TIM KASSE

PRACTICAL INSIGHT INTO CMMI®

SECOND EDITION

Practical Insight into CMMI®

Second Edition

For a listing of recent related Artech House titles,
turn to the back of this book.

Practical Insight into CMMI®

Second Edition

Tim Kasse

ARTECH
HOUSE

BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the U.S. Library of Congress.

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

ISBN-13: 978-1-59693-275-3

Cover design by Igor Valdman

685 Canton Street
Norwood, MA 02062

10 9 8 7 6 5 4 3 2 1

This book is dedicated to the process improvement efforts of the individuals who worked for and with the Software Engineering Institute (SEI) for the past 20 years. The vision, the hard work, and the willingness to share their ideas with the rest of the world have resulted in a multiple-disciplined CMMI®-DEV that has advanced the state of the art in thinking about process improvement and quality. It has also resulted in the CMMI® Framework Architecture, which allows for the development and use of multiple models. The CMM® for Software that reached the world first is now a constellation of components used together with common materials called the CMMI® Model Foundation or CMF to construct models, training materials, and appraisal materials for areas of interest such as development, acquisition, and services.

I am proud to say that I was given the chance to work with some of those people when I worked for Mr. Watts Humphrey from 1988 to 1992 and I am prouder still of being able to work with, learn from, and support the continuous process improvement efforts of those who are with the SEI today.

Contents

Foreword

I first met Tim Kasse in 2007 during delivery of a course titled Improving Process Performance Using Six Sigma (IPPSS) at the Software Engineering Institute in Pittsburgh, Pennsylvania. Tim's knowledge of the CMMI® was impressive and far exceeded my own knowledge, because my background centered more heavily on Six Sigma and Design-for-Six-Sigma. Yet, similar to Tim, I spent a major portion of my career bringing disciplines together in synergistic ways, beginning first in finance and accounting, and then leveraging those principles in the fields of quality and process improvement. Subsequently, I worked in applied statistics and reliability engineering, and sought to integrate each of these disciplines into practice via pursuit of Six Sigma and CMMI® improvement projects.

With that said, I believe this second edition of Tim's book *Practical Insight into CMMI®: Management and Engineering Principles Behind the Goals and Practices of the CMMI®* represents a great source of valuable insight into understanding aspects of the CMMI®. Tim's latest edition sheds much more light on the activities behind the CMMI®, and parallels new CMMI® high maturity understandings that otherwise might only be available from the latest SEI courses. The contents of the IPPSS course and the successor course, Designing Products and Processes Using Six Sigma (DPPSS), include a host of superior, industry-proven measurement practices that implement a wide variety of practices in the CMMI®, especially high-maturity practices. Tim's immediate appreciation for the practicality and business value of the methods in these courses, as well as the mapping of these methods to the CMMI®, was remarkable. He recognized that implementing these high-maturity methods would, indeed, produce superior results at both the project and organizational levels.

I would like to thank Tim for assisting with the ongoing, cultural change in the community, namely, by publishing this new edition of his book and by continuing to evangelize the rich implementations of the CMMI® high-maturity practices. I would also recommend this latest edition as a must read for those seeking a practical reference on CMMI® and high-maturity understanding.

Robert W. Stoddard II
Certified Motorola Six Sigma Master Black Belt
Senior Member of Technical Staff
Software Engineering Institute,
Carnegie Mellon University
August 2008

Preface

The total quality management (TQM) concepts pioneered by Dr. W. Edwards Deming, Dr. J. M. Juran, and Phil Crosby have been the basis for the CMM® for Software, CMMI® v1.1, and now CMMI®-DEV v1.2. In the explanation of his 14 points for management, Dr. Deming, offered many golden nuggets for us to examine, think about, and try to implement in our own environments. The sentences below are taken from point 3 of Dr. Deming's 14 points:

- Routine 100% inspection to improve quality is equivalent to planning for defects and acknowledgment that the process does not have the capability required for the specifications.
- Inspection to improve quality is too late, ineffective, and costly—when the product leaves the door of the supplier, it is too late to do anything about quality. "You cannot inspect quality into a product."
- Inspections may be carried out at the right moment to ensure minimal total cost.
- Inspection, downgrading, and rework should never be considered corrective action on the process.
- Quality comes from improvement of the process!

The CMMI®-DEV v1.2, which now includes a stronger focus on all engineering disciplines exemplified in the examples, provided for systems, hardware, and software engineering. It clearly brings us back closer to the TQM principles. Rather than pontificate anew on tried and true quality principles, this second edition of *Practical Insight into CMMI®* attempts to present the nuggets found inside of the CMMI® in a pragmatic fashion.

Practical Insight into CMMI®

The second edition of *Practical Insight into CMMI®* provides an understanding of the engineering, project management, process management, and quality management principles behind the specific and generic goals and practices of the CMMI®. It does so without listing the specific goals and specific practices in their technical entirety. It offers clear interpretations that are based on my experience with the definition and the use of the CMMI®. It borrows examples from the Kasse Initiatives systems/hardware/software engineering workshops in order to provide practical insight into those principles behind the CMMI®. It integrates years of experience with the concepts presented in the CMMI®.

This book presents complicated engineering concepts in a manner that will enable both higher level managers and practitioners to visualize what it would be like to work in an organization that fully understood and embraced the best concepts of the CMMI® Reference Model. Its detailed explanation of the concurrency of the engineering concepts as an organization moves from requirements elicitation, analysis, early phase validation, and architecture toward alternative solutions, implementation, integration, testing, and even production will enable higher level managers and practitioners alike to understand what it would be like to work in an organization that understood and embraced the concepts of the CMMI®. In addition, this book explains how to use process areas in more than one context or category, so that the power that can be tapped through its guidance for systems, hardware, and software engineering development can be effectively utilized.

Benefits

This book is designed for and will benefit:

- DoD organizations and contracting firms that are being pushed to accept and comply with the CMMI® v1.2 around the world (United States, Europe, India, Israel, and so on);
- Organizations that have been engaged in SWCMM-based process improvement and have finally decided to embrace the CMMI v1.2 now that the SEI has sunset the CMM® for Software;
- Organizations that must evaluate their suppliers' process capability and corresponding performance before making a contract award or for progress checks during the supplier's project life cycle;
- Organizations that are just beginning to develop a process improvement initiative and want to utilize the full features that the continuous and staged representation have to offer to help them build and implement a measurable and incremental approach;
- Systems engineering–oriented firms who seek the capability to manage multidisciplined projects and need the full guidance that CMMI® offers;
- Hardware or manufacturing-oriented companies that develop software and have stopped short of embracing the CMM® for Software or the Systems Engineering CMM®;
- Quality management consulting firms around the world that have not yet accepted the Capability Maturity Model from the SEI or its assessment method, the CBA IPI, but who are not happy with SPICE or ISO 9001:2000 or related standards;
- Universities that offer a systems engineering curriculum;
- INCOSE, the systems engineering organization that organizes systems engineering conferences worldwide that are equivalent to the Software Engineer-

ing Process Group conferences that take place each year in the United States, Europe, and Asia;
- Companies that appreciate and want the discipline that the CMMI® practices offer;
- Individuals, groups, and companies that want a practical understanding of the CMMI® together with examples from industry before they have to plow through the technical details of the CMMI® technical report or SEI-sponsored book;
- Managers at all levels who want and need an understanding of the CMMI® principles and practices along with practical examples, but who do not want the full details of the CMMI® model;
- Those people who purchased the first edition of this book and are now looking for upgraded guidance based on CMMI v1.2;

What is different, interesting, and useful in this second edition compared to the original edition?

- *New chapter on reviews and testing,* The CMMI® calls for peer reviews in Specific Goal 2 of the Verification process area. The examples given for peer reviews are inspections and structured walkthroughs. Based on 30 years of experience using structured walkthroughs and developing an inspection workshop used on three different continents and in the Masters of Technology curriculum at the National University of Singapore, this chapter describes different types of reviews including:
  - Management or progress reviews;
  - Peer reviews including buddy checks, circulation review, technical reviews, structured walkthroughs, and inspections.

  It also includes a brief overview of testing and provides an interpretation for the classic types of testing including:
  - Unit testing, integration testing, systems testing;
  - Acceptance testing;
  - Regression testing (high cohesion and loose coupling);
  - Designing systems that are maintainable and expandable.
- *Quality factors, quality criteria, and quality metrics.* Detailed description of the quality factors or quality requirements called for in the technical solution. The quality requirements, or quality factors as they are often referred to, are not frequently discussed in requirements gatherings nor are they designed in nor verified or validated throughout the project life cycle, yet these nonfunctional requirements are often at the heart of acceptance testing. In this edition of *Practical Insight into CMMI®*, a discussion of the goal–question–metric paradigm will be given along with an overview of quality factors, quality criteria, and quality metrics. Examples will be provided taken from a technical paper written by Tim Kasse on measuring maintainability and expandability.
- *Make middle managers process owners.* The chapter on improving processes at the organizational level will include a discussion of the infrastructure

required for the organization to have a successful process improvement initiative. An argument is made for making middle managers the "process owners" instead of the technical experts. A technical presentation on this topic was recently presented at the CMMI®-2006 Conference in Denver.

- *Additional guidance on architecture.* Architecture is the weakest part of the CMMI® model with only a very short and high-level description of functional and physical architecture given in the CMMI® v1.2. One chapter will offer the reader an enhanced look at architecture and its importance in overall project and product success.
- *In-depth look at change control boards.* The description of configuration management in support of project management will include an in-depth discussion on the definition and authority of multiple levels of change control boards. A comparison of the configuration management system and change management system in the CMMI® versus the IT Infrastructure Library (ITIL) will also be offered.
- *Totally revised section on supplier agreement management.* The need for supplier management continues to rise as major international companies dominate the global market. The scope of supplier management will be expanded based on the author's experience and input provided by clients worldwide. Some of the enhancements the reader can expect to see include:
  - Upgraded diagram on what is a supplier "outside" of the project's boundaries;
  - New section on certification of new suppliers;
  - New section on supplier management with QA to support procurement on initial audits and ongoing audits;
  - New section on supplier management meetings at the supplier's site and the buyer's site to ensure the process is being followed, and to increase confidence that the supplier will be able to keep its commitments and deliver high-quality products and product components;
  - New section presenting the concept of the buyer supporting the supplier to try to help the supplier get its performance up to expectations;
  - New section about buyers using techniques like kaizen events to support their suppliers;
  - New section on preferred suppliers and a supplier's performance evaluation (For instance, does the supplier's rating get changed or does the supplier get dropped from the supplier's list entirely?).
- *Interfaces.* The importance of interfaces will be stressed with a focus on controlling interfaces for better interoperability across multidisciplined projects.
- *Constraints on alternative solutions.* The CMMI® requires that alternative solutions provide a life-cycle balanced solution against cost, schedule, functionality, performance, and risk. The chapter on alternative solutions will provide motivation and explanation to consider an expanded list including:
  - Costs;
  - Schedule;

    - Performance;
    - Quality;
    - Functionality;
    - Risk;
    - Customer satisfaction;
    - Support;
    - Training;
    - Maintainability;
    - Sustainability;
    - Disposal.

- *Causal analysis techniques.* The CMMI® suggests that causal analysis techniques be used to identify special causes of variation to achieve stable processes and predictability and for common causes of variation to continually improve the performance of the organization's set of standard processes. This edition of *Practical Insight into CMMI®* will provide an expanded set of causal analysis techniques with a more detailed explanation of their uses than in the first edition. While not attempting to provide Six Sigma Green Belt training, this enhanced explanation of causal analysis techniques should prove useful for all organizations that are trying to improve their organization's processes.
- *Evolving measurements.* This second edition offers an enhanced description of evolving measurement for maturity levels ML2, ML3, and especially ML4 and ML5. It provides guidance for developing and using process performance baselines and process performance models. It distinguishes between the visualization techniques such as fishbone diagrams, bar charts, scatter diagrams, and affinity diagrams with statistical techniques such as control charts and confidence intervals and prediction intervals. It describes and illustrates the links between the process areas of organizational process performance, quantitative project management, causal analysis and resolution, and organizational innovation and deployment.
- *Applying CMMI® to manufacturing.* Chapter 25 is a special chapter that presents a case study on how the CMMI® was successfully applied to manufacturing to help improve production processes in parallel to development processes and achieve CMMI® ML3. The contributing authors are Sarit Assaraf and Itzhak Lavi from Israel Aerospace Industries, Ltd., in Israel.

You are invited to read and apply the concepts found in this second edition of *Practical Insight into CMMI®*. Some of the ideas will be useful immediately. Some will need to be discussed and adapted to address the concerns and focus of your particular organization with its vision, business objectives, measurement objectives, and culture. Your improvement suggestions are always highly welcomed.

Success always!

Acknowledgments

The author would like to express his sincere thankfulness and gratitude to Pamelia S. Rost who worked exhaustively behind the scenes as a reviewer, graphics designer, researcher, editor, copier, interface to the publisher, and nonstop supporter of this undertaking. She not only wholeheartedly supports Kasse Initiatives in multiple ways, but she unhesitatingly provides her support to me both professionally and personally when it is needed and sometimes when I don't even know that I need it.

I would also like to thank Karyn Tegtmeier, who again contributed many hours toward making the chapters of this book more readable and technically acceptable.

Special thanks are offered to Six Sigma and High Maturity instructors of the Software Engineering Institute for contributions to my understanding of high-maturity concepts.

Special thanks are offered to Bob Stoddard who has helped me to get on the right path toward understanding high maturity with his practical approach based on many years of industrial experience. He never tired of the relentless questions asked of him in my quest for understanding. He helped the SEI to grow in its ability to offer insight into the high-maturity ideas encased in the CMMI® and he helped the rest of us to appreciate the brilliance behind the CMMI® that has been there for all of these years.

I would also like to thank my wife, Jan Kasse, for her support of the long hours, including lost weekends and holidays, that I put in to complete this book. Without your love and support, my efforts would not be as effective as I think they were.

Book Overview

This book provides the reader with an insight into what activities an organization would be engaged in and what the role of each level of management and the practitioners would be if their systems, hardware, and software engineering processes were based on the CMMI®-DEV v1.2. It captures the essence of each of the process areas by presenting them in a practical context without the technical structure of the CMMI® masking the valuable nuggets of information. It provides a view into the new understanding that SEI is presenting regarding high-maturity activities and what makes a high-maturity organization. It also takes numerous examples from Kasse Initiatives' workshops and consulting experience to provide the management and engineering principles behind the goals and practices of the CMMI®

Part I: Introduction

Chapter 1: Engineering Systems Thinking

This chapter starts practical insight off by presenting the merger of systems engineering, hardware engineering, and software engineering process improvement ideas. It provides a brief overview of the systems engineering, hardware engineering, and software engineering sources that were merged to develop the CMM® Integration v1.2 (CMMI®). It provides definition detail for systems engineering. In addition, it stresses or recaptures the focus on "Engineering Systems Think" that seems to have gotten lost over the years where the focus has strictly been on the artistic side of software engineering. This is, in part accomplished by an examination of the principles captured in Peter Senege's *The Fifth Discipline: The Art and Practice of the Learning Organization*. The laws of systems engineering thinking are presented.

Chapter 2: Oriented to Business Results

The software industry has long criticized the CMM® model for its lack of focus on business results. Other total quality management models such as Malcolm Baldrige and the European Foundation for Quality Management place heavy emphasis on business results—not just on documenting practices—to gain industry certification. The CMMI® clearly and repeatedly states the need for all process improvement activities to measurably support the organization's business objectives. CMMI®-DEV v1.2 almost overemphasizes the need to have clearly stated and communicated business objectives for all process areas but especially for MA, OPD+IPPD, IPM+IPPD, OPF, OPD, OT, OPP, QPM, CAR, and OID.

Chapter 3: Process Improvement Based on the CMMI®

The CMMI® was developed to provide a single model to be used by organizations pursuing business unit, regional, and even enterprise-wide process improvement. Not only does it provide needed guidance for integrating systems, hardware, and software development activities, but it supports the coordination of multidisciplined project activities that are or may be required to successfully build a product or product component. This chapter stresses the link between a focus on process and a successful business. It asks and gives one possible answer to the question: "Are process descriptions enough?" It provides provocative guidance to categorizing the process areas in ways not presented by the staged or continuous representations for process improvement purposes. It presents multiple entry strategies into process improvement as opposed to only believing an organization must first have an assessment to get its process improvement initiative going. This chapter also illustrates how CMMI®-based processes can be translated into business processes for other organizational department such as human resources, finance, marketing, computer services, and contract management.

Chapter 4: The Language of the CMMI®

This chapter intends to provide the reader with a better understanding of some of the more critical vocabulary that is used throughout the CMMI®. Over the years, many people have suggested that the CMMI® was written in an artificial language. The CMMI® Language does require effort to understand and apply its practices throughout an organization. Although an organization is certainly not required to strictly adopt CMMI® terms in order to show compliance with its principles and guidance, some knowledge of the most important terms starts the journey of getting the "look and feel" of what it would be like to implement the CMMI® concepts in an organization.

Chapter 5: Roles and Responsibilities

This chapter provides the reader with some ideas of the roles and responsibilities of the various levels of management and practitioners needed to support a CMMI®-based process improvement initiative. Those roles and responsibilities include:

- Senior management;
- Middle management;
- Higher level management;
- Project leaders;
- Practitioners;
- Process group;
- Quality assurance;
- Configuration management;
- Testing;

- Measurement;
- Systems engineers.

Part II: Project Management

Chapter 6: Enabling Project Managers to Manage and Control Better Through Project Planning and Project Monitoring and Control

Chapter 6 is the first of five chapters (Chapters 6 through 10) that focuses on the inclusive topics of project management. It introduces the author's philosophy regarding the contributing components of project management. The initial paragraphs include a short introduction to risk management, quality management including quality assurance and configuration management, supplier management, and integrated project management. The topics of project planning and project monitoring and control are the focus of the "look and feel" description of this chapter following the general project management introduction.

Chapter 7: Enabling Project Managers to Manage and Control Better Through Risk Management

Risk management has been placed in the CMMI® as a separate process area to call attention to its importance in managing successful projects and successful businesses. The basic risk management functions of risk identification, risk analysis to determine probability, impact and time frame, risk prioritization, determining a risk management strategy, developing a risk mitigation plan, determining and evaluating contingency plans, and proactively tracking and managing the risks are all included in this chapter with direct links to the overall umbrella theme of project management.

Chapter 8: Enabling Project Managers to Manage and Control Better Through Quality Management

While the continuous representation of the CMMI® chooses a categorization scheme that placed CM and QA is the category of "support," it is the author's experience that effective use of the engineering principles of CM and QA are best realized by thinking of them as project management/quality management functions. The quality management process areas of Process and Product Quality Assurance and Configuration Management are described in this chapter as project management functions that provide input to a project manager to help them manage and control better and not simply go through the motions to satisfy audit or assessment criteria.

The project's quality plan is emphasized in the section on process and product quality assurance. The section on configuration management describes in sufficient detail the configuration management functions of identification, baselining, change control, status accounting, interface control, supplier control, configuration auditing, the configuration management system, and release.

Chapter 9: Enabling Project Managers to Manage and Control Better Through Supplier Management

Subcontracting or working with suppliers is a common, but still uncomfortable fact of life. Companies that insisted they would not use subcontractors in the past are finding themselves in an everyday business situation that puts them in the position of deciding how to select a qualified subcontractor or supplier. These companies either write their RFPs to try and totally control their supplier's every move or they seek out a supplier that states it is CMMI® Maturity Level 5 in the belief that the supplier will manage itself, requiring no effort from the project that selected the supplier.

