size of the bar indicates the percentage of key practices within the key process area that are addressed in either ISO 9001 or ISO 9000-3

APPENDIX B

Comparison with other Quality Management Standards

ISO

The term ISO 9000 refers to a set of quality management standards. ISO 9000 currently includes three quality standards: ISO 9000:2000, ISO 9001:2000, and ISO 9004:2000. In the past, ISO had three standards: ISO 9001:1994, ISO 9002:1994, and ISO 9003:1994. Now there's only one standard: ISO 9001:2000! ISO 9002 and ISO 9003 have been dropped. ISO 9001:2000 presents requirements, while ISO 9000:2000 and ISO 9004:2000 present guidelines. All of these are process standards (not product standards). ISO first published its quality standards in 1987, revised them in 1994, and then republished an updated version in 2000. These new standards are referred to as the "ISO 9000 2000 Standards". ISO's purpose is to facilitate international trade by providing a single set of standards that people everywhere would recognize and respect. The ISO 9000 2000 Standards apply to all kinds of organizations in all kinds of areas. ISO standards are too generic to be successfully implemented to software industry. A special version of ISO for Software Industry also exists, its called ISO 9000-3 [8]. Many people get confused between ISO 9001 and ISO 9000-3.The following statement best explain the difference between ISO 9001 and ISO 9000-3 "ISO prepared the 9000-3:1997 quality guidelines to help organizations to apply the ISO 9001:1994 requirements to computer software. Use ISO 9000-3 if you develop, supply, install, and maintain computer software. ISO 9000-3:1997 is really an expanded version of the old ISO 9001:1994 standard. ISO has simply copied the old text from ISO 9001 and pasted it into the new version of ISO 9000-3, and then added some new text that refers only to software."[8]. The ISO 9000 standards are being improved/modified continuously. The next ISO standard review will abolish ISO 9000-3 and would make this a part of ISO 9001 - this would reduce the above mentioned confusion.

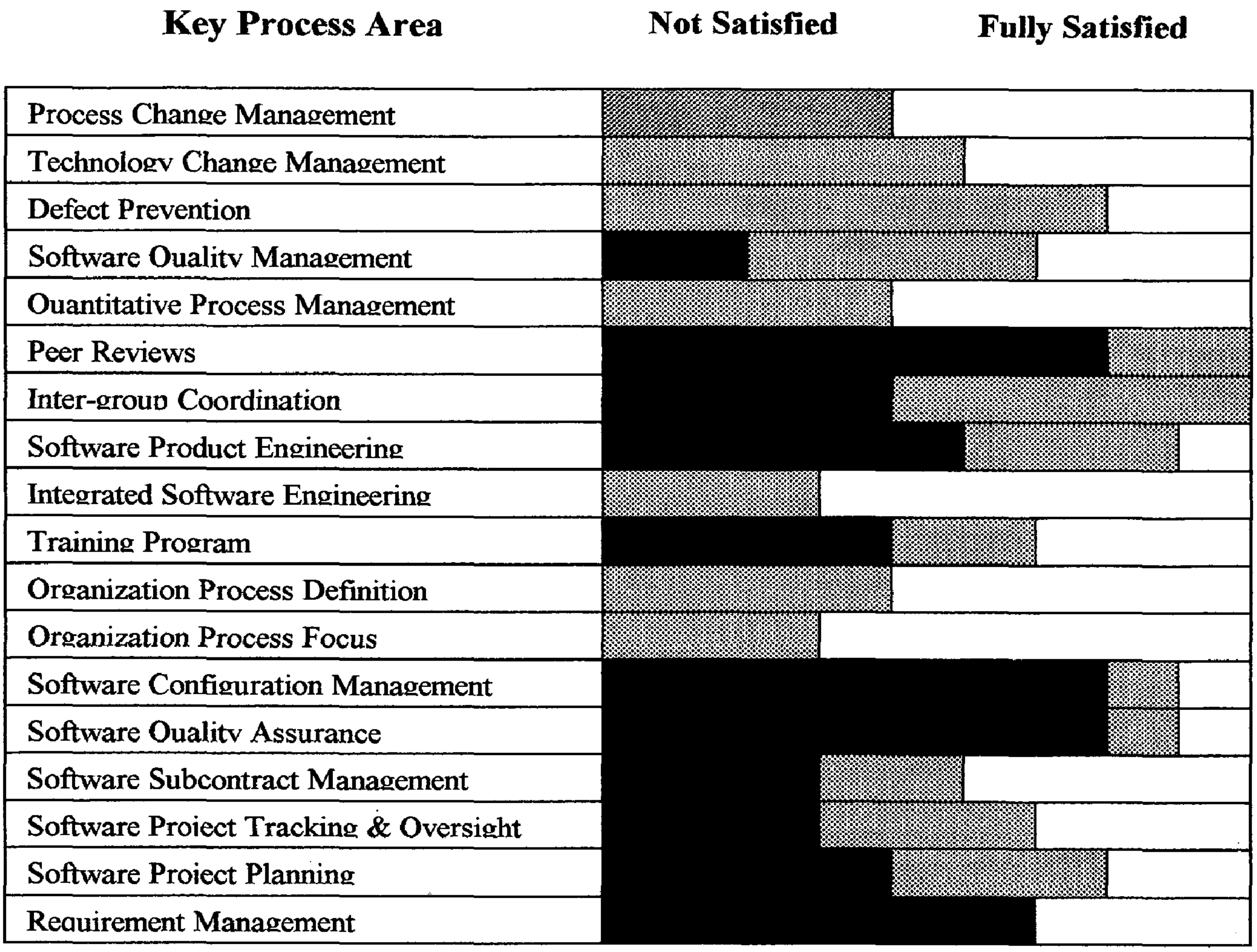


Figure B.1:- Key Process Area Profile of an ISO 9001- compliant Organization