This chapter describes the spectrum of types of suppliers outside of the project's boundaries that should be managed. It examines the starting position, when a company decides use a supplier, and walks the reader through each acquisition step from assembling a contract management team, to developing supplier selection criteria, to managing the supplier based on the supplier agreement, until the product or product component is delivered and accepted. It takes the approach that effective supplier management means that a project and/or business unit must have effective requirements for engineering, project management, and quality management processes established and maintained for their own use to be able to properly and effectively apply them to their suppliers.

Chapter 10: Enabling Project Managers to Manage and Control Better Through Integrated Project Management

Integrated project management takes project management to another dimension as it describes the project management function discussed in Chapters 6 through 9 based on the organization's set of standard processes. This chapter also serves as the conclusion to the overall discussion of project management. It emphasizes project management based on the organization's set of standard processes that are used together with the organization's defined tailoring guidelines to create the "project's defined process," which is then used to develop and manage the integrated project plan. It stresses the integration of all of the plans that affect the project because the demands of Maturity Level 3 for project management are higher and deserve close attention.

Part III: Engineering

Chapter 11: The Recursive Nature of Requirements Engineering

Collecting and understanding requirements is the necessary, but not necessarily sufficient, start of a successful project. While certainly not true in all cases, the requirements phase for many projects has been largely restricted to requirements gathering. In probably far too many cases, design and even implementation were started before requirements were known or stabilized to a sufficient point. This chapter presents the "recursive" nature of the total requirements gathering and analysis process from initial identification of stakeholders to deriving requirements to validating requirements at all stages. The CMMI® process areas covered in this chapter will be requirements development and requirements management. Topics covered will include:

- Identifying stakeholders;
- Eliciting requirements;
- Documenting customer requirements;
- Translating customer requirements into product and product components;
- Identifying interface requirements;
- Developing and evolving operational concepts;
- Developing and evolving operational scenarios;
- Deriving requirements;
- Performing functional analyses;
- Discovering additional requirements;
- Analyzing and validating requirements at all stages.

Chapter 12: Solution Definition

This chapter presents the concepts and guidelines that the CMMI® has to offer on establishing criteria and selecting product or product component solutions from alternative solutions. It includes the concepts of decision analysis and resolution for more formal decision making. Critical insight is provided that shows alternative solutions are not only different ways of addressing the same requirements, but they also reflect a different allocation of requirements among the product components comprising the solution set. This chapter also focuses on the design, implementation, and unit testing of the product or product component. Topics covered include:

- Determining the alternative solution selection criteria through analysis and definition of ranges of constraints that are used to select the best possible solution given those constraints;
- Selecting alternative solutions;
- Understanding concurrency and iteration of the practices in requirements development, technical solution, and product integration;
- Developing and trying out alternative architectures following the evolutionary approach;
- Developing alternative designs;
- Discovering additional requirements;
- Making use of decision analysis and resolution for more formal decision making regarding selecting architectures, suppliers, COTS products, and alternative solutions;
- The importance of peer reviews and unit testing on product components in preparation for product integration and corresponding verification and validation.

Chapter 13: Reviews and Testing

The CMMI® calls for peer reviews in Specific Goal 2 of the Verification process area. The examples given for peer reviews are inspections and structured walkthroughs. Based on 30 years of experience using structured walkthroughs and developing an Inspection workshop used on three different continents and in the Masters of Technology curriculum at the National University of Singapore, this chapter describes different types of reviews including:

- Management or progress reviews;
- Peer reviews including buddy checks, circulation review, technical reviews, structured walkthroughs, and inspections.

It also includes a brief overview of testing and provides an interpretation for the classic types of testing including:

- Unit testing, integration testing, systems testing;
- Acceptance testing;
- Regression testing (high cohesion and loose coupling);
- Designing systems that are maintainable and expandable.

Chapter 14: From Components to Products: Gluing the Pieces Together

This chapter presents the CMMI® process areas of Product Integration, Verification, and Validation as a "triple" and shows how their use guides projects from the building blocks developed during technical solution to an integrated, verified, and validated set of product components that are then ready for packaging and delivery. It emphasizes the need to define, manage, and control interfaces from the very start of requirements gathering to interface testing in support of rigorous systems testing.

Part IV: An Organizational Focus

Chapter 15: Improving Processes at the Organizational Level

This chapter describes the organizational components necessary to establish and keep the organization on a path of continuous process improvement. It includes the CMMI® process areas of Organizational Process Focus and Organizational Process Development. The description of the Organizational Process Development process area emphasizes the various components that must be in place before an organization can claim compliance with the requirements and guidance provided by this process area (PA). It will distinguish between a product life cycle and a process description. It will clearly show the importance of establishing and enforcing "tailoring guidelines" for project use of organizational processes and it will show the importance of collecting, advertising, and using good examples for project uniformity and success.

Part V: Individual Knowledge and Skills to Integrated Teams

Chapter 16: The Knowledge and Skills Base

This chapter provides the reader with an underlying understanding of an organizational training program that takes into consideration:

- What business the organization is in;
- What core competencies must be developed or acquired to support that business;
- The knowledge and skills currently available in the organization's workforce;
- The training, mentoring, and coaching needed to develop or enhance the workforce knowledge and skills to accomplish individual, project, and organizational goals.

Chapter 17: Integrated Teams

An integrated team, also known as an integrated product team, is composed of members who are collectively responsible for delivering the work product. Team members include empowered representatives from both the technical disciplines and business functional organizations involved with the product and have a stake in the success of the work products produced. Within defined boundaries, these representatives have decision-making authority and the responsibility to act for their representative organizations. These integrated teams may be viewed as a microversion of the company or business unit itself.

Integrated teaming is built on top of integrated project management and the process assets described in Specific Goal 1 of the Organizational Process Definition. This chapter describes the conditions under which integrated teams are considered, built, and managed. The contents are taken from Organization Process Definition: Specific Goal 2 and Integrated Project Management: Specific Goal 3.

Part VI: Measurement and Analysis to High Maturity

Chapter 18: Establishing the Measurement Foundation

This chapter illustrates the strong measurement focus that can be found and utilized from the CMMI®. It starts with basic project management measures and an understanding of getting a measurement program started through the implementation of the concepts found in the Measurement and Analysis PA. The concept of supporting management's "information needs" is examined in depth. Information needs are linked to examples of project measurement objectives. Readers are taught to ask "why?" when a measure is required at the organizational or project level to ensure that everyone involved with the data collection and analysis understands the true information need. An example is provided to illustrate why a common organizational business objective can and normally should be supported by different project business objectives. Basic and derived measures are discussed.

Basic statistical analysis is introduced in this chapter and discussed in detail in Chapter 19. It is important for all organizations involved in CMMI®-based process improvement to understand that the concepts of basic statistics and some Six Sigma Green Belt concepts are expected to be understood and used at Maturity Level 2.

This chapter also guides the reader to the establishment of an organizational measurement repository at Maturity Level 3. It focuses on the collection of peer review and test data, and the evolution of the organizational process measures that provide the building blocks for statistical process control and quantitative project management.

Chapter 19: Basic Statistics and Probability Distributions

This chapter is intended to bring basic statistics concepts into the forefront of the reader's consciousness with regards to building a measurement and analysis system. It is not intended to teach anyone details about statistics. It is, however, intended to illustrate the importance of a basic statistics vocabulary in order to understand the guidance the CMMI® provides to those organizations who wish to be high maturity. Common statistical terms briefly defined in this chapter include:

- Mean;
- Median;
- Mode,
- Variance;
- Discrete probability distributions;
- Poisson distribution;
- Continuous distribution;
- Normal distribution;
- Exponential distribution;
- Central limit theorem;
- Attribute or discrete data;
- Continuous or variable data;
- Measurement system error;
- Hypothesis testing;
- Confidence intervals;
- Prediction intervals.

Chapter 20: Quantitative Management

When higher degrees of quality and performance are demanded, the organization and projects must determine if they have the ability to improve the necessary processes to satisfy the increased demands. This is an important indicator of higher maturity organizations. This chapter focuses on the concepts surrounding high maturity and the guidance provided by the Organizational Process Performance and Quantitative Project Management PAs. Concepts embodied in the words *statistical management, quantitative management,* and *predictability* are explained. Examples of quantitative quality and process performance objectives that are often established at the project level to support organizational business objectives are provided. Process performance baselines are defined and illustrated. Examples of process performance models and related statistical methods are offered.

Quantitative project management is presented as project oriented to complement Organizational Process Performance (OPP), which is organization oriented. The project's quality and process performance objectives are revisited, but this time from a quantitative point of view. Why it is necessary to statistically manage critical subprocesses throughout the project life cycle is highlighted.

Understanding variation is a fundamental premise of statistical control. Understanding special and common causes of variation are presented based on original concepts developed by Dr. Walther Shewhart and succinctly captured by Dr. Donald Wheeler. Visualization tools and techniques such as fishbone diagrams, scatter diagrams, and Pareto analysis are listed.

Finally, a simple scenario is offered that illustrates how the process areas of Maturity Level 4 and Maturity Level 5 interactively work together to help any organization to achieve a superior position in its respective industry.

Chapter 21: Beyond Stability

Reducing the variation of a process or subprocess by eliminating the special causes of variation and stabilizing the process is only the first step in attempting to achieve the quantitative process performance and/or quality objective. The second and most important step is to find an incremental, innovative, or improvement solution that will actually result in producing a product that meets the expectations or requirements/specifications of the customer. This chapter describes the causal analysis and process improvement innovations that can be built on the quantitative and predictable knowledge of an organization's processes to solve business needs that otherwise could not be solved simply through hard work and management concerns. Basic causal analysis techniques employed for causal analysis and resolution are presented through a brief description of each causal analysis technique and graphical representation.

Incremental and innovative improvements that measurably improve the organization's processes and technology and in turn support the organization's quantitative quality and process performance objectives derived from its business objectives are explained in this chapter's coverage of the Organizational Innovation and Deployment PA. The link between the Organizational Innovation and Deployment PA and the Organizational Process Focus PA is illustrated.

This chapter is also used to summarize the discussion of causal analysis and resolution and organizational innovation and deployment, their interactions with each other and their interactions with OPP and quantitative project management.

Chapter 22: Reducing Variation

This chapter presents an evolutionary path within the CMMI® model that illustrates how process improvement steps taken to move from an individual focus, to a project focus, to a measurement-oriented organizational focus, to a quantitative management focus can be thought of as successive steps in reducing variation in an organization's processes and business results. The process areas of Project Planning, Project Monitoring and Control, Measurement and Analysis, Organizational Process Definition, Integrated Project Management, Organizational Process Performance, and Quantitative Project Management are used to support this chapter's concepts.

Part VII: Institutionalization

Chapter 23: Repeatable, Effective, and Long Lasting

This chapter examines the specific and generic goals and their associated specific and generic practices from both the staged and continuous representation point of view to help the reader more easily understand the concept of institutionalization of the process areas. The theme of implementing a process area or collection of process areas so that they are repeatable, effective, and long lasting is emphasized.

Part VIII: Advanced Concepts

Chapter 24: The Constagedeous Approach to Process Improvement

This chapter makes the case for the understanding and application of the principles of both the staged and continuous representations of the CMMI® in supporting an organization's process improvement initiative. An example of using an incremental approach to process improvement to achieve CMMI® Maturity Level 2 will be provided to illustrate the theme of constagedeous process improvement.

Chapter 25: Applying CMMI® to Manufacturing

This is a special chapter that presents a case study on how the CMMI® was successfully applied to manufacturing to help improve production processes in parallel to development processes and achieve CMMI® Maturity Level 3. The contributing authors are Sarit Assaraf and Itzhak Lavi from Israel Aerospace Industries, Ltd., in Israel.

Part IX: Summary

Chapter 26: Summary and Concluding Remarks

This chapter brings together the major themes that I have tried to share in this book. It presents the essence of the CMMI®. It reminds the reader of the importance of engaging in a process improvement initiative based on the CMMI® that is satisfying the organization's business objectives with measurable results. It indicates the importance of using the CMMI® in a "constagedeous" approach to process improvement by combining the features of the staged and continuous representations of the model. It presents the CMMI® as the overarching framework of a series of cascading process improvement umbrellas that can be used to guide an organization in determining what it needs to enable a successful process improvement journey.

Appendices

SEI Disclaimer

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Artech House was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

CMM®, CMMI®, Capability Maturity Model®, Capability Maturity Modeling® OCTAVE®, Carnegie Mellon®, CERT®, and CERT Coordination Center® are registered marks in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Architecture Tradeoff Analysis Method, ATAM, CMM® Integration, COTS Usage Risk Evaluation, CURE, EPIC, Evolutionary Process for Integrating COTS-based systems, Framework for Software Product Line Practice, IDEAL, Interim Profile, OAR, Operationally Critical Threat, Asset, and Vulnerability, Evaluation, Options Analysis for Reengineering, Personal Software Process, PLTP, Product Line Technical Probe, PSP, SCAMPI, SCAMPI Lead Assessor, SCAMPI Lead Appraiser, SCE, SEI, SEPG, Team Software Process, TSP are service marks of Carnegie Mellon University.

Special permission to use the following materials in *Practical Insight into CMMI®, Second Edition,* is granted by the Carnegie Mellon University Software Engineering Institute: Characterizing the Software Process: A Maturity Framework, SEI-87-TR-11, ESD-TR-87-112 © June 1987, CMMI® for Development, Version 1.2, CMU/SEI-2006-TR-008, ESC-TR-2006-008 © August 2008, CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1 Continuous Representation, CMU/SEI-2002-TR-011, ESC-TR-2002-011© March 2002, History of CMM®s, Figure 2.6, from CMMI® for Development, Version 1.2 CMU/SEI-2006-TR-008, ESC-TR-2006-008 © August 2006, CMMI v1.2 Architectural Framework, Figure 4.1, © December 2005, Data Types, Figure 19.11, from “Designing Products and Process Using Six Sigma”, © 2007, Types of Hypothesis Tests, Figure 19.12, from “Designing Products and Process Using Six Sigma”, © 2007, Confidence and Prediction Intervals – Example, Figure 19.13, from “Understanding High Maturity Practices: Statistical Thinking” © 2007, Organization and Project Measurement Flowdown to Critical Process and Subprocesses, Figure 20.4, © 2007, Qualitive and Quantitative Models to Process Performance Models, Figure 20.5, © January 2008.

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR

PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

The Software Engineering Institute and Carnegie Mellon University do not directly or indirectly endorse the work of Kasse Initiatives LLC.

PART I

Introduction

CHAPTER 1

Engineering Systems Thinking

Engineering systems thinking is a discipline for seeing the whole. It is a framework for understanding interrelationships and repeated events rather than seeing things in isolation. Engineering systems thinking is necessary for successful product development and for successful process improvement. The merger of systems engineering, hardware engineering, and software engineering concepts, examples, and process improvement ideas has resulted in the development of the CMMI® for Development Version 1.2 or CMMI®-DEV v1.2. This chapter provides a brief overview of the systems engineering, hardware engineering, and software engineering sources that were merged to develop the CMMI®-DEV v1.2. In addition, it focuses on the concept of "engineering systems thinking" to guide us to not only try to find the easiest solution, but one that takes into consideration the system being built, the organization receiving it, and the environment in which it must operate.

CMM® for Software

The Software Engineering Institute (SEI) as a part of Carnegie Mellon University (CMI) was established in 1984 as a federally funded research and development center (FFRDC) with the mission to provide leadership in advancing the state of the practice of software engineering in order to improve the quality of systems that depended on software.

CMM® v1.0 to CMM® v1.1

Early questionnaires and model beginnings focused predominantly on software engineering processes. The Capability Maturity Model® (CMM®) for Software v1.0 that was released in August 1991 strongly referenced the link to the overall system and the requirements from which the software developers were supposed to work. These requirements were referred to as "systems engineering requirements allocated to software."

The Need for a Systems Engineering CMM®

While the use of the CMM® for Software grew rapidly worldwide, it became very clear, very fast, that the strict focus on software engineering did not satisfy the business or systems needs of the companies that were developing software-intensive systems, nor the ones that were developing predominantly hardware systems. Various CMM® models were developed for a myriad of disciplines other than software. These included models for systems engineering, software acquisition, workforce management and development, and integrated product and process development.

Although these models proved useful to many organizations, the use of multiple models also became problematic. Organizations desiring to focus their improvement efforts across the disciplines within their organizations ran into difficulties. The differences among the discipline-specific models, including their architecture, content, and approach, limited the ability of those organizations to focus their process improvements efforts successfully. In addition, the application of multiple models that were not consistently integrated within and across an organization resulted in higher costs in terms of training, appraisals, and improvement activities. A model that included practices, references, and examples for multidisciplined projects was needed.

The Need for an Integrated Model

The SEI formed the CMM® Integration project in 1997 to sort out the problem of using multiple CMMs®. The CMM® Product Team's mission was to combine three source models—(1) Capability Maturity Model® for Software (SW-CMM®) v2.0 draft C, (2) Electronic Industries Alliance Interim Standard (EIA/IS) 731, and (3) Integrated Product Development Capability Maturity Model® (IPD-CMM®) v0.98—into a single improvement framework for use by organizations pursuing enterprise-wide process improvement. CMMI® v1.1 was originally developed to cover product and service development and maintenance and to provide an extensible framework, but its focus was on four bodies of knowledge:

- Systems engineering;
- Software engineering;
- Integrated product and process development;
- Supplier sourcing.

Systems Engineering

Systems engineering (SE) covers the development of total systems that may or may not include software. Systems engineers focus on transforming customer needs, expectations, and constraints into product solutions and supporting these product solutions throughout the entire life cycle of the product.

Software Engineering

Software engineering (SW) covers the development of software systems. Software engineers focus on the application of systematic, disciplined, and quantifiable approaches to the development, operation, and maintenance of software.

Integrated Product and Process Development

Integrated product and process development (IPPD) is a systematic approach that achieves a timely collaboration of relevant stakeholders throughout the life of the product to better satisfy customer needs, expectations, and requirements. The processes to support an IPPD approach are integrated with the other processes in the organization. The IPPD process areas, specific goals, and specific practices alone cannot achieve IPPD. If a project or organization chooses IPPD, it performs the IPPD-specific practices concurrently with other specific practices used to produce products (e.g., the engineering process areas). That is, if an organization or project wishes to use IPPD, it chooses a model with one or more other disciplines in addition to selecting IPPD.

Supplier Sourcing

When the work effort becomes complex, projects may have to use suppliers to perform functions or add modifications to products that are specifically needed by the project. When those activities are critical, projects benefit from monitoring the supplier's selected processes and subprocesses and evaluating corresponding selected work products before product delivery. The supplier sourcing (SS) body of knowledge was developed to support the acquisition of products and services from suppliers.

Generating Multiple Models from the CMMI® Framework

The CMMI® framework was also created to provide the ability to generate multiple models. The models either focused on a single discipline or a combination of the bodies of knowledge. Some of the combinations of the models that were the most popular included:

- CMMI®-SE/SW;
- CMMI®-SE/SW/IPPD/SS;
- CMMI®-SE/SW/SS.

The need for the integrated model became more apparent as the CMMI® evolved into the set of models referenced earlier.

A number of companies did set a precedent with their belief and approach to systems, software, and business goals. They integrated all three using a unified quality management approach in their process improvement initiatives. Among those are Motorola, whose Microsystems business unit started developing "product life cycles" in 1985 that included systems, software, hardware, marketing, and manufacturing. In 1990, AT&T realized an increase in productivity and product quality by creating integrated teams that forced marketing, systems, software, and hardware representatives to work together and to be accountable as a team for the delivery of the product.

In addition, integrating systems engineering and software engineering activities enabled Ford Aerospace between 1989 and 1992 to regain its competitive position in the command and control market place and reach CMM® Maturity Level 3 at the same time.

Hardware and Other Engineering Disciplines

Systems engineering was described as covering the development of a total system; transforming customer needs, expectations, and constraints into product solutions; and supporting these product solutions throughout the entire life cycle of the product. A large part of the systems industry accepted that the CMM® for Systems Engineering and the reference to systems engineering in CMMI® v1.1 covered all of the engineering disciplines that were required to build products in their company. Others, however, focused on a more narrow definition of systems engineering and demanded that the SEI provide more specific guidance in the CMMI® for hardware and other engineering disciplines.

CMMI®-DEV v1.2, released by the SEI in August 2006, now includes a definition of hardware engineering, six hardware amplifications to add emphasis to CMMI® descriptions for the hardware discipline, and various hardware examples integrated through all of the CMMI®-DEV model components. The CMMI®-DEV glossary defines hardware engineering as, "the application of a systematic, disciplined, and quantifiable approach to transform a set of requirements representing the collection of stakeholder needs, expectations, and constraints using currently available techniques and technology to design, implement, and maintain a tangible product" [1].

According to the CMMI®, hardware engineering represents all of the technical fields that transform requirements and ideas into tangible and producible products. I have had the privilege of supporting an organization that applied the CMMI® principles to the entire company consisting of six engineering disciplines plus production across four divisions. They were highly successful when they applied the CMMI® to *all* of the engineering disciplines. I have also had the privilege to support an organization applying the CMMI® to a predominantly manufacturing organization. See Chapter 25 for an account of that experience. To offer a bit more background to the readers for the other hardware disciplines these definitions are included here [2]:

- *Chemical engineering*. The design, construction, and operation of plants and machinery for making such products as acids, dyes, plastics, and synthetic rubber by adapting the chemical reactions discovered by the laboratory chemist to large-scale productions. The chemical engineer must be familiar with both chemistry and mechanical engineering.
- *Mechanical engineering*. The design, construction, and operation of power plants, engines, and machines. It deals mostly with things that move. One common way of dividing mechanical engineering is into heat utilization and machine design. The generation, distribution, and use of heat are applied in boilers, heat engines, air conditioning, and refrigeration. Machine design is concerned with hardware, including the use of heat processes.
- *Electrical engineering*. Electrical engineering encompasses all aspects of electricity from power engineering, the development of devices for the generation

and transmission of electrical power, to electronics. Electronics is a branch of electrical engineering that deals with devices that use electricity for control of processes. Subspecialties of electronics include computer engineering, microwave engineering, communications, and digital signal processing.

- *Aeronautical engineering*. The designing of aircraft and missiles and in directing the technical phases of their manufacture and operation. Supporting aeronautical engineering is mineral engineering, which includes mining, metallurgical, and petroleum engineering, which are concerned with extracting minerals from the ground and converting them to pure forms.
- *Optics (optical engineering)*. The field of optics usually describes the behavior of visible, infrared, and ultraviolet light; however, because light is an electromagnetic wave, analogous phenomena occur in x-rays, microwaves, radio waves, and other forms of electromagnetic radiation. Optics can thus be regarded as a subfield of electromagnetism. Optics, as a field, is often considered largely separate from the physics community. The pure science aspects of the field are often called optical science or optical physics. Applied optical sciences are often called optical engineering.
- *Hydraulics*. Hydraulics is a topic of science and engineering dealing with the mechanical properties of liquids. Hydraulics is part of the more general discipline of fluid power. Fluid mechanics provides the theoretical foundation for hydraulics, which focuses on the engineering uses of fluid properties. Hydraulic topics range through most science and engineering disciplines, and cover concepts such as pipe flow, dam design, fluid control circuitry, pumps, turbines, hydropower, computational fluid dynamics, flow measurement, river channel behavior, and erosion.
- *Other engineering disciplines*. Other important branches of engineering are agricultural engineering, engineering physics, geological engineering, naval architecture and marine engineering, and nuclear engineering.

Systems Engineering and Systems Management

We must first discuss the questions of “What is systems engineering and systems management?” and “What are some of the commonly accepted functions of a systems engineer?” in order to properly talk about “engineering systems think.”

The precise definition of systems engineering is not commonly agreed on throughout the industry or within the major companies that employ systems engineers. This has led to a great deal of confusion and long hours of wasted debates. Some of the common and not-so-common names for individuals who work in systems engineering include:

- Systems engineers;
- Systems architects;
- Systems integrators;
- Systems management engineers;
- Systems quality assurance engineers;

- Systems theorists;
- Systems reengineers;
- Operations research (closely related profession);
- Management science (closely related profession).

We will try to clarify that confusion. First, let us take a look at the engineering of systems. Systems engineering is concerned with the engineering of systems. Systems management is concerned with strategic-level management of systems engineering. Systems engineering efforts therefore should involve:

- Systems engineering method and tools or technologies;
- Systems process;
- Systems management.

These three levels of systems engineering are illustrated in Figure 1.1.

Systems engineering can be viewed as a management technology. If we break management technology down into its components we will be able to derive a clearer understanding of systems engineering.

Technology

Technology is the organization, application, and delivery of scientific knowledge as well as other forms of knowledge for the betterment of a client group. Technology can be viewed fundamentally as a human activity.

Management

Management involves the interaction of the organization with the environment. The purpose of management is to enable organizations to better cope with their environments in order to achieve purposeful goals and objectives.

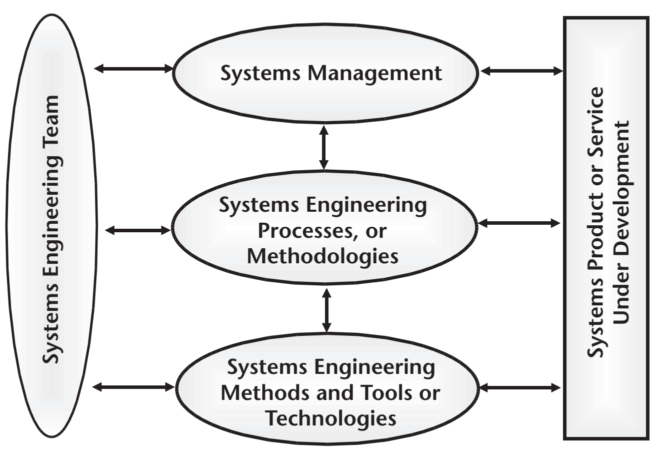


Figure 1.1 Three levels of systems engineering. (*From:* [3]. © 1999 John Wiley & Sons, Inc. Reprinted with permission.)

Management of Technology

The management of technology (Figure 1.2) involves the interaction of:

- Technology;
- Organizations, which are collections of people concerned with the evolution and use of technologies;
- Environment.

Systems Engineering Definition

Thus, if we combine all of the definition pieces we have just presented, we can define systems engineering to be the management technology that controls a total system life-cycle process, that evolves and results in the definition, development, and deployment of a system that is of high quality and is cost effective and on schedule in order to meet the user's needs.

Engineering Systems Thinking

During the past 7 to 10 years, the author has worked with a large number of companies that did not have a strong engineering discipline among its software systems developers or its management. While this did not seem to pose a problem when an organization was going through the initial phases of its process improvement initiative, it quickly manifested itself into a deep-seated problem when the organization attempted to develop its action plan. Working groups or process action teams (PATs) found it difficult to take the necessary steps to move from assessment recommendations to implementation tasks that were:

- Described in enough detail to be placed into a project-level plan (complete with milestones and deliverables);

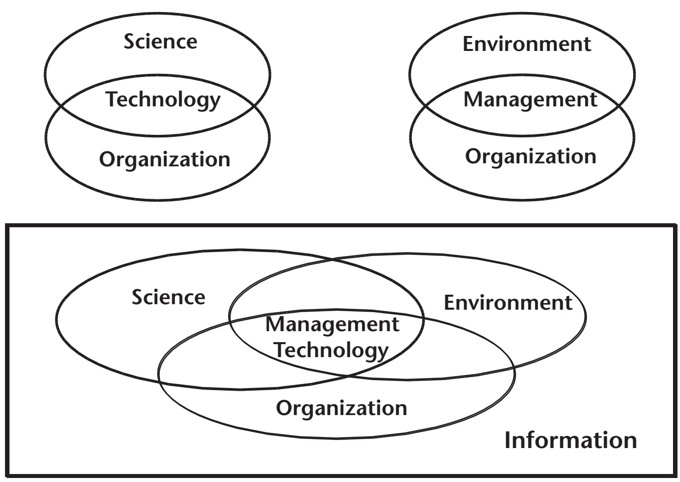


Figure 1.2 Management technology. (*From:* [3]. © 1999 John Wiley & Sons, Inc. Reprinted with permission.)

- Prioritized according to organizational business needs and project criticality;
- Scheduled for implementation according to defined increments;
- Able to be implemented within the financial and resource constraints of the organization.

Guidance for Action Planning

The author developed a method for successfully helping an organization make the transition from process assessment to action planning and implementation calling it Guidance for Action Planning. This proved to be immensely helpful for those organizations that did not have a strong systems engineering orientation as mentioned earlier.

The concepts behind the Guidance for Action Planning come from the application of the engineering principle of decomposition of high-level system descriptions into more manageable subsystems and modules. In other words, this is the application of basic engineering principles to process improvement tasks. Yet, if an organization does not have both a management team and a workforce that embraces basic systems engineering thinking, it can prove to be painfully hard, if not impossible, to develop actionable process improvement plans.

The Guidance for Action Planning provides a "starter kit" for development of the action plan or plans for individual focus areas. It provides the vital intermediate details to get the organization going in the right direction, thus enabling it to implement and to complete its action planning within a reasonable time frame. It provides direction for the organization's senior management team and the process improvement champions to ensure that coordinated action will result from the assessment effort, and that this effort will support the organization's business objectives.

The Guidance for Action Planning also provides senior management with a "big picture" of the requirements for process improvement by showing what needs to be done, who needs to be involved, and what it might take to accomplish true and lasting improvement. This analysis provides a process improvement road map to support management decision making by clarifying how process improvement priorities can map to the corporate vision and business environment (Figure 1.3).

The Guidance for Action Planning also provides important information for the software/systems engineering process group (SEPG) and others involved in the development of the action plan because it defines the initial steps in the development of the focus area sections of the action plan. These steps are direct transitions from the assessment results. The Guidance for Action Planning also assists the SEPG's facilitation of the development and implementation of the action plans that will most support the process improvement needs of the organization. The Guidance for Action Planning road map provides the bridge between the assessment results and the activities necessary to develop actionable plans for improvement.

Systems Thinking

Systems thinking is a discipline for seeing the whole. It is a framework for understanding interrelationships and repeated events rather than seeing things in isolation. Systems thinking is about seeing patterns of change rather than static

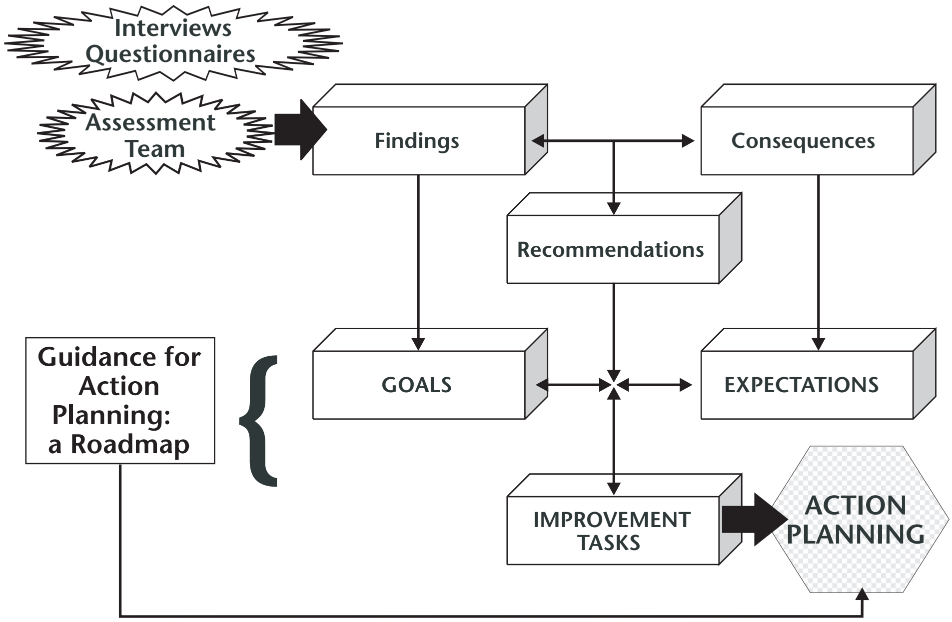


Figure 1.3 Guidance for Action Planning℠ road map.

snapshots. Systems thinking embodies the idea that the interrelationships among the parts relative to a common purpose of a system are what is important. Systems thinking is necessary for successful product development and for successful process improvement.

The Fifth Discipline

Peter Senge led a team of authors to create a book titled *The Fifth Discipline, The Art and Practice of the Learning Organization* to describe a path for the "learning organization." According to Peter Senge, systems thinking is the "fifth discipline": "… and is the catalyst and cornerstone of the learning organization that enables success through the other four disciplines" [4]:

- Personal mastery through proficiency and commitment to lifelong learning;
- Shared mental models of the organization markets and competitors;
- Shared vision for the future of the organization;
- Team learning.

Peter Senge describes certain truths or laws about the fifth discipline. A few of them are included here to support the concepts of engineering systems thinking.

- Short-term improvements often lead to long-term difficulties. When organizations only focus on the "low-hanging fruit," they frequently do not establish the basics functions necessary to continue long-term future growth.
- The easy solution may be no solution at all.

- Quick solutions, especially at the level of the symptoms, often lead to more problems than existed initially.
- Cause and effect are not necessarily closely related, either in time or in space. Sometimes solutions implemented here and now will have impacts far away at a much later time.
- The entire system, comprised of the organization and its environment, must be considered together.

Laws of Engineering Systems Thinking

Engineering systems thinking is not merely a slogan; it is composed of basic "laws" that can and should be applied to everyday business life. Here is a summary of some of the more important engineering systems thinking laws according to the author's educational background and professional experience. Process areas that have been defined in the CMM® Integration (CMMI®) appear in parentheses as indicators of the effort undertaken in the development of the CMMI® in order to incorporate engineering systems thinking.

1. In all of the project's phases/stages, and along the system's life, the systems engineer has to take into account [Requirements Development (RD)]:
   - The customer's organization vision, goals, and tasks;
   - The customer's requirements and preferences;
   - The problem to be solved by the system and the customer's needs.
2. The whole has to be seen as well as the interaction between the system's elements. Iterative or recursive thinking must replace the traditional linear thinking [RD – Technical Solution (TS)].
3. Project members should always look for the synergy and the relative advantages from the integration of subsystems [TS – Product Integration (PI)].
4. The solution is not always an engineering one. Remember to take into account:
   - Business and economic costs;
   - Reuse or utilization of products and infrastructure already developed;
   - Organizational, managerial, political, and personnel considerations.
5. The systems engineer should consider as many different perspectives as possible.
6. Systems engineers and project managers should always take into account (RD – TS):
   - Electrical considerations;
   - Mechanical considerations;
   - Environmental considerations;
   - Quality assurance considerations;
   - Quality factors such as reliability, maintainability, expandability, and testability.

7. Future logistic requirements should be evaluated in all development phases including (RD – TS):
   - Spare parts;
   - Maintenance infrastructure;
   - Support;
   - Service;
   - Maintenance levels;
   - Technical documentation.
8. When a need arises to carry out a modification to the system, always take into account (RD – TS – Requirements Management):
   - The engineering and nonengineering implications;
   - The effects on the form, fit, and function;
   - The system's response to the changes;
   - The needs, difficulties, and attitudes of those who must live with the modification.
9. Each problem may have more than one possible working solution (TS):
   - All possible alternatives should be examined and compared to each other by quantitative and qualitative measurements.
   - Engineering trade-offs and cost effectiveness should be considered at every stage.
10. Systems engineers should be encouraged to look for the changes that might introduce significant improvements with minimum effort.
11. Processes that result in slow or gradual results should be taken seriously [Organizational Innovation and Deployment (OID)].
12. A known solution is not always suitable for the current problem. Each available solution should be considered along with the risks, dependencies, and constraints inherent in the evolving system [Generic Practice (GP 2.2)].
13. Development risks must be taken into account throughout the entire product development cycle [Project Planning – Risk Management (RSKM)].
14. It is impossible to run a project without [PP – Project Management and Control (PMC) – Configuration Management (CM) – Generic Practices)]:
   - Control;
   - Configuration management;
   - Milestones;
   - Management;
   - Scheduling methods.
15. The end user must be considered a major part of the system (RD – PP).
16. Engineering systems thinking requires the integration of expertise from different disciplines and requires the examination of different perspectives, calling for teamwork to cover those perspectives [Integrated Project Management + Integrated Product and Process Development (IPM + IPPD)].

17. Selecting partners and subcontractors is critical. Do not enter into a partnership unless the partner is willing to share your risks as well as your successes and profits [Supplier Agreement Management (SAM)].
18. Limit the responsibility assigned to an external factor because this increases the dependency on it [SAM – Commercial off-the-shelf software (COTS)].
19. Take the shelf life into consideration when selecting components for production.
20. Engineering systems thinking includes probability and statistics both when defining the systems specifications and when determining the project targets such as cost and performance [Decision Analysis and Resolution (DAR)].

Summary

One of the most prominent problems observed in software and software/systems organizations today is the lack of engineering discipline, as cited by managers at all management levels. The CMM® for Software was developed to encourage organizations to develop processes to guide its software development. The evolution of the CMMI®, now the CMMI®-DEV v1.2, recognizes that engineering systems thinking is an important asset for building systems and has put the “engineering” representing all necessary engineering disciplines back into process improvement.

References

[1] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI® Guidelines for Process Integration and Product Improvement,* 2nd ed., Reading, MA: Addison-Wesley, 2007.

[2] Columbia Encyclopedia Online, Columbia University Press, “Engineering,” January 2008, http://education.yahoo.com/reference/encyclopedia/entry/engineer.

[3] Sage, A. P., and W. B. Rouse, *Handbook of Systems Engineering and Management*, New York: John Wiley & Sons, 1999.

[4] Senge, P. M. P., et al., *The Fifth Discipline: The Art and Practice of the Learning Organization*, New York: Doubleday, 1990.

CHAPTER 2

Oriented to Business Results

The software industry has criticized the CMM® for Software model for its lack of focus on business results. Other total quality management models such as Malcolm Baldrige, the European Foundation for Quality Management, the Singapore Quality Award, and the Israel Quality Award place heavy emphasis on business results, not just on documenting processes to gain industry certification. The CMMI® clearly and repeatedly states the need for all process improvement activities to measurably support the organization's business objectives.

Today, more than ever in the past, companies feel pressure to deliver products and services faster and faster. Time to market has become a "mantra" for many companies throughout the world. A focus on quality and business results faces intense pressure from those who claim they no longer fit in "the real world." Yet along with the intense pressure to provide products and services faster, the high-technology industry is also demanding more features and functions at the cheapest price possible. Many companies have found that these demands are more than they can handle by themselves, and are therefore turning to suppliers, subcontractors, and outsourcing partners to augment their capabilities in an attempt to satisfy their customers. However, this solution requires that the organization be willing and able to manage all of the outside sources of products or product components and services. Effective management, critical to business success, requires an understanding and application of an integrated approach.

Engineering Competency

During the past 10 years, the author's process improvement consultancy activities with engineering firms, financial institutions, insurance companies, IT companies and business units, and government software/systems developers have revealed a significant lack of engineering competency. Overall, there appears to be an inability to manage the engineering processes. In many cases, organizational process definitions are poor to nonexistent. In others, the processes appear to be very well developed but are not trained, supported, or enforced. As a result, these processes are not uniformly used and are not "living." A common complaint among managers at all levels is that there is a lack of engineering discipline. However, then they declare that a systems approach to problem solving appears to be useful only for "the other guy." The comment we hear often is: "… but my project has such great pressures that we don't have time to think of alternatives and do things in a systems way."

Quality management and the underlying quality functions are misunderstood and are poorly implemented throughout the life cycle. Many managers believe that quality assurance lies with the testing group. The difference between quality control (evaluates the products and life-cycle work products—peer reviews and testing) and quality assurance (evaluates the processes—evaluations and assessments) is seldom clearly understood. Measurement programs are often implemented late and are poorly focused. Too often, I am told that a business unit's focus is on hours spent on reviews, but not on the result of those reviews. In addition, even if processes exist and attempts are made to follow them, efficiency and effectiveness measurements are rarely implemented.

An example will serve to illustrate this critical lack of understanding and use of measurements. A quality assurance manager in a large IT department routinely reported on the number of audits and reviews that he and his staff participated in each month. When confronted with the question "So what?" he replied that often the quality engineers had spent more than 50 hours per week on those activities. It was then suggested that for his quality group's activities to be considered effective he might concentrate on how many serious noncompliances were uncovered by these quality assurance activities and how many of those noncompliances his staff helped fix.

The reporting of numbers of audits and hours of work certainly represents "status." The result of following the quality assurance processes and producing results such as the elimination of noncompliances in a given reporting period and increasing the probability of producing a high-quality product indicates effectiveness.

Another engineering competency weakness that we have seen is that people are frequently hired to address a specific project need. There is little evidence that the people are hired because of an organizational understanding of the core competencies needed to support business objectives. Indeed, the facts seem to indicate that people are routinely hired without an acceptable percentage of core competency skills and, even worse, they are hired without a demonstrable ability to learn and master core competency skills.

Quality Management and Continuous Improvement Principles

In the 1930s, Walter Shewhart began work in process improvement with his principles of statistical quality control. The Shewhart cycle [1] was developed and published by Walter Shewhart in 1939 (Figure 2.1). The concepts were documented in his publication *Statistical Method from the Viewpoint of Quality Control,* written when he was part of the Department of Agriculture's graduate school in Washington, D.C. The Shewhart cycle is helpful as a procedure to follow for improvement at any stage and as a procedure for finding a special cause detected by a statistical signal. In 1950, Dr. Deming took the concepts of the Shewhart cycle to Japan to support the manufacturing companies in understanding and applying the concepts of statistical quality control. These concepts soon became known as the Deming cycle and are still referred to in quality engineering literature that way today. If one takes a small step back, it is not hard to see that the CMMI®-DEV v1.2 is a series of plan–do–check–act (PDCA) applications that build subpractices, practices, goals, and process areas through all of the capability and maturity levels.

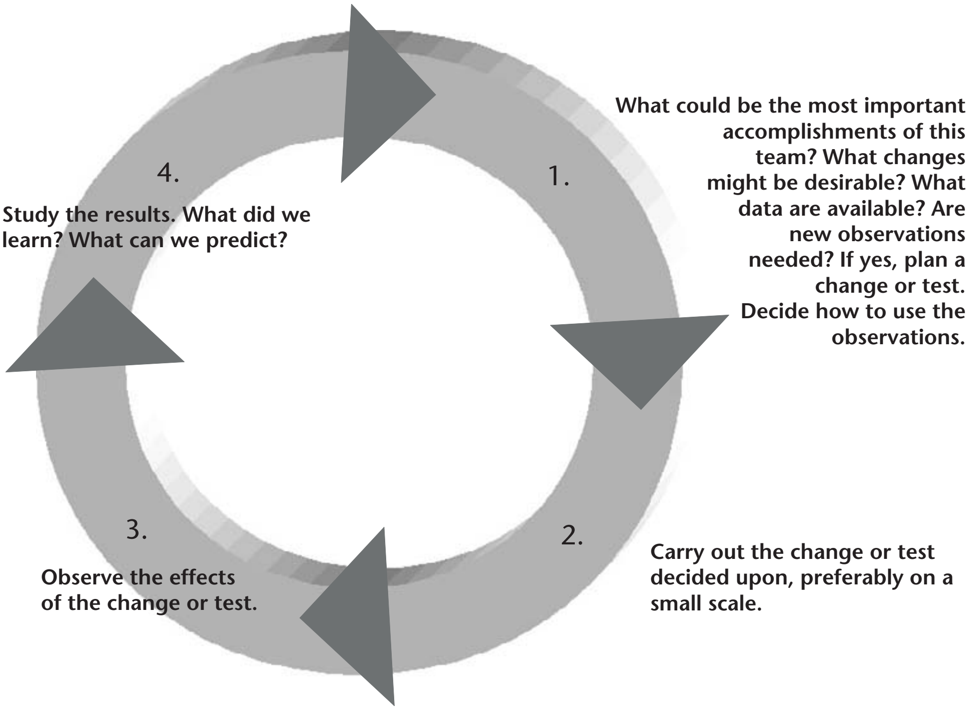


Figure 2.1 The Shewhart cycle. (*From:* [1]. © 1939 W. A. Shewhart. Reprinted with permission.)

These principles were refined by W. Edwards Deming, Phillip Crosby, and Joseph Juran. Watts Humphrey[1], Ron Radice, and others extended these principles even further and began to apply them to software in their work at IBM and the SEI. Humphrey's book, *Managing the Software Process* [2], identifies process improvement principles on which most of the capability maturity models are based today.

The SEI adopted an often repeated process management premise—"The quality of a system or product is highly influenced by the quality of the process used to develop and maintain it"—and embedded it into all of the CMMs® that it has produced. Deming's 14 points for management provide the source for this process management premise. Deming's third point stated "Cease dependence on inspection to achieve quality" [3]. Routine 100% inspection to improve quality is equivalent to planning for defects and acknowledgment that the process does not have the capability required for the specifications.

End item inspection to improve quality is too late, ineffective, and costly—when the product leaves the door of the supplier, it is clearly too late to do anything about quality. As Dr. Deming frequently stated, "You cannot inspect quality into a product" [3]. Quality comes from improvement of the process!

Although the early thinking on process improvement and quality management is correctly attributed to leaders such as Shewhart, Deming, Juran, and Crosby, in fact,

1. Watts S. Humphrey founded the Software Process Program of the Software Engineering Institute (SEI) at Carnegie Mellon University. He is a Fellow of the Institute and is a research scientist on its staff. From 1959 to 1986 he was associated with IBM Corporation where he was director of programming quality and process. His publications include many technical papers and more than 100books. He received the 2003 National Medal of Technology for his contributions to the software engineering community.

the man most indirectly responsible for promoting quality management and statistical quality control was General Douglas MacArthur. General Douglas McArthur was the supreme commander of the Allied Powers in Japan after World War II. General MacArthur wanted a large number of highly reliable radios manufactured quickly and at a reasonable cost so that a radio could be in every Japanese home in order to broadcast American propaganda. But the Japanese manufacturing companies could neither achieve the quantity nor the quality of radios that MacArthur desired. While looking for a solution to this problem, General MacArthur was introduced to Dr. Walther Shewhart, who had invented the control chart for predicting stable processes in manufacturing. Dr. Shewhart was unable to go to Japan himself, but sent the best men available. Those "best" included Homer M. Sarasohn, Charles W. Protman, and W. Edwards Deming who had been a graduate student of Dr. Shewhart's. General MacArthur then ordered the top management of Japan's best manufacturing firms to attend an 8-week course in statistical quality control (SQC). During this often intensive 8 weeks, the top leaders of Japan's manufacturing industry were taught to focus first on quality. MacArthur got his high-quality radios mass produced and the Japanese continued to focus on quality. They have continued to establish themselves as quality leaders in the world for many products and services.

The Five-Level Model

Many people the world over have wondered how the SEI came up with its original five-level maturity model. It turns out that "having five fingers or five toes" was not the motivation that Watts Humphrey had utilized for establishing its structure. This was not the basis he had passed along to the team that helped produce the CMM® for Software and the Software Process Assessment Method in the early days of the SEI. Rather the initial CMM® for Software (CMM®) was based on the work of Phillip Crosby as documented in his book *Quality Is Free* [4].

Phillip Crosby worked for International Telephone and Telegraph (IT&T) in manufacturing. He developed a method that would focus both managers and production personnel alike on a common path to increase measurable product quality. His maturity grid, partially completed in Figure 2.2, had different names for the five levels. However, the "Cost of quality as a percentage of sales" is still one of the driving forces for organization's to achieve CMMI® Maturity Level 5 today. Crosby's focus, like Deming's, was quality first. On October 6, 1986, Humphrey started to apply software engineering concepts to that five-level model concept. This resulted in the first SEI maturity framework (Figure 2.3).

The ideas behind the model were summarized from *Characterizing the Software Process: A Maturity Framework* [5]. Instead of characterizing the maturity levels, Humphrey provided the motivation to move to the "next" maturity level. So, if your organization was at the Initial level, it was suggested that the organization focus its work on project management, management oversight, product assurance, and change control.

Figure 2.4 represents the next evolutionary step, and Figure 2.5 shows the structure of the CMM® for Software version 1.0 as released for public use in August 1991. The CMMI® has evolved from these roots, as shown in Figure 2.6, into the

Measurement Categories	Stage I: Uncertainty	Stage II: Awakening	Stage III: Enlightenment	Stage IV: Wisdom	Stage V: Certainty
Management understanding and attitude					
Quality Organization Status					
Problem Handling					
Cost of Quality as % of sales	20	18	12	8	2.5
Quality Improvement Actions					
Summation of company quality posture	We don't know why we have problems with quality				We know why we do not have problems with quality

Figure 2.2 Quality management maturity grid. (*From:* [4]. © 1979 McGraw-Hill. Reprinted with permission.)

Maturity Level	Key actions required to advance to the next level
Optimized	
Managed	Automate process data collection, turn management focus from product to process
Defined	Process Measures, Process Database, Measurement support, Product Quality targets and assessment
Repeatable	Process Group, Process Architecture, Software Engineering methods and technologies
Initial	Project Management, Management Oversight, Product Assurance, Change Control

Figure 2.3 A maturity framework. (*After:* [5].)

CMMI®-DEV constellation for development and maintenance that is part of CMMI® v1.2 today, as shown in Figure 2.7.

It is important to observe that while Crosby's model was always focused on quality management, all versions of the CMM® have indicated that for an organization to achieve a maturity level higher than initial or ML 1, a strong focus on quality is needed. Today, for the CMMI®, the process area is called Process and Product Quality Assurance and the focus has been expanded to include quality for all engineering disciplines and to include product quality as well as process quality.

Senior Management's Vision

A vision is a guiding image of success formed in terms of a contribution to society. A mission statement answers these questions: Why does our organization exist? What

Level	Characteristic	Key Challenges	Result
5 Optimizing	Improvement fed back into process	Still human intensive process Maintain organization at optimizing level	Productivity & Quality
4 Managed	(quantitative) Measured process	Changing technology Problem analysis Problem prevention	
3 Defined	(qualitative) Process defined and institutionalized	Process measurement Process analysis Quantitative quality plans	
2 Repeatable	(intuitive) Process dependent on individuals	Training Technical practices • reviews, testing Process focus • standards, process groups	
1 Initial	(ad hoc/chaotic)	Project Management Project Planning Configuration management Software quality assurance	Risk

Figure 2.4 Maturity level/key challenges. (*After:* [6].)

Level	Process Characteristics	Key Process Areas
5 Optimizing	Process improvement is institutionalized	Process Change Management Technology Change Management Defect Prevention
4 Managed	Product and process are quantitatively controlled	Software Quality Management Quantitative Process Management
3 Defined	Technical practices are integrated with management practices and institutionalized	Process Focus, Process Definition Training Program, Software Product Engineering Peer Reviews, Intergroup Coordination Integrated Software Management
2 Repeatable	Project management practices are institutionalized	Requirements Management, Project Planning Configuration Management, Quality Assurance Project Tracking and Oversight Subcontract Management
1 Initial	Process is informal and ad hoc	

Figure 2.5 Capability Maturity Model® overview. (*After:* [6].)

business are we in? What values will guide us? A vision, however, is more encompassing. It answers the question "What will success look like?" It is the pursuit of this image of success that really motivates people to work together.

The concept of visioning is not yet a comfortable concept for many senior managers, especially as it relates to process improvement. Yet, in recent years, visioning has become recognized as a major process improvement factor. If the process improvement initiative does not support management's vision, then the probability is very low that management will support process improvement over the long term.

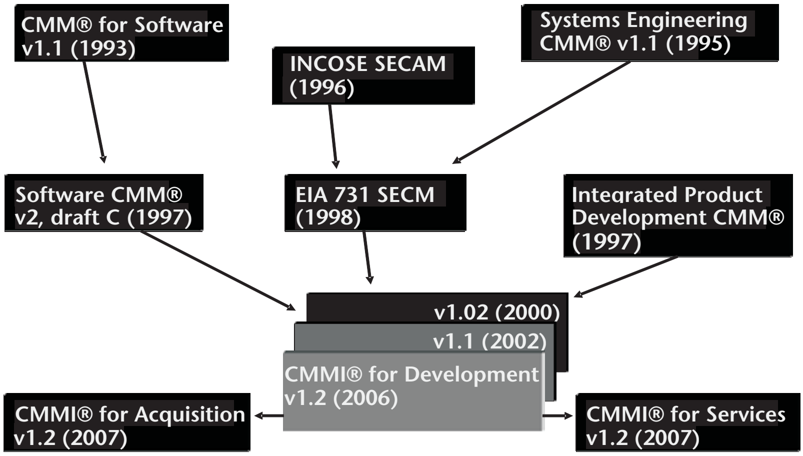


Figure 2.6 History of CMMs®. (*From:* [7]. © 2008 SEI. Reprinted with permission.)

Level	Process Characteristics	Process Areas	
5 Optimizing	Focus is on quantitative continuous process improvement	Causal Analysis and Resolution Organizational Innovation and Deployment	
4 Quantitatively Managed	Process is measured and controlled	Quantitative Project Management Organizational Process Performance	
3 Defined	Process is characterized for the organization and is proactive	Requirements Development Technical Solution Product Integration Verification Validation Decision Analysis & Resolution	Organizational Process Focus Organization Process Definition Organizational Training Integrated Project Management Risk Management
2 Managed	Process is characterized for projects and is often reactive	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management	Product and Process Quality Assurance Configuration Management Measurement & Analysis
1 Initial	Process is unpredictable, poorly controlled, and reactive		

Figure 2.7 CMMI®-DEV v1.2 overview.

In today's highly competitive world, organizations are merging and/or downsizing. Employees of these evolving organizations want to know how their efforts can contribute to the health and well-being of their organization. Some senior managers have a strong vision, but often it is not communicated down to the practitioners. Some practitioners have a fuzzy idea of what the vision must be, but are not able to articulate it. Some senior management teams express great disagreement as to what the common vision should be.

Visioning is such an important concept for today's business that its value cannot be overemphasized. The following questions may help an organization to define its vision:

- Where does senior management think the organization will be in the next year, and in the next 2 to 5 years?
- What products will be in the mainstream?
- Who will be the competitors?
- Will there be collaborators or strategic alliance partners?
- What technology changes are expected and/or will be required to support the vision?
- What does the organizational structure have to be to support this vision?
- Who will be the organization's suppliers?
- What must the organizational culture be to support this vision?
- How will a process improvement initiative support this vision?

Organization's Business Objectives

For a process improvement initiative to be successful, it must be tied to the organization's business objectives. Understanding and supporting the business objectives play important roles in the definition of critical process areas such as Organizational Process Focus, Organizational Process Definition, Organizational Training, Organizational Process Performance, Organizational Innovation and Deployment, Risk Management, Quantitative Project Management, Requirements Development, Technical Solution, Measurement and Analysis, and Decision Analysis and Resolution.

To get an understanding of the organization's business objectives, consider some questions that can be presented to the senior management team:

- What are the organization's highest priorities?
- What business consequences have resulted from weak or ineffective processes?
- What action is being taken to correct the cause?
- How can a focus on process improvement support the organization's business objectives?

Business objectives should be able to be stated in "everyday" terms. An example of a business objective would be "Reduce system errors that are discovered by customers." This business objective may have been derived from the statistics of the last two or three releases in which large numbers of errors were being reported by the customers. Having the organization's business objectives clearly stated and documented helps ensure that the process improvement effort will be aligned with those business objectives and will result in desired business results.

Examples of common business objectives include:

- Reduce time to market;

- Reduce system errors that are discovered by customers;
- Improve delivery time;
- Increase quality of products;
- Find and fix software defects once and only once;
- Reduce project risks;
- Gain control of suppliers;
- Improve service delivery;
- Improve service availability and capacity;
- Shorten find-to-fix repair rate.

Figure 2.8 presents this business objectives orientation in the form of an upgraded people–process–technology perspective developed by the SEI in the early development days of the CMM® for Software.

Organization's Measurement Objectives

For an organization to be able to show measurable improvement toward its business objectives, it must be able to define and implement appropriate measurement objectives. The project's own measurement objectives and their link to the organization's measurement objectives will be described in Chapter 18. An organization's measurement objectives might include:

- Reduce time to delivery to a specified percentage.
- Reduce total life-cycle costs of new products by a percentage.
- Deliver specified functionality by a specified increased percentage.
- Improve prior levels of quality by reducing the number of defects of type A that get shipped with the product.

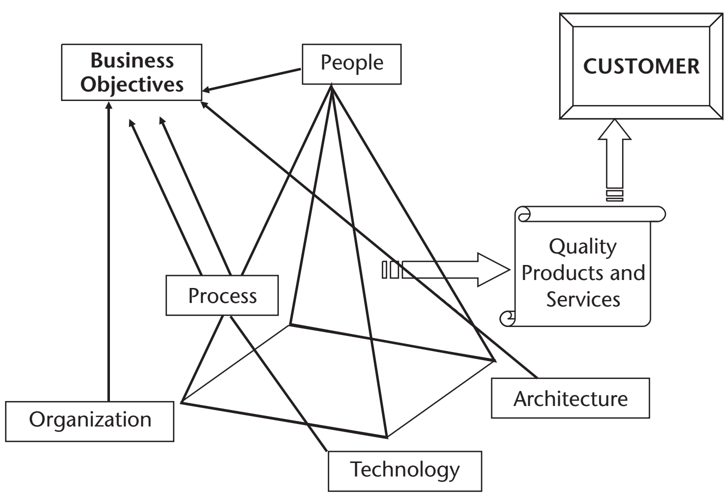


Figure 2.8 Business process perspective.

- Improve prior customer satisfaction ratings by a specified percentage compared to past ratings.

Support for Project Leaders to Manage and Control Better

Clearly if the CMMI®-based process improvement initiative is to support achieving business results, an organization must be able to show what measurable value will be brought to the project leaders who bear the line responsibility for product delivery. Today's project leaders are under ever increasing pressure to deliver products that meet or better the schedule, for less than the budget allows, and with higher quality and certainly no less functionality. It only seems reasonable that if they are willing to support a process improvement initiative they should be able to expect some or all of the suggested results:

- More accurate schedules;
- Higher productivity of developers;
- Better quality products;
- Traceable requirements;
- Controlled configuration items;
- Reviews focused on critical components;
- Better control of suppliers;
- Reduction of potential risks.

End-to-End Quality

While the CMMI® model is oriented to supporting organization's process improvement efforts in technical disciplines such as systems, hardware, and software engineering, it is being proven that its concepts can be readily transferred to the definition and control of an organization's business processes. Developing integrated systems are only one part of the organization's value chain that strives to provide end-to-end quality to its customers.

It can be argued that it is not sufficient for an organization to have high-maturity technical and project management processes, but must strive to have a set of integrated business processes for all of its departments and functions.

As each organization sets up its process improvement initiative, it should periodically assess its overall business process improvement efforts using questions like these:

- What business are you in?
- How does each department contribute to the business's success?
- How do these departments interact with each other to maximize company profit and achieve business goals?
- What business processes exist in each department to optimize its product quality and minimize interface conflicts?

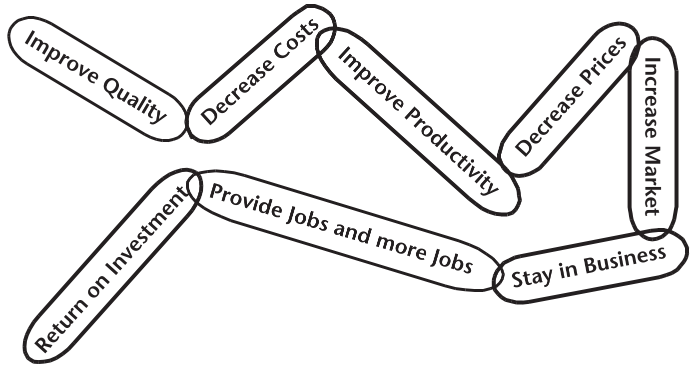


Figure 2.9 The Deming quality chain. (*After:* [3].)

- What standards and models are being used to accomplish daily tasks?
- What personnel processes are being used for each person to optimize his/her performance?
- Does each person understand his/her role in supporting the organization's business quality goals?

Dr. Deming tied business, process improvement, and quality together in his 14 points on quality. These TQM principles represent the backbone of the CMMI® as well. The essence of those 14 points is represented in Figure 2.9.

Summary

We conclude this chapter on business results with one more look at the Deming quality chain illustrated in Figure 2.9. Dr. W. Edwards Deming always advocated focusing on quality first. His belief was that if an organization continually focused on improving its product and service quality, it would in turn decrease costs, see improved productivity, be able to decrease prices, increase market share, stay in business, provide more and more jobs, and as a final result see the much sought after return on investment. It is the author's opinion that the CMMI® provides more guidance in its description of the process areas, goals, practices, and examples to support Dr. Deming's ideas of quality than any other model of its kind in the world.

References

[1] Shewhart, W. A., *Statistical Method from the Viewpoint of Quality Control*, Washington, D.C.: Graduate School of the Department of Agriculture, 1939.

[2] Humphrey, W. S., *Managing the Software Process*, Reading, MA: Addison-Wesley, 1989.

[3] Deming, W. Edwards, *Out of the Crisis*, Cambridge, MA: MIT Center for Advanced Engineering Study, 1986.

[4] Crosby, P. B., *Quality Is Free*, New York: McGraw-Hill, 1979.

[5] Humphrey, W. S., "Characterizing the Software Process: A Maturity Framework," SEI-87-TR-11, Pittsburgh, PA: Software Engineering Institute, 1987.

[6] Software Engineering Institute, Training Materials, 1989.

[7] CMMI® Product Team, CMMI® for Development, Version 1.2, CMU/SEI-2006-TR-008, ESC-R-2006-008, August 2008.

CHAPTER 3

Process Improvement Based on the CMMI®

Why base your organization's process improvement initiative on the CMMI®? The CMMI® represents integration of multiple military, ISO, IEEE, and commercial standards and procedures that cover all aspects of building systems. It is closely linked to ISO 9001:2000. The emphasis of the CMMI® is to support the development processes and changing of cultures to show a measurable benefit for the organization's business objectives and vision. The CMMI® provides a framework from which to organize and prioritize engineering, people, and business activities. It supports the coordination of multidisciplined activities that are or may be required to successfully build a project and it returns the concept of "engineering systems think" to project development. Process improvement is about supporting an organization's business with measurable results, and the CMMI® is about providing the necessary guidance to support successful organizational process improvement.

The CMMI® was developed to provide a single model to be used by organizations pursuing enterprise-wide process improvement including all of its divisions and business units. It provides needed guidance for integrating systems, hardware, and software development activities. It supports the coordination of multidisciplined activities that are or may be required to successfully build a project.

A Simple View of the CMMI®

If one were to take a high-level view of the CMMI® and its process areas without the interference of the staged or continuous representation that will be discussed later, it would not be hard to classify the CMMI® as project management, quality management, and engineering all glued together by process management. And until your organization reaches the business need to gain statistical control over processes or subprocesses and use models to predict future process and performance behavior, that is all the CMMI® is. Granted there are more than 100 years of theory and practice about management, quality, and engineering integrated into the CMMI®, and granted that it also focuses on the organizational development side of the equation to have a successful process improvement initiative as well as a concentration on improving the technical processes, but the model is not more complicated than that.

What seems to daunt organizations is the detail that is provided for each of the process areas. We all have "work to do" and do not see improving our processes as support of doing "the real work." Yet it is the details in the CMMI®'s front-end material, its description of institutionalization factors, its introductory notes, and its goals, practices, subpractices, and examples that make the CMMI® a one-of-a-kind process improvement model in the world today. There are many other standards and some other models that offer short and sweet sentences to give the reader a hint as to what is expected to reach high maturity and how these concepts might help the organization reach its business objectives, but none provide the guidance that the CMMI® provides today.

Some major industry process improvement standards today are merely subsets of high-level standards. What guarantee of process improvement and quality is expected from them? Or is the complaint of the size and complexity of the CMMI® simply a hope that intelligence and hard work on the part of the workforce will be sufficient to overcome all other business constraints?

Process and Business

Why use the CMMI® to support your organization's business objectives? Some reasons were provided in Chapter 2. Here are other thoughts for your consideration:

- Process defines how a business does business:
    - Software engineering processes;
    - Hardware engineering processes;
    - Systems engineering processes;
    - Manufacturing processes;
    - Financial processes;
    - Human resources processes;
    - Legal processes.
- Process helps to establish the business culture and then sets guidelines and expectations.
- Process can be viewed as a methodology that is applied from elicitation of requirements to design through delivery.
- Process helps the development and maintenance efforts to build in and retain the quality of the products and services and ensures profitability for the business.
- There are no shortcuts—there are no other alternative methods that a business can adopt that embraces a "cradle to grave" philosophy to ensure quality and profitability with *control* every step of the way.

The CMMI® through the Verification and Validation process areas states that organization's need to "build the product right" (i.e., meet the requirements) and "build the right product" (i.e., meet the customer's needs in the intended environment). We believe that senior management should adapt this model terminology for its business:

- We will build the business right—through process.

- We will build the right business—with guarantees of product and service quality and customer satisfaction.

Process is the fastest, lowest cost path to get there and know if you are there!

Are Process Descriptions Enough?

Organizations often emphasize the development of technical processes. And indeed, the CMMI® provides detailed practice descriptions to guide the development of those technical processes as well as others such as management processes and service processes. However, as previously stated, the main focus needs to be on the organization's vision and business objectives.

With that in mind, I would like to share the following scenario, which continuously repeats itself during my consulting work. I ask the following four questions:

1. Does your organization have process descriptions at the project and/or organizational level?
2. Does your organization provide training on the processes the project and support groups are supposed to follow?
3. Does your organization perform objective evaluations to ensure the processes that are defined and trained are being adhered too?
4. Does it work?

It is the last question that starts the debates, shouts and frustration. "What do you mean … Does it work?" I was asked.

I replied, "I mean, does it work? Are you meeting your schedules? Are you keeping with the budget constraints? Are you generating products with the expected quality or higher? Does it work?"

The next response is the one that amazes me the most. "The CMMI® does not say the processes have to work!"

I hope you are now smiling or outright laughing, but this point is not often stressed during assessments and process improvement support activities. Process improvement based on the CMMI® is about achieving measurable business results. Otherwise, there is little point in investing any resources in the process improvement initiative.

Other Groupings of Process Areas to Support Process Improvement

Process improvement based on the CMMI® is linked primarily to two categorizations of the process areas in the staged representation and the continuous representation. Figure 3.1 shows the standard CMMI® staged representation and Figure 3.2 shows the standard CMMI® continuous representation.

Both representations provide ways of implementing process improvement to achieve business goals. Both representations provide essentially the same content

Level	Process Characteristics	Process Areas	
5 Optimizing	Focus is on quantitative continuous process improvement	Causal Analysis and Resolution Organizational Innovation and Deployment	
4 Quantitatively Managed	Process is measured and controlled	Quantitative Project Management Organizational Process Performance	
3 Defined	Process is characterized for the organization and is proactive	Requirements Development Technical Solution Product Integration Verification Validation Decision Analysis & Resolution	Organizational Process Focus Organization Process Definition Organizational Training Integrated Project Management Risk Management
2 Managed	Process is characterized for projects and is often reactive	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management	Product and Process Quality Assurance Configuration Management Measurement & Analysis
1 Initial	Process is unpredictable, poorly controlled, and reactive		

Figure 3.1 CMMI®-DEV v1.2 staged representation.

Module	Process Areas
Process Management Concepts	Organizational Process Focus Organizational Process Definition Organizational Training Organizational Process Performance Organizational Innovation and Deployment
Project Management Concepts	Project Planning Project Monitoring and Control Supplier Agreement Management Integrated Project Management Risk Management Quantitative Project Management
Engineering Concepts	Requirements Development Requirements Management Technical Solution Product Integration Product Verification Validation
Support Concepts	Configuration Management Process and Product Quality Assurance Measurement and Analysis Decision Analysis and Resolution Causal Analysis and Resolution

Figure 3.2 CMMI®-DEV v1.2 continuous representation.

and use the same model components but are organized in different ways. Chapter 24 elaborates on the staged and continuous representations.

An organization's process improvement initiative, however, is not restricted to using either the staged or continuous representation categorizations of the process areas. An organization can, in fact, categorize the process areas in other ways that support the achievement of their business objectives. For example, Figure 3.3 shows the process areas categorized in a project management–oriented categorization.

The organization may have a strong need to improve its project management activities. This might also be an organization that is not supportive of Quality Assurance and Configuration Management as Quality Management functions, but supports them as Project Management functions. Then organizing Requirements Development and Requirements Management is a requirements concepts category and Project Planning, Project Monitoring and Control, Supplier Agreement Management, Risk Management, Process and Product Quality Assurance, Configuration Management, and Measurement and Analysis as project management functions might be the right approach to implementing these process areas in this organization.

In a similar fashion, Figure 3.4 represents one view of a practical usage categorization. This categorization groups together only strongly connected project management functions, creates a category for quality management, and keeps all of the engineering process areas together, as they are in the continuous representation. If the organization truly supports quality management in this manner, this grouping of process areas could be the most beneficial to successfully implementing the CMMI® process areas in the organization.

Module	Process Areas
Requirements Concepts	Requirements Development Requirements Management
Project Management Concepts	Project Planning Project Monitoring and Control Supplier Agreement Management Risk Management Process and Product Quality Assurance Configuration Management Measurement and Analysis
Process Management Concepts	Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management
Engineering Concepts	Technical Solution Product Integration Verification Validation Decision Analysis and Resolution
Integrated Teaming Concepts	OPD + IPPD (SG 2) IPM + IPPD (SG 3)
Quantitative Management Concepts	Quantitative Project Management Organizational Process Performance
Optimizing Concepts	Causal Analysis and Resolution Organizational Innovation and Deployment

Figure 3.3 Project management–oriented categorization.

Module	Process Areas
Project Management Concepts	Project Planning Project Monitoring and Control Risk Management Integrated Project Management
Quality Management Concepts	Configuration Management Process and Product Quality Assurance Measurement and Analysis Decision Analysis and Resolution
Engineering Concepts	Requirements Development Technical Solution Requirements Management Product Integration Product Verification Validation
Acquisition Concepts	Supplier Agreement Management
Process Management Concepts	Organizational Process Focus Organizational Process Definition Integrated Project Management Organizational Training
Integrated Teaming Concepts	OPD + IPPD (SG 2) IPM + IPPD (SG 3)
Quantitative Management Concepts	Quantitative Project Management Organizational Process Performance Causal Analysis and Resolution Organizational Innovation and Deployment

Figure 3.4 Practical usage categorization.

The CMMI® as the Complete Whole

Although the CMMI® was initially integrated from the CMM® for Software, the Systems Engineering CMM® (EIA-731), and the Integrated Product Development CMM® and has undergone several evolutions to become CMMI®-DEV v. 1.2, which includes systems, hardware, and software engineering practices and examples, there are still weak areas that can and will be improved in the CMMI® over time. And although most of us would like to have only one model to have to learn and be responsible for compliance with, the goal of achieving measurable business objectives must be brought to the surface one more time. There are a number of standards and books available that should be part of any process improvement champion's tool kit. The ones listed here are not being offered as the absolute best or as an exhaustive list, but they have served me well and I offer the list to you for your consideration as well.

- Systems engineering
  - EIA-731—Systems Engineering CMM®
  - EIA-632—Processes for Engineering a System
  - IEEE 1220—IEEE Standard for Application and Management of the Systems Engineering Process
  - ISO/IEC 15288—System Lifecycle Processes

    - *Handbook of Systems Engineering*—Andrew Sage, John Wiley & Sons, 1999
- Software engineering
    - ISO/IEC 12207—Software Lifecycle Processes
- SPICE - ISO 15504—Software Process Improvement and Capability Determination—highly used in the automotive engineering industry
- Quality management
    - ISO 9001:2000—Quality Management System Design
    - AS9100—ISO 9000 for Aerospace
    - TL 9000—ISO 9000 for Telecommunications
    - ISO 16949—Automotive Engineering
    - *Handbook of Software Quality Assurance* (4th ed.)—G. Gordon Schulmeyer, Artech House, 2007
- ITIL—IT Infrastructure Library (focuses on service management and service delivery; major input to the CMMI-SVC v1.2 model that is under development)
- Six Sigma
    - *The Six Sigma Handbook*—Thomas Pyzdek
    - *CMMI® and Six Sigma*—Jeannine Sivy, Lynn Penn, and Robert Stoddard
- Lean manufacturing
- Project Management Institute (PMI) (PMI's PMBOK has gained popularity and OPM3 has extended the project management to the "organizational level" to be more in line with CMMI®)
- Basic statistics (see Chapter 19).

A book I would like to give special mention to is *CMMI® Survival Guide—Just Enough Process Improvement* by Suzanne Garcia, Richard Turner (Addison-Wesley). After reading the book, I was compelled to write a phrase of praise. I found it to be a great reference even for those of us who helped start the industry because it provides clear and useful answers to those tough questions we are asked all of the time.

Other CMMI® support references are available and you are encouraged to pick up a new or different one now and then so you can keep challenging your thinking on how the CMMI® can be better used to support an organization's process improvement initiative.

The CMMI® Is Not an Engineering Handbook

The CMMI® is not an engineering handbook. No matter how badly we would like the CMMI® to answer all of the management, engineering, and business questions we may have, it was not designed for that. Organizations must understand or get training, mentoring, and coaching in the engineering principles behind the CMMI® model. The CMMI®:

- Won't tell you how to conduct requirements analysis or what technique to use or what requirements traceability tool to use;
- Won't tell you how to establish a quality engineering group;
- Won't tell you what configuration management tool to use;
- Won't tell you what your business objectives are;
- Won't tell you to use PMI or Prince2 or any other project management standard or tool;
- Won't tell you how to build your product integration environment or whether you can use it for verification and validation as well;
- Won't set the risk management thresholds for you to trigger the implementation of the risk mitigation plans;
- And so forth.

Sticky Areas

A quick glance at the CMMI® process area purpose statement or introductory notes might give one a false sense of security that the basic concepts found in the CMMI® are known and understood and for the most part practiced throughout the organization. The author and his associates have extensive appraisal experience by virtue of conducting hundreds of appraisals worldwide. It is our experience that there are classic "sticky areas" that have caused many organizations problems and appraisal teams late nights, grief, and frustration. Examples of these "sticky areas" include:

- Project management
  - Size and complexity estimation (LOC, function points, gates on a gate array, number of interfaces, and so on);
  - Having "historical data" to support the estimation;
  - Making estimations based on hours is *not* size estimation;
  - Understanding the importance of the WBS for project management and for supplier management;
  - Commitment process (see Special Goal 3 on commitment and reconciliation of differences between the estimates and the available resources);
  - Stakeholders are not "just the project team members"; they include both internal and external groups such as quality assurance, configuration management, regulatory agencies, customers, end users, independent test, and so forth.
- Configuration management
  - Status accounting (supporting project managers, verification, and evaluation of system completeness)
  - Configuration auditing: baseline auditing, functional configuration audits, and physical configuration audits.
- Quality assurance

- Project quality plan describes how the “quality functions” are going to be handled on the project throughout the project life cycle and by whom. The project quality plan does not contain the details of quality audits, configuration management, testing, peer reviews, and so forth.
- The quality assurance plan is a separate plan from the project quality plan that describes the quality function support the quality assurance representatives or group are going to provide to the project.
- Quality assurance and quality control are different focuses on quality. Quality control is exemplified by peer reviews and testing on life-cycle products. Quality assurance focuses on process and product compliance.

See the Appendix A for a full set of sticky areas descriptions.

Entry Strategies into Process Improvement

One of the more important steps in starting a process improvement initiative is to determine the appropriate tasking and the scope of the process improvement program [1]. There is a great temptation for an organization to attempt to take on too much too fast, especially if it feels that it must catch up to its competition. Although it is natural to want to initiate a program quickly, it is important for an organization trying to get a process improvement initiative started to be as realistic as possible during these beginning stages. It might not be appropriate for one organization to conduct an appraisal straightaway; yet another organization might want to focus on only a few areas to get its process improvement initiative started, show positive results, and then expand.

Many factors must be taken into consideration when an organization is trying to establish its organizational process. The process improvement entry strategy an organization chooses depends on a number of factors. These factors have been critical for organizations that have started their process improvement programs in the past 20 years:

- History of previous process improvement programs or quality improvement programs;
- Financial resources to fund the process improvement initiative;
- Human resources able to be dedicated to process improvement;
- Software engineering capability of the developers;
- Technology support available;
- Contractual obligations;
- Scope;
- Customs and culture of the organization;
- Standards (industry, corporate, organizational, project, customer, environmental);
- Understanding and support from all levels of management and practitioners;
- Corporate political pressure;
- Vision;

- Business objectives;
- Measurement objectives.

Now more than 20 years ago when the Software Engineering Institute, under the direction of Watts Humphrey, was beginning to conduct software process assessments, the path to process improvement was clear:

- Watts Humphrey would conduct an assessment and process improvement briefing with the senior management team of an organization.
- The senior manager would commit to the assessment and to the action planning afterward.
- The organization was invited to the SEI to receive training in the assessment process.
- The assessment was conducted.
- The assessment results were delivered to the senior management team.
- Six weeks later an assessment report was delivered to the organization that contained the assessment results and a set of recommendations.
- The organization was expected to develop an action plan that the SEI would review.
- The organization was expected to implement the action plan and advance to the next level of process maturity.

The very strong assumption implicit in all of this was that the organization had the skills to develop an action plan and implement it. Years later, the process improvement industry is still seeing more and more organizations getting involved with process improvement initiatives now based on the CMMI® that may not be able to succeed on such a process improvement path. The factors that may affect a process improvement initiative must be determined and used to guide the organization into choosing the right entry strategy. One size does not fit all, and an assessment may not even be the right place to start! Note that the entry strategies presented here are not mutually exclusive.

Training in the CMMI®

While those of us heavily involved with CMMI®-based process improvement believe that the CMMI® is a de facto standard in the world, many individuals, departments, business units, and companies are not very familiar with the CMMI®. The senior management team may have been told that their business unit must be a CMMI® Level 2, but there is often a lot of ignorance about what the CMMI® is and what guidance it offers.

One process improvement entry strategy is to provide CMMI® training for all levels of management and practitioners to build a process improvement and CMMI® vocabulary. Once the intent and value of the CMMI® are understood at every level in the organization, its value as a process improvement support tool goes up. Practitioners will know what it can and cannot provide. Practitioners will understand and eventually realize how CMMI® can be used to guide an organization to support the organization's business objectives and management's vision. As a result, the threat of the CMMI® to the management team can be eliminated or at least greatly reduced. The CMMI® training may be carried out in two stages:

- Detailed CMMI® training for EPG members, project managers, and quality management members as a way of introducing the CMMI® as a road map so that the CMMI® concepts can be supported by internal members of the organization. This training should emphasize how the process areas of the CMMI® are interrelated to each other.
- Training in the CMMI® concepts for all levels of management and practitioners as a way of introducing the terminology and helping them to understand what guidance the CMMI® can provide the organization. Such training can help the organization achieve its process improvement goals. It is important for each level of management and the practitioners to have a common CMMI® vocabulary.

Focused Training on the Management and Engineering Principles Behind the CMMI® Model

While the CMMI® serves as an excellent road map that an organization can follow in its quest to achieve product excellence, it is not, as we have previously stated, an engineering handbook. Training in the engineering principles that support the achievement of the management and technical practices of the CMMI® may be the next step for an organization that has had basic CMMI® training, but it also may be an appropriate process improvement entry strategy.

Training in process areas such as Project Management, Requirements Engineering, Quality Engineering, and Configuration Management will provide the management team and select practitioners with a more in-depth understanding of these critical activities. This more detailed understanding of the management and technical practices can lead to a desire to learn how to integrate and apply these practices. This in turn could lead to training in the CMMI® for all levels of management and practitioners to gain the understanding of the CMMI® road map that can help guide the organization in the action planning and implementation of these practices.

Engineering Workshop Training, Action Planning, and Process Mentoring

Training in the engineering principles behind the CMMI® management and technical practices sets the framework for investigation into what the organization is currently doing in those areas and for subsequent improvement. One conclusion is that an organization could conduct an engineering workshop in the management practices and follow this up with an appraisal and action plan. But experience has shown that an appraisal may still not be an appropriate follow-up activity. Detailed training in one or more of the management practices, such as quality assurance, provides a vocabulary, but does not institutionalize it in the project or the organization.

Obtaining the training is not the same as incorporating the practices into the organization, so an appraisal would only serve to show that the concepts were not being used and that they did not perform the management practices well. A more interesting approach might be to combine the engineering workshop training together with a small amount of diagnosis, action planning and implementation, and process mentoring.

In this approach, an organization could first conduct a 3- to 5-day workshop in a particular area, for example quality assurance (QA). Following the workshop, QA group members might examine available documentation and conduct spot interviews to get a better understanding of what documented processes exist and to what level they are being implemented. This small amount of diagnosis is important as a catalyst to build an action plan for improvement in that specific area.

Another alternative that has been used with documented success is to develop a blend of training, diagnosis, and recommendations-action planning. The individual training modules are followed up with spot interviews and recommendations-action planning.

Quick Check

When an organization thinks it should get involved with process improvement, but is not sure just what that is, a Quick Check can provide an understanding of the principles of process improvement and develop a gestalt of what the management and practitioners think are their most pressing problems.

The first day of the Quick Check is dedicated to reviewing the concepts of process improvement through a process improvement seminar. Some of the topics covered include:

- Process improvement model;
- Process management concepts;
- Assessment principles and general flow;
- Guidance for Action Planning;
- Action planning;
- Managing process change;
- Process improvement infrastructure;
- CMMI® overview;
- Approaches of the process improvement leaders;
- Getting process improvement support from above and below;
- Putting process improvement into perspective.

The second day of the Quick Check consists of a series of interviews, set up by the organization, in which the process consultant moves from one group or person to another and brainstorms or serves as a sounding board. This allows the individuals or groups of managers and practitioners to test their understanding of process improvement and suggest their theories in privacy. At the end of the day of interviews, the process consultant usually has a reasonable view of the key process issues facing the organization and can suggest the next step, which normally involves training, action planning, and implementation focused on a few areas.

Workshop Assessment

In a workshop assessment, the senior management team, project leaders, process improvement champions, quality managers, quality engineers, and other interested groups and individuals in process improvement participate in a 2- to 3-day workshop that includes:

- An overview of process management concepts and principles;
- An overview of the CMMI®;
- An understanding of the principles of managing change;
- Completion of a process maturity questionnaire;
- A high level understanding of the organization's process maturity;
- A determination of which process areas to focus on for an initial process improvement effort.

The workshop assessment provides an organization with an overview of its processes as they relate to the CMMI®. During the workshop assessment, the participants receive a broad introduction to the concepts and techniques for improving the organization's processes and gain a high-level understanding of the CMMI®'s process areas. They also characterize their organization's and/or project's process capability and maturity by responding to a key practices questionnaire. The key practices questionnaire that I have used had five possible answers:

1. ❑ Always _______________...
2. ❑ Usually _______________...
3. ❑ Sometimes ____________...
4. ❑ Rarely ________________...
5. ❑ Never ________________...

In addition to the squares to be checked next to the possible answers, the respondents were asked to offer comments especially when their answer was "Always" or "Usually" or even "Sometimes." These comments were to include the respondent's "confidence level." Frequently a respondent would check the "Always" box and then add "Except in the following situations...." The confidence level was an attempt to get the respondent to think about what he/she actually did and answer the questionnaire responsibly.

The first part of the workshop assessment is a process improvement seminar described in the Quick Check section of this chapter. Next, the key practices questionnaire is filled out by selected participants. At the conclusion of the seminar, the data from the filled out questionnaire is placed into a spreadsheet that produces a process area (PA) profile, which is then fed back to the selected participants. During the brainstorming, the confidence level with the resulting profile is discussed. The end result of this discussion is a determination by the participants as to which two or three PAs are the most problematic for the organization.

Although the workshop assessment is not very reliable (no documentation is reviewed and no interviews are conducted outside of the brainstorming with the selected participants), it can provide a focus for the organization. At this point, the organization can decide to engage in a detailed assessment, get training in the areas they feel are the most critical, or simply focus on recommendations and action planning.

Focused Process Area Assessment

Some organizations realize that they are not able to support a process improvement initiative with six to nine target areas that require improvement. There may be concerns about whether the process improvement effort will provide a business benefit or there may be other company constraints that restrict the process improvement initiative. In this situation, an organization may choose to do a detailed assessment, but limit it to two or three focused PAs. Those areas of focus may be determined by a workshop assessment or they may be already known in the organization. The focused PA assessment uses the maturity questionnaires mentioned in the workshop assessment description. It includes interviews with senior management, middle management, project leaders, and practitioners; however, each interview focuses on just the PAs that were chosen for this assessment. It produces results that include:

- *Process capability:* those PA strengths the project or organization is demonstrating in comparison to the activities described in the specific PAs of the CMMI®;
- *Findings:* the weaknesses or process problems that are having a negative effect on the organization's or project's process capabilities with regard to that PA;
- *Consequences:* the perceived, derived, or real risks that are associated with having those weaknesses in that PA;
- *Recommendations:* a description of how the process problems might be solved to result in process and product quality improvement.

The third part of the focused PA assessment is the Guidance for Action Planning (GAP). The output of the GAP indicates to senior management what support must be provided to get action going in the particular PA. It also provides guidance for the EPG and working group(s) that will be tasked with fully developing the action plan for improvement in a particular process area.

The advantage of the focused PA assessment approach is that it allows the organization to get started on action planning and implementation right away. The price of a focused PA assessment is significantly less than that for a full assessment because of the reduced time necessary to complete the diagnosis. Also, the demand on the organization's resources is reduced. If the process improvement effort is successful focusing on the chosen two to three areas, then it is probable that the organization will expand into the next set of PAs.

SEI SCAMPI℠ Appraisals

Of course, an organization may feel that it is in a comfortable enough position to ask for an SEI SCAMPI℠ B or A appraisal. The scope of the SCAMPI℠ appraisal should be thoroughly discussed with the senior management team, the process improvement champions, and the lead appraiser.

Establishing and Maintaining an Action Plan

It is one thing to have an assessment conducted to get your process improvement initiative going, but it is quite another to develop a complete action plan that looks and acts just like a project plan for the process improvement initiative. An organizational action plan may be composed of many chapters that concentrate on one or more process areas or even a subset of a process area we might call a focus area.

Typical contents of a process action plan include:

- Improvements that will be covered;
- Procedures for planning and tracking progress for each process action;
- Strategies for implementing the process actions including the identification and selection of pilot projects;
- Training, mentoring, and coaching needed for the pilot project members;
- Responsibility and authority for implementing the process actions;
- Resources, schedules, and assignments for implementing the process actions;
- Evaluation criteria for successful implementation of the process actions;
- Methods for determining the effectiveness of the process actions;
- Risks associated with the process action plans;
- Progress reviews with senior management and the steering committee.

I should point out that even with multiple and difficult business constraints, senior management teams frequently promise their support for improvement in up to 18 process areas without having a sound idea of what that commitment will actually mean for all of the levels of management, the process improvement champions, the quality assurance group or in terms of tools, training, and funding that may be needed. The impact to critical projects is often ignored. A frequent statement made is "We know that we are under tremendous pressure right now but we have to make this process improvement initiative happen and reach CMMI® ML 3!" It is great to have that type of spirit from the senior management team, but it is often impractical if not impossible. A workforce that is already working 125% of the time is not always too interested in adding another 25% to 50% to their load for the sake of CMMI®-based process improvement. And when reality sets in and project management gets pressured for delivery dates, process improvement activities are often the ones that get ignored.

It has been my experience that it is better for the senior management team to promise to support improvement in one process area and support that one 150% than it is to promise improvement in most of the process areas of ML 2 and ML 3 and fail to accomplish any of them.

Process Improvement for Systems, Hardware, Software, and Business Based on the CMMI®

Why base your organization's process improvement success on the CMMI®? This indeed may be the question on your mind if you are thinking about all of the models, standards, and methods that have been produced so far and are still in existence.

First and foremost, the emphasis of the CMMI® is to support the development processes and changing of cultures to show a measurable benefit for the organization's business objectives and vision. The CMMI® provides a framework from which to organize and prioritize engineering, people, and business activities. It supports the coordination of multidisciplined activities that are or may be required to successfully build a project. As the CMMI® is used as the basis for improving an organization's software, hardware, and systems processes, it is being discovered that these processes can be translated into other organizational departments such as human resources, finance, marketing, computer services, and contract management.

The CMMI® provides the basis for an organization to develop and control its own project management and engineering processes so that it can, in turn, manage the results of its suppliers' processes. It ensures identification and control of an organization's core competencies and it enables an organization to competitively "posture" itself in today's fast changing world.

A frequently overlooked advantage of the CMMI® is that the CMMI® captures lessons learned from the use of the CMM® for Software and other models, methods, and standards during the past 15 years. These lessons learned can be found in many of the CMMI®-DEV v1.2's process areas including:

- Engineering process areas of Requirements Development, Technical Solution, Requirements Management, and Product Integration, Verification, and Validation;
- Decision Analysis and Resolution;
- Project Planning and Project Monitoring and Control;
- Integrated Project Management;
- Supplier Agreement Management;
- Measurement and Analysis;
- Risk Management;
- Integrated Product and Process Development;
- Organizational Process Focus;
- Organizational Process Definition;
- Organizational Process Performance;
- Quantitative Project Management;
- Organizational Innovation and Deployment.

The CMMI® supports:

- Systems engineering;
- Software engineering;
- Hardware engineering:
  - Electrical engineering;
  - Mechanical engineering;
  - Optical engineering;
  - Electro-optical;

    - Electromagnetics;
    - Electromechanical;
    - Hydraulics.

- Manufacturing;
- “Glass factories”;
- Embedded systems;
- Information systems;
- Information technology,
- Banks and insurance companies;
- Medical systems;
- And so forth.

The SEI supports the CMMI® worldwide.

The CMMI® and Engineering Systems Thinking

Because the CMMI® has integrated the concepts of the Systems Engineering CMM® and the CMM® for Software, it returns the concept of “engineering systems think” to projects and organizations where it has been sorely lacking for more than a decade.

The laws of engineering systems thinking, presented in Chapter 1 can be encapsulated in a few key ideas here:

- In all of the project’s phases/stages, and along the system’s life, the systems engineer has to take into account:
    - The customer’s organizational vision, goals, and tasks;
    - The customer’s requirements and preferences;
    - The problem to be solved by the system and the customer’s needs.
- The whole has to be seen as well as the interaction between the system’s elements, that is, iterative or recursive thinking must replace the traditional linear thinking.
- The solution is not always an engineering one. We must remember to always take into account:
    - Business and economic costs;
    - Reuse or utilization of products and infrastructure already developed;
    - Organizational, managerial, political, and personal considerations.

Consider these benefits of a CMMI®-based process improvement initiative:

- Increased control of costs and ability to predict development cycle length and costs of multidisciplined product and product components;
- The ability to remove defects early and efficiently from the life-cycle work products;

- Reduced rework leading to reduced development cycle time;
- Increased predictability and control of product quality;
- Enhanced ability to make cost–benefit trade-offs of development methodologies, technologies, and processes;
- Increased capability to select and manage qualified suppliers;
- Enhanced ability to make risk management decisions based on quantitative data;
- More time available for top innovators to spend on problems and challenges requiring creative energy;
- Knowledge retention and expansion;
- Satisfied customers.

It is the CMMI® and only the CMMI® that has successfully combined the tried and true ideas presented by the CMM® for Software and embraced engineering systems thinking at the same time.

Summary

The CMMI® represents integration of multiple military, ISO, IEEE, and commercial standards and procedures that cover all aspects of building systems. It is closely linked to ISO 9001:2000. The emphasis of the CMMI® is to support the development processes and changing of cultures to show a measurable benefit for the organization's business objectives and vision. The CMMI® provides a framework from which to organize and prioritize engineering, people, and business activities. It supports the coordination of multidisciplined activities that are or may be required to successfully build a project and it returns the concept of "engineering systems think" to project development.

Process defines how a business does business. It helps to establish the business culture and then sets guidelines and expectations. The CMMI® provides detailed practice descriptions to guide the development of technical processes as well as management processes and service processes. However, as previously stated, the main focus needs to be on the organization's vision and business objectives. The CMMI® is not about process descriptions, but about supporting measurable business objectives.

The CMMI® provides a framework from which to organize and prioritize engineering, people, and business activities. Regardless of what process improvement approach your organization starts with or evolves too, the CMMI® framework can yield the necessary guidance for your organization's success.

Reference

[1] Kasse, T. C., and P. A. McQuaid, "Entry Strategies into the Process Improvement Initiative," *Software Process: Improvement and Practice,* Vol. 4, 1998, pp. 73–88.

CHAPTER 4

The Language of the CMMI®

Over the years, many people have suggested that the CMMI® was written in an artificial language. This chapter provides the reader with a better understanding of some of the more critical vocabulary that is used throughout the CMMI®. The CMMI® language does require effort to understand and apply its practices throughout an organization. Although an organization is certainly not required to strictly adopt CMMI® terms in order to show compliance with its principles and guidance, some knowledge of the most important terms starts the journey of getting the look and feel of what it would be like to implement the CMMI® concepts in an organization.

According to the introductory words found in the glossary of the *CMMI® Second Edition, Guidelines for Process Integration and Product Improvement* [1], the CMMI® glossary defines the basic terms used in the CMM® models. Multiple sources were consulted by the SEI authors, including *Webster's* online dictionary and the original source models (i.e., EIA-731; SW-CMM® v2.0, draft C; and IPD-CMM® v0.98). Other standards such as DoD-Std-2167a were consulted as needed. The list of these standards can be found in the CMMI® v1.2 glossary.

Model

CMMI® (Capability Maturity Model® Integration) is a process improvement maturity model for the development of products and services. A model is not a process. A model is a structured collection of elements that describes characteristics of effective processes. A model is used to help set process improvement objectives and priorities and to improve processes. A model is used to help ensure that the processes we put in place throughout our organizations will be stable, capable, and mature. A model is intended to be used as a guide for improvement of project and organizational processes. A model provides:

- A place to start;
- The benefit of an industry's prior experiences;
- A common language;
- A shared vision;
- A framework for prioritizing actions;

- A reminder of the project management, quality management, and engineering activities that should be considered and normally implemented on projects;
- A way to define what *improvement* means for the organization;
- A guide to help an organization with the way it does business.

Model Options

The CMMI® v1.1 offered a number of model options. They included:

- Systems Engineering + Software Engineering + Integrated Product and Process Development (CMMI®-SE/SW/IPPD);
- Systems Engineering + Software Engineering + Integrated Product and Process Development + Supplier Sourcing (CMMI®-SE/SW/IPPD/SS).

The CMMI®-DEV v1.2 by virtue of the CMMI® Product Suite, CMMI® Framework, and the concept of CMMI® "constellations," all defined later, is an *application-specific model* whose purpose is to help organizations improve their development and maintenance processes for both products and services.

There are two other models:

- CMMI®-ACQ: Provides guidance to enable informed and decisive acquisition leadership. This model describes the practices that support the process of obtaining products and services through contract.
- CMMI®-SVC: Provides guidance for those providing services within organizations and to external customers. This model will include service deployment practices.

CMMI® Product Suite

The CMMI® Product Suite is the complete set of products developed around the CMMI® concept:

- Framework itself;
- Models;
- Appraisal methods;
- Appraisal materials;
- Various types of training produced from the CMMI® Framework.

CMMI® Framework

The CMMI® Framework is the basic structure that organizes CMMI® components, including common elements as well as rules and methods for generating mod-

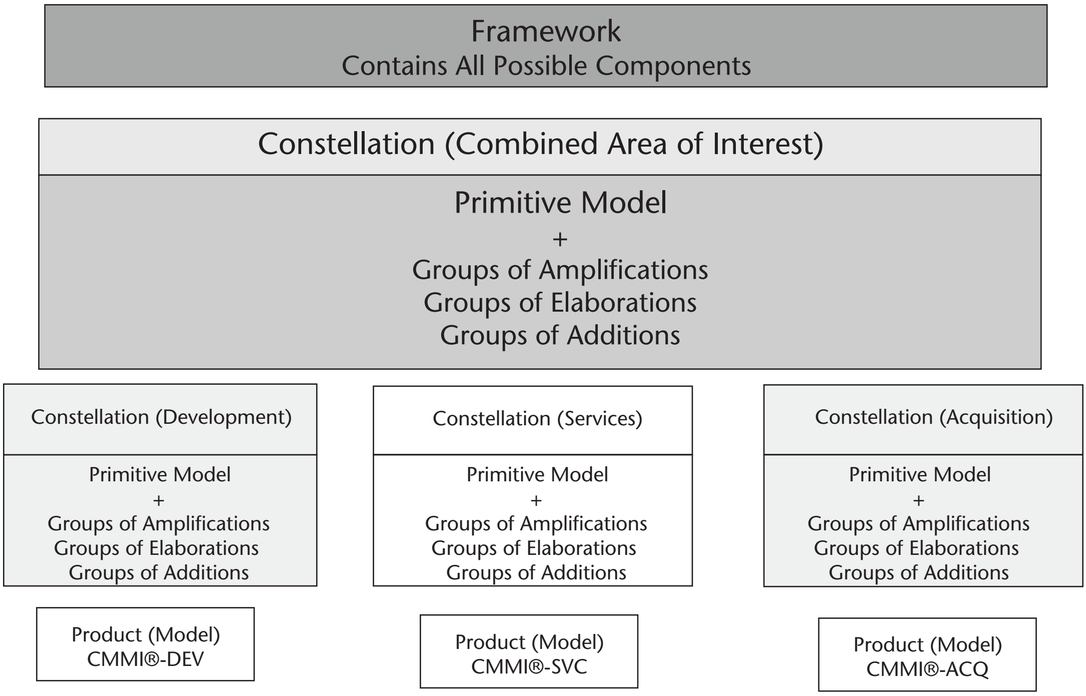


Figure 4.1 CMMI® v1.2 architectural framework (*From:* [2]. © 2005 SEI. Reprinted with permission.)

els, their appraisal methods (including associated artifacts), and their training materials. See Figure 4.1.

CMMI® Architecture

The CMMI® architecture is used to create the CMMI® models and their associated training and appraisal materials for nondevelopmental efforts such as CMMI® for Services and CMMI® for Acquisition.

Constellation

A constellation is a collection of CMMI® components in the CMMI® Framework that are organized into groupings and include a model, its training materials, and appraisal-related documents for areas of interest. Each constellation is composed of a "model foundation" or core components that cannot be deleted or changed, including:

- Selected sections of front matter;
- All generic goals and generic practices;
- Core process areas;
- Core glossary.

The core process areas include the introductory notes and the notes appropriate to all constellations; however, they do not include additions, generic practice elaborations, or amplifications. “Shared” materials are those materials that are in two or more CMMI® constellations. “Specific” materials are unique to the model in a specific constellation.

Disciplines

The term *discipline* usually refers to a structured and systematic approach to a task, such as an approach to training for a sport or for an engineering process to solve problems. If an athlete works hard and seems to be able to ignore distractions, we say he/she is disciplined. The CMMI® uses the term *discipline* to refer to the engineering disciplines an organization or project may require. It refers to a body of knowledge, developed skill sets, and a sequence of processes that are related to product or process development. When we read about disciplines in the CMMI®, we typically think of software engineering, systems engineering, electrical engineering, mechanical engineering, optics or optical engineering, electro-optical engineering, manufacturing, and so forth. Systems requirements may be allocated to be implemented in one or more of these specific disciplines.

Addition

An addition is a clearly marked model component that contains information of interest to particular users. It does not refer to a particular engineering discipline. That would be defined by the CMMI® glossary term of *amplification*. In the CMMI® model, all additions having the same name (i.e., IPPD Addition) may be optionally selected as a group to use.

Adequate, Appropriate, as Needed

The terms *adequate, appropriate,* and *as needed* appear in the CMMI® to allow managers at all levels and practitioners to interpret the specific goals and generic goals and the specific practices and generic practices in light of the organization’s business objectives. These words do not necessarily refer to the “minimum.” *Adequate,* for example, may refer to an entire quality assurance organization, not just one person who holds a Quality Assurance Responsible title.

Generic Practice 2.3 for the process area of Risk Management states “Provide adequate resources for performing the risk management process, developing the work products, and providing the services of the process.” In this case, the term *adequate* could be satisfied by:

- Numbers of people;
- Skills of the people carrying out the risk management process;
- The extent to which the risk mitigation activities are defined;

- Any tools that might be used to support the risk mitigation activities;
- People who must monitor the risks;
- People who ensure that the Top 10 Risk List is current.

Establish and Maintain

The phrase *establish and maintain* includes making sure a process is used as well as documented. This should include proof such as minutes of meetings, audit reports, peer review data, measurement data, and proof of use. For example, "Establish and maintain the strategy to be used for risk management." The risk management strategy must be developed and contain content such as:

- Scope of the risk management effort;
- Methods and tools to be used for risk identification, risk analysis, risk mitigation, risk monitoring, and communication;
- How these risks are to be organized, categorized, compared, and consolidated;
- Risk mitigation techniques to be used, such as prototyping, simulation, alternative designs, or evolutionary development.

The procedures and templates used to develop a risk management strategy should be documented, trained, and be able to be coached. Each project would have its risk management strategy defined along with a risk management plan. A risk management plan may be incorporated into a project management plan as a section. It does not need to be a stand-alone document. The project monitoring and control activities would show that the risk management strategy is followed if risks cross established risk thresholds and/or become problems.

In addition, the phrase *establish and maintain* means that a plan or process description is updated as needed. This might happen when the monitoring and control activities show that the actuals tracked against the estimates indicate significant deviations and, hence, corrective action must be taken.

Customer

A customer is the individual, project, organization, group, and so on, that is responsible for accepting the product or for authorizing payment. The customer is external to the project but not necessarily external to the organization. Customers are a subset of the stakeholders, as Figure 4.2 shows.

The customer requirements represent the stakeholder needs (see stakeholder definition later in the chapter), expectations, constraints, and interfaces that have been collected, consolidated, resolved, and translated into customer requirements. These customer requirements will evolve into technical, product, or product component requirements.

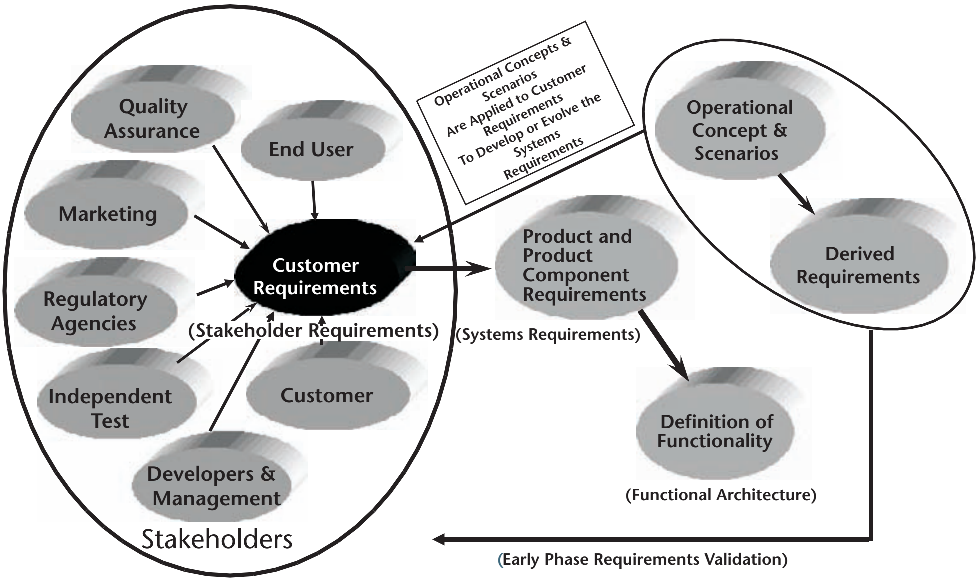


Figure 4.2 Customer, product, and product component requirements.

Policy

A policy or an organizational policy as used in the CMMI® is defined as a guiding principle. It is typically established by senior management and adopted by the organization to influence and determine decisions. The organizational policy defines the organizational expectations for the process and makes these expectations visible to those in the organization who are affected.

Another related view of policy is that it is supposed to be a "behavior expectation setting document." The policy should describe how an organization's employees should go about doing their daily work. The policy should describe how the employees are supposed to behave or act and what values they are supposed to live each working day. For example, a policy on peer reviews may state that all projects must plan and carry out peer reviews on critical product components or subsystems or even modules. If an evaluation were to be carried out by an independent evaluator, he/she could read the policy on peer reviews and then interview project managers and project members to find out if they have identified critical product components and if they have conducted or plan to conduct peer reviews on the life-cycle work products defining those product components. Finding out how the peer reviews are conducted, what type of training reviewers receive, and what analysis is conducted on defects that are found would be an issue for determining the process of conducting the reviews. The policy, if implemented, would ensure that the peer reviews were conducted.

Stakeholder

A stakeholder is a group or individual who is affected by the outcome of a project or can affect the activities or output of the project. Stakeholders can include:

- Customers;
- End users;
- Developers;
- Producers;
- Testers;
- Quality assurance;
- Database;
- Configuration management;
- Suppliers;
- Marketers;
- Maintainers;
- Safety regulation agencies;
- Managers.

Relevant Stakeholder

The term *relevant stakeholder* is used to designate a stakeholder who has been identified for involvement in specified activities and is included in an appropriate plan such as the project plan. The chief financial officer (CFO) could be a relevant stakeholder for a project that is operating under a fixed-price budget and the project is critical to the organization's business success. The CFO might be involved at the beginning of the project when the budget is first established and before the contract is signed. He/she might be involved at milestone reviews to ensure that the project is progressing within the budget constraints. Marketing would be a relevant stakeholder during the customer and systems or product engineering requirements elicitation phase, but would normally not be a relevant stakeholder during the design and development phase.

Project Manager

A project manager is the person responsible for planning, directing, controlling, structuring, and motivating the project. He/she is ultimately responsible for timely delivery, schedule, resources, and product quality. He/she may provide both technical and administrative direction and control to those performing project tasks or activities within his/her area of responsibility. The project manager is ultimately responsible to the customer. The project manager takes on different roles and responsibilities as the size, diversity, and complexity of the project changes:

- Small project;
- Large integrated project consisting of systems, software, mechanical, electrical and plastics engineering components.

Higher Level Management

Higher level management refers to the person or persons who provide the policy and overall guidance for the process, but do not provide the day-to-day monitoring and control of the process. In particular, higher level management includes senior management according to the definition of GP 2.10, Review Status with Higher Level Management [1]: "Review the activities, status, and results of the process with higher level management and resolve issues. The purpose of this generic practice is to provide higher level management with the appropriate visibility into the process. Higher level management includes those levels of management in the organization above the immediate level of management responsible for the process. In particular, higher level management includes senior management. These reviews are for managers who provide the policy and overall guidance for the process, and not for those who perform the direct day-to-day monitoring and controlling of the process."

Senior Manager

The term *senior manager* as it is used in the CMMI® refers to a management role at a high enough level in an organization that the primary focus of the person is the long-term health and success of the organization rather than the short-term project and contractual concerns and pressures. A senior manager may be responsible for the oversight of a program that may contain many projects that are managed by project managers. He must be aware of the critical dependencies between projects. The senior manager normally establishes the business objectives and ensures that his/her vision is presented to and understood by all levels of management and practitioners. A senior manager has the authority to direct the allocation or reallocation of resources in support of the organization's process improvement initiative. It is also typical for the senior manager to have profit and loss responsibility. Synonyms for *senior manager* include *executive* and *top-level manager* and the role may even be filled by the CEO of the organization.

Organization

An organization is a structure in which people collectively manage one or more projects as a whole and whose projects share a senior manager and operate under the same policies. An organization is:

- A unit within an enterprise or company, agency, or service;
- Most often within a single geographical site;
- Increasingly defined over multiple sites, multiple countries, and multiple cultures;
- Normally self-contained.

An organization operates within a defined set of business objectives and according to the senior manager's vision. An organization can also be a major product line within a business unit. Some attributes of organizations include:

- Common management;
- Common business focus;
- Desire to have a common process improvement focus;
- Profit and loss requirements.

Enterprise

The term *enterprise* is used to refer to very large companies that consist of many organizations in many different locations with different customers.

Development

Development, as it is used throughout the CMMI®, implies maintenance activities as well as development activities. Experience has shown that best practices should be applied to both development and maintenance projects if an organization is in pursuit of engineering excellence.

The introductory notes for the Requirements Development PA illustrate this important concept [1]: "All projects have requirements. In the case of a project that is focused on maintenance activities, the changes to the product or product components are based on changes to the existing requirements, design, or implementation. . . . Regardless of their source or form, the maintenance activities that are driven by changes to requirements are managed accordingly."

Product

A product may be thought of as any tangible output or service that is the result of following a process and is intended for delivery to a customer or end user. A product can also be any work product that is delivered to the customer according to contract.

Product Component

Product components are generally the lower level components of a product and are integrated to "build" the product. Product components may be a part of the product delivered to the customer or serve in the manufacture or use of the product. For example, for those companies who manufacture mobile phone batteries, the mobile phone battery is a product. For those companies that build and deliver mobile phones, the battery is a product component.

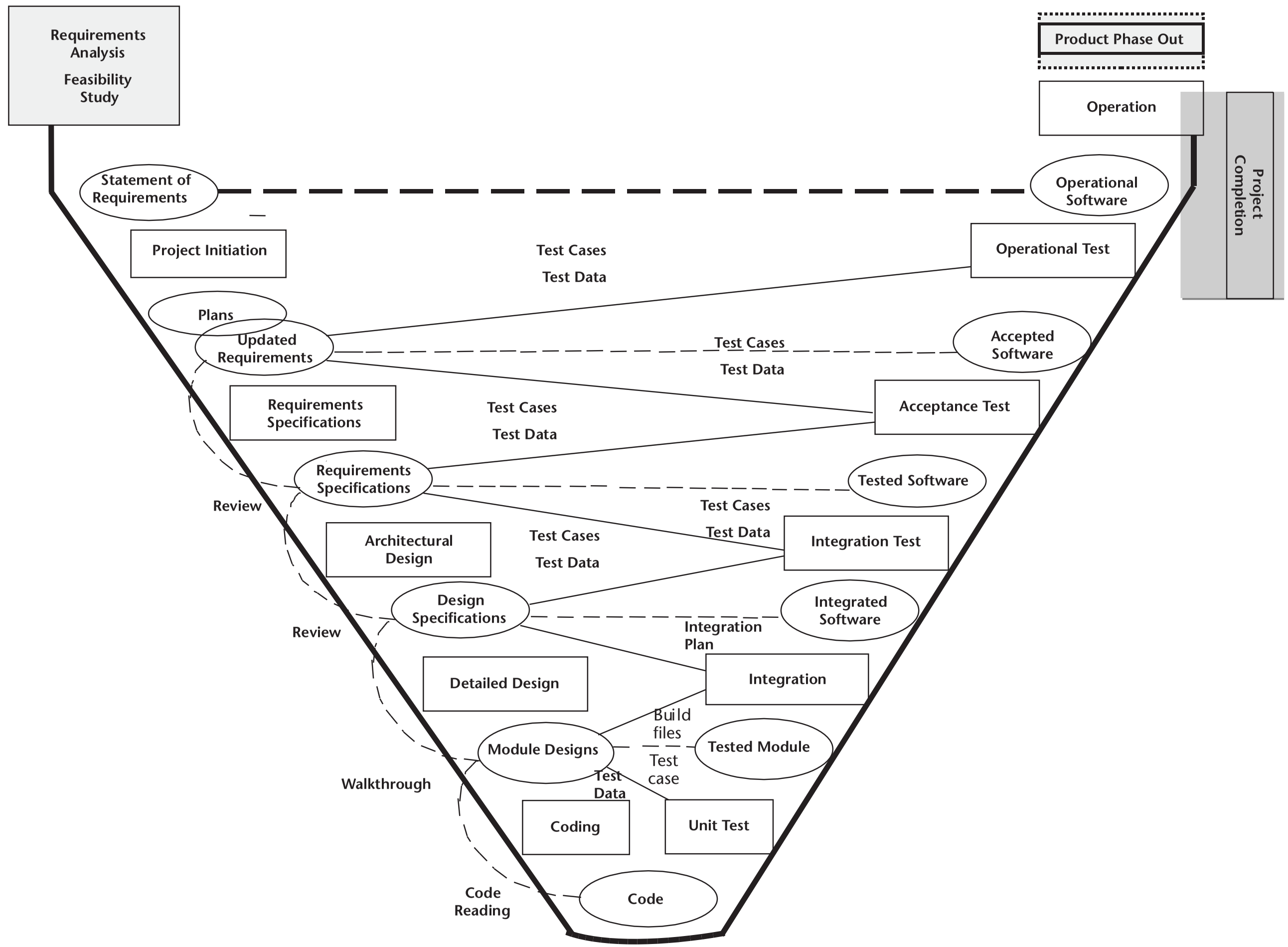


Figure 4.3 V-model development life cycle.

Work Product (Life-Cycle Work Product)

A work product is any artifact produced by a life-cycle process (Figure 4.3) that is accepted as a "useful result" from following the process. It can also be referred to as a life-cycle work product. Life-cycle work products can include:

- Requirements specifications;
- Interface specifications;
- Architecture specifications;
- Design documents;
- Code for software,
- Unit test plans;
- Integration and system test plans;
- A process such as a manufacturing product assembly process;
- Project plans;
- Risk management plans;
- Quality assurance plans;
- Configuration management plans.

Nondevelopmental Item (NDI)

A nondevelopmental item is an item of supply that was developed prior to its current use in an acquisition or development process. The item may require minor modifications to meet the requirements of its current intended use.

Commercial Off-the-Shelf (COTS)

Commercial off-the-shelf or COTS refers to an item that can be purchased from a commercial vendor.

Project

A project is a managed set of interrelated resources that delivers one or more products to a customer or end user. The set of resources has a definite beginning (project startup) and a definite end and operates according to a plan. Such a plan typically specifies:

- Product to be delivered or implemented;
- Resources and funds used;
- Work to be done;
- Schedule for doing the work.

Note that a project can be composed of a set of projects.

Project Startup

Project startup refers to the time when a set of interrelated resources is directed to develop or deliver one or more products for a customer or end user.

Product Life Cycle

A product life cycle is the period of time, consisting of phases, that begins when a product is conceived and ends when the product is no longer available for use. Organizations may define a set of approved product life-cycle models. See the description of Organizational Process Definition (OPD) in the CMMI®-DEV v1.2. See Figure 4.4 for an example of a systems engineering life cycle.

Tailoring Guidelines

Tailoring a process makes, alters, or adapts process descriptions, normally described at the organizational level, for use on a particular project. For most organizations, one organizational process definition cannot or will not be followed 100% for all of the projects. Some adaptation is normally needed. Tailoring guidelines, then, describe what can and cannot be modified and identify process components that are allowable candidates for modification. Tailoring guidelines can cover:

- Selecting a standard process;
- Selecting an approved product life cycle;

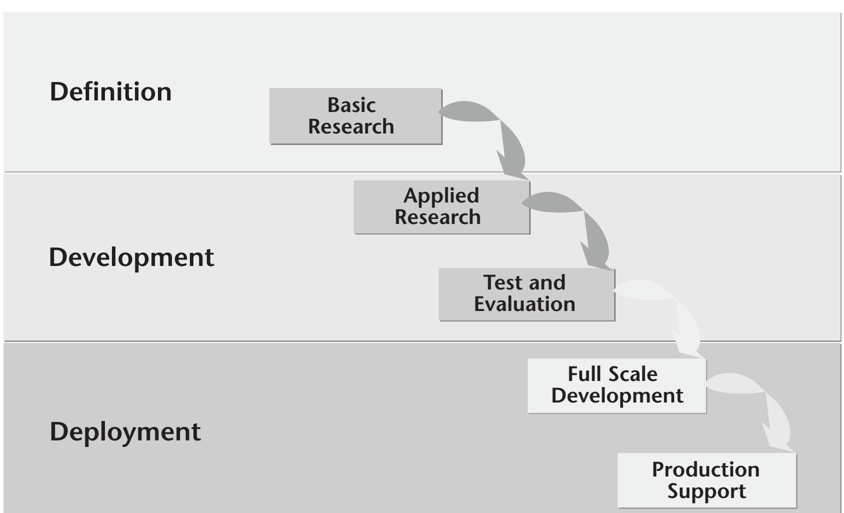


Figure 4.4 Research, development, test, and evaluation life-cycle model.

- Tailoring the selected standard process and life cycle to fit the project needs.

Appraisal

An appraisal is an examination of one or more processes by a trained team of professionals using an appraisal reference model as the basis for determining strengths and weaknesses.

Assessment

An assessment is an appraisal that an organization conducts for itself for the purposes of process improvement.

Capability Evaluation

A capability evaluation is an appraisal by a trained team of professionals that is used as a discriminator to select suppliers, to monitor suppliers, or to determine and enforce incentives. Capability evaluations are most useful to gain insight into the process capability of a supplier organization and they:

- Help decision makers;
- Improve subcontractor performance;
- Provide insight to a purchasing organization.

Operational Concept and Operational Environment

The term *operational concept* refers to a general description of how the system is envisioned to operate, where in the operating environment the system will reside or be distributed, how long the system must operate, and how effective the system's performance must be. Operational concepts evolve during the requirements engineering and development activities to facilitate the selection of product-component solutions that, when implemented, will satisfy the intended use of the product.

Operational Scenarios

An operational scenario is a sequence of events that might occur that includes the interaction of the product with its environment and users, as well as the interaction among its product components. Operational scenarios are often created by defining a possible state and ask "what if" questions. Operational scenarios are used in eliciting requirements from stakeholders (end users, management, regulatory agencies, testing, and so on) as a way to help the customer to better understand their requirements and understand what it will really take to implement those requirements.

Operational scenarios are used effectively to understand and refine requirements elicited from the customers.

Verification

Verification includes verification of the product and intermediate work products against all selected requirements, including customer, product, and product component requirements. Verification is inherently an incremental process. It begins with the verification of the requirements, progresses through the verification of the evolving work products, and culminates in the verification of the completed product.

Verification addresses whether the work product properly reflects the specified requirements. Verification assures "you built it right."

Validation

Validation demonstrates that the product, as provided (or as it will be provided), will fulfill its intended use in the operational environment. Validation assures "you built the right thing."

Document

A document is a collection of data, regardless of the medium on which it is recorded. It generally has permanence and can be read by humans or machines. Documents include both paper and electronic documents.

Data

Data refers to recorded information, regardless of the form or method of recording including:

- Technical data;
- Computer software documents;
- Financial information;
- Management information;
- Administration;
- Engineering;
- Configuration management;
- Logistics;
- Quality;
- Safety;
- Manufacturing;

- Procurement;
- Data of any nature that can be communicated, stored, and processed.

Data or project data may also refer to:

- Reports;
- Manuals;
- Notebooks;
- Charts;
- Drawings;
- Specifications;
- Electronic files or paper files;
- E-mail;
- Correspondence.

Data may be deliverable or nondeliverable such as:

- Informal data;
- Trade studies and analyses;
- Internal meeting minutes;
- Internal design review documentation;
- Lessons learned;
- Action items.

Data Management

Data management is the disciplined process of planning, acquiring, and controlling business and technical data, consistent with data requirements.

Quality and Process Performance Objectives

The phrase *quality and process performance objectives* covers objectives and requirements for:

- Product quality;
- Service quality;
- Process performance.

Process capability is defined to be the range of expected results that can be achieved by following a process. The maturity of an organization as defined by the staged representation of the CMMI® was supposed to be used only as an indicator of that process capability.

If an organization was assessed at CMMI® Maturity Level 2, a project of similar size, complexity, and application domain should be able to be contracted to an organization and the customer should be able to expect similar results in terms of product and service quality and process performance.

Process performance is a measure of the actual results achieved by following a process. It is characterized by both process measures (e.g., effort, cycle time, and defect removal efficiency) and product measures (e.g., reliability, defect density, and response time).

Hardware Engineering

Hardware engineering can be defined as the application of a systematic, disciplined, and quantifiable approach to transform a set of requirements using documented techniques and technology to design, implement, and maintain a tangible product. Hardware engineering in the CMMI® represents all technical fields:

- Electrical engineering;
- Mechanical engineering;
- Optics;
- Electro optics;
- Electromechanical;
- Hydraulics.

Systems Engineering

The CMMI® defines systems engineering as the interdisciplinary approach governing the total technical and managerial effort required to transform a set of customer needs, expectations, and constraints into a product solution and support that solution throughout the product's life. This definition includes the definition of technical performance measures, the integration of engineering specialties toward the establishment of a product architecture, and the definition of supporting life-cycle processes that balance cost, performance, and schedule objectives. Systems engineering is defined by INCOSE as "…a branch of engineering whose responsibility is creating and executing an interdisciplinary process to ensure that customer and stakeholder's needs are satisfied in a high quality, trustworthy, cost efficient and schedule compliant manner throughout a system's entire life cycle, from development to operation to disposal. … The systems engineering process is not sequential: the tasks are performed in a parallel and iterative manner…" [3]. Systems engineering focuses on defining customer needs and required functionality early in the development life cycle, eliciting requirements, conducting early phase validation with the customer, creating alternative system design concepts, conducting trade-off analyses, and selecting and implementing the best life-cycle balanced solution, while considering the complete problem from cradle to grave.

Summary

One of the important benefits to an organization that chooses the CMMI® is the development of a consistent engineering, quality management, and project management vocabulary across its business units. This chapter was written to provide the reader with some of the more critical vocabulary that is used throughout the CMMI®. The reader is encouraged to thoroughly study the CMMI®-DEV v1.2 either through the SEI technical report titled *CMMI® for Development, Version 1.2* (CMU-SEI-2006-TR-008), obtainable as a download from the SEI Web site, or the book titled *CMMI® Second Edition, Guidelines for Process Integration and Product Development.*

References

[1] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI® Second Edition, Guidelines for Process Integration and Product Improvement*, Reading, MA: Addison-Wesley, 2007.

[2] Philips, M., “CMMI® v1.2 and Beyond,” PowerPoint presentation, Software Engineering Institute, Carnegie Mellon Institute, December 13, 2005.

[3] International Council on Systems Engineering, Communications Committee, “What Is Systems Engineering?” February 2008, http://www.incose.org/practice/whatissystemseng.aspx.

CHAPTER 5

Roles and Responsibilities

This chapter provides the reader with a short description of the roles and responsibilities of the various levels of management and practitioners:

- Senior management;
- Higher level management;
- Middle management;
- Project managers;
- Practitioners;
- Process group;
- Quality assurance;
- Configuration management;
- Integration and systems testing;
- Measurement team;
- Systems engineering.

Descriptions of the various roles and responsibilities can be found embedded in various places throughout the CMMI® including the front material, the process areas, and the generic practices. These ideas will be combined with the author's own ideas and interpretations to provide the widest and most flexible but usable definitions.

Senior Management

Establish Policies

The term *senior manager* was defined in Chapter 4 as a management role at a high enough level in an organization that the primary focus of the senior manager is the long-term health and success of the organization as opposed to short-term project and contractual concerns and pressures.

The senior manager establishes the business objectives and ensures that his/her vision is presented to and understood by all levels of management and practitioners. Along with the authority to establish the business objectives and to share his/her vision with the organization, comes the responsibility of caring for the profit and loss of the organization.

Senior management is responsible for defining policy in line with the vision and business objectives. A policy, as previously defined, is a “behavior expectation setting document” that describes how the organization’s employees should go about their daily work. It makes the senior manager’s expectations visible and provides high-level guidance for the development of more detailed processes, procedures, guidelines, templates, and checklists.

Allocate or Reallocate Resources

A senior manager has the responsibility and authority to direct the allocation or reallocation of resources in support of the organization’s process improvement initiative. This can involve the allocation of people, computers, tools, environments, telecommunication equipment, production environment, other tools and techniques, funding, training, and even physical space to groups or activities such as:

- Systems/software engineering process group (SEPG);
- Quality assurance group;
- Configuration management group that utilizes a CM tool product suite such as SERENA Software, Inc., or Telelogic Synergy™ and IBM® Rational® ClearCase®;
- Integration and systems test group;
- Measurement group;
- Development;
- Production;
- Program management;
- Logistics;
- Supply management.

Establish Authority and Responsibility

Project managers and other responsible persons and groups must not only have the responsibility for carrying out a process such as requirements engineering, but must also have the authority to do so. It is quite common to hear that project managers have the responsibility to lead their project to successful completion within the given budget and time constraints without being given the authority to make decisions that may even involve replanning and renegotiation with the customer. Projects that are designated as critical are often not only watched closely by higher level management, but are virtually run by them. Senior managers must take greater time to choose their project managers carefully and to train, mentor, and coach them, and then give them the actual responsibility that they need in order to successfully manage and control their projects.

Authorize Training

Training budgets and training capabilities need the visible support of the senior manager regardless of whether it is to provide training at the organizational level in order to support the organization's strategic business direction, the set of organizational processes, or to address common project needs. Training may take the form of in-house training, e-learning through local or online universities, vendor-provided training, or on-the-job training. On-the-job training should never be classified as a training option by itself, but should always be presented together with coaching. Frequently, human resource departments are tasked with the establishment of an organization-wide training program. Any endeavor at this level requires the support of senior management.

Provide Visible Support

The activities that we have discussed so far are part of the set of what is required of senior management to provide necessary visible management support. Other senior management activities include:

- Ensuring effective bidirectional communication from senior management through the various levels of management to the developers and support groups;
- Developing or overseeing the development of management and technical policies;
- Establishing a SEPG if one does not exist, and also a SEPG chairman role;
- Setting up a QA program at the organizational and project level;
- Ensuring that the configuration management function is established and operating on all projects;
- Ensuring that the necessary funding and sufficient time are allocated to support the defined process improvement activities;
- Ensuring that necessary training, mentoring, and coaching is provided to support the process improvement initiative throughout the organization;
- Ensuring that all other resources that are required to support the process improvement initiative are available;
- Participating in the review of all commitments that are being made on behalf of the organization to ensure that all commitments that have been made before and including this one can be honored;
- Making sure that all levels of management and practitioners understand the vision and business objectives of the organization;
- Letting the organization know on a continuous basis the importance of the process improvement initiative from the senior management point of view.

It is important to add that simply sending out occasional e-mails or conducting a once-per-year general assembly speech espousing care for process improvement and quality is not sufficient. Visible management support means "walking the talk" in the activities in which the senior manager participates in order to show by example. This means ensuring that a discussion on processes and quality is always a part of the monthly senior management progress meeting in addition to the standard focus on costs, schedules, and risk.

Approve Organizational Commitments

Watts Humphrey established the commitment process and senior management oversight review meetings as the pillars of the CMM® and CMMI®. Misunderstandings often arise about "indicating commitment" for process improvement on the part of senior management and the "commitment process." A high-level view of the commitment process includes these steps:

- A work breakdown structure is defined.
- An initial estimate is made of the magnitude of the commitment.
- The disciplines involved participate in developing the implementation plan including estimating the size, effort, and schedule.
- An independent review of the plan is held.
- Agreement is reached on the commitment through negotiation and reconciliation.
- Senior management approves the commitment.
- A mechanism is provided to renegotiate the commitment in the event of a requirements change.
- A postimplementation review is held to discover what went right and what went wrong to learn how to improve future commitments and to compare actual performance with original estimates.

Of course, project teams and supporting teams must have confidence that the work can be performed within cost, schedule, and performance constraints.

The commitment process is actually on the critical path for three process areas:

- Project Planning;
- Project Monitoring and Control;
- Requirements Management.

Figure 6.7 in Chapter 6 on project planning and project monitoring and control illustrates this and complements the concepts for the commitment process presented here. Committing to the project plan or changes to it (from determining significant deviations from the plan or accepting a requirements change request) must involve negotiations with all of the relevant stakeholders and documentation of appropriate changes. SP3.3 of Project Planning Subpractice 2, Subpractice 3, and Subpractice 4 state, respectively [1]:

- Document all organizational commitments, both full and provisional, ensuring an appropriate level of signatories. Commitments must be documented to ensure a consistent mutual understanding as well as for tracking and maintenance. Provisional commitments should be accompanied by a description of the risks associated with the relationship.
- Review internal commitments with senior management as appropriate.
- Review external commitments with senior management as appropriate.

Senior management's involvement with the commitment process comes in its support of the proper execution of the commitment process and in its own visible participation. Senior management is one of the more influential relevant stakeholders. There is a misunderstanding that this involvement means that the senior manager was to engage in a technical review of the commitment being proposed. Actually, the purpose of having senior managers review internal and external commitments, or recommitments, is to ensure the necessary support for these commitments together with all of the other existing and pending commitments that have been made throughout the organization. In other words, what is needed here is a business commitment review on the part of the senior manager, not a technical one.

Senior Management Oversight

Humphrey's vision was that senior managers should conduct senior management oversight reviews for each of the projects on a periodic basis. These senior management oversight reviews should include an understanding of the processes that are being used on the projects, their efficiency, their effectiveness, and the resulting product quality that is being reached when they are followed. This oversight may be provided by discussions with "higher level management" or by process improvement presentations to the senior management team itself.

Higher Level Management

The definition of higher level management found in the CMMI®-DEV v1.2 glossary refers to the person or persons who provide the policy and overall guidance for the process, but do not provide the day-to-day monitoring and controlling of the process. Higher level management may be made up of middle managers, middle-senior managers, or the senior management team itself. At the very least, it is the responsibility of higher level management to provide the ongoing management oversight of the process improvement activities throughout the organization. Higher level management either makes decisions regarding process improvement activities that support the organization's business objectives or makes recommendations to the senior management team regarding the same. The responsibilities of higher level management include:

- Ensuring that the process improvement initiative addresses the business objectives of the organization;
- Ensuring that a discussion on processes and quality is part of the monthly or bimonthly process improvement progress meeting;
- Being aware of all of the sources of process improvement and helping to prioritize them,
- Understanding what process improvement suggestions are being considered and providing support for those that support the organization's business objectives and vision the best;
- Understanding the status of the processes being used on the various projects;

- Understanding the effectiveness of the processes being used on the projects and what the resulting product quality is and making recommendations to senior management;
- Understanding and taking into consideration the resources that are being made available to support the process improvement initiative and making recommendations to senior management;
- Making process improvement priority suggestions to senior management based on the vision and business objectives,
- Supporting the advertisement of the technical and cultural changes that are being made as a result of the process improvement initiative;
- Ensuring that new and/or revised processes are being deployed on all project start-ups;
- Ensuring that all new and/or revised processes are being applied to projects at appropriate life-cycle phases where they can and should be used.

Middle Management

Corporate Bridge

Middle management provides the corporate bridge between the programs and projects and the senior management team. For many medium to large size companies, middle management literally means a management layer that is between the senior management team and the project managers. Some organizations classify the program managers who control multiple projects as middle managers. Small organizations need to clearly spell out middle management by the functions that must be performed. A small organization may have a middle manager who performs the duties of both a middle manager and a project manager. Other small organizations may have only what we call senior managers who carry out the middle manager's role.

It is middle management who must truly understand the organization's strategic direction and the senior manager's vision. This understanding of strategic direction and vision must be communicated to all other levels of management and practitioners, so that the daily project management, development, or maintenance activities can support them. It is middle management that must emphasize these policies and procedures to the project managers and practitioners in support of the vision and business objectives. Middle management may not know all of the small technical details of the processes and procedures their people should be following, but they should be keenly aware of what is expected of their project members and what processes, methods, standards, guidelines, templates, checklists, and tools those project members are supposed to be in compliance with.

Risk Management Decision Making

Middle management should receive project and process management information. This information, combined with the knowledge of the organization's strategic direction and vision, allows them to make sound risk management decisions to guide the organization's daily operations. Thus, middle management is expected to be the advocate of a strong organizational measurement program that provides deeper insight into product and service quality along with process performance.

Process Improvement Steering Committee

Without senior management involvement, the process improvement initiative may flounder and not be oriented toward supporting the organization's business objectives. Without middle management involvement, the critical process improvement resources and the individuals who are needed to work on specific focus area improvements will probably not be made available. Middle management must be "on board" to ensure the "right" people are provided where they are needed for as long as they are needed. If individuals are to devote 25% of their time to a process improvement working group, that 25% time must be made as important to them as the 75% they are being asked to devote to project work. They must be evaluated for their contributions to the process improvement effort in the same manner they are evaluated for their development efforts. In addition, if an individual is asked to devote 25% time to process improvement, this must not be 25% above the 120% they are already working. It is middle management who owns these resources and who must balance the work such that practitioners can accomplish the business goals set by the senior management team. It is middle management who must "protect" the time allocated and spent on the process improvement working groups or the entire process improvement effort will be in danger from day one. This may be hard if the organization is small and the middle manager is also the senior manager.

Middle managers normally serve on the process improvement steering committee and should be involved in the following activities:

- Ensure that the process improvement activities are in line with the vision and business objectives that have been established by the senior management advisory board:
  - Review the proposed budget for the improvement effort.
  - Make recommendations to the senior management advisory board regarding program direction, budget, and program risks.
- Ensure that the necessary resources for the working groups and SEPG are available in a timely fashion:
  - Establish the working groups to concentrate on prioritized focus areas (e.g., commitment process, estimation procedures, testing methods).
  - Support, where needed, negotiations for people's dedicated time to the process improvement effort.
- Conduct process improvement program oversight reviews on a periodic basis (recommended once per month):
  - Ensure that software process improvement activities progress in line with documented budgets and plans.
  - Perform reviews and approval of working group deliverables.
- Provide visible support for the SEPG and working groups.

Process Owner

Although the term *process owner* has different implications in different organizations and different process improvement initiatives, I think it is important for middle managers to take on the role of the process owner. For the purposes of this discussion, *process owner* refers to the person responsible for all of the activities surrounding a process focus area. The process owner may simply be the sponsor of a working group that is facilitated by an engineering process group member, or the process owner may be a middle manager who has significant knowledge and desire to actually participate in the development of new processes or the revision of existing ones. The middle manager as process owner must participate in the periodic senior management oversight meeting and report the progress on his/her process focus area.

In this way middle management is forced to be involved with the process improvement initiative and forced to know exactly what progress has been made, what activity is being worked on, what measurable benefits have been realized, what issues have been logged, and what risks have been identified. Armed with that knowledge, the middle manager can provide additional resources and support of his/her process focus area, can report true progress to the senior management team, and can participate much more proactively in discussions and actions that ensure the process improvement initiative is achieving business results. In addition, the middle manager can provide stronger direction to the project managers and practitioners that work under his/her control.

When middle managers serve as process owners, they also find themselves with the ability to more easily relate to process improvement and quality management questions such as these:

- Would you please describe your role (as you currently see it and live it) in process improvement and change?
- How do you encourage and show your support for process improvement efforts in your organization?
- Do you feel you must change anything about your attitude, commitment, or skill base to support process improvement and change in the future?
- What change rate to you think can be realistically expected and supported for the organization worldwide?
- How will current stress on the organization affect the process improvement plans and expectations?
- How will quality management and process improvement make the senior management vision come alive?
- What level of coaching skills does the senior management and middle managers need to support the process improvement needs of the organization in the coming years?
- Where do training, mentoring, and coaching fit in with on-the-job experience? How does it work today? What are the plans for tomorrow?
- How will you help lead the organization to stay abreast of the ever changing state-of-the-art in technology?
- What are the organization’s current ideas for integrating cultures in the near and midterm future?

- What skill sets in quality management, process improvement, and change management do you think the process group, quality group, and change management group will need in order to support improvement worldwide, now and in the midterm future?
- What is the organization's greatest process improvement achievement so far?
- How does the organization recognize and reward its process improvement champions?

Project Manager

Definition of Project Management

Project management is the combination of people, processes, techniques, and technologies necessary to bring the project (or program) to successful completion. Measures of success depend on the particular project. Most projects are measured on what may now be considered the quadruple constraints of:

- Time (schedules);
- Cost (on or over budget);
- Performance (based on specifications);
- Quality (meeting or exceeding customer expectations in the intended use environment).

Looking at the business processes, these constraints are often expanded to include:

- Functionality;
- Risk;
- Customer satisfaction.

The Old Project Management Role

Historically, the project manager's role was to plan, control, organize, and direct the work of several individuals or departments so the project could succeed.

The New Skills Required of a Project Manager

Today project managers are expected to be:

- Better educated;
- Open, friendly, and people oriented;
- Better listeners;
- Quality conscious;
- Receptive to new ideas;
- More participative;

- Facilitators;
- Skilled at group process and group dynamics;
- Able to encourage others to participate in plans and decisions;
- Able to coach, inspire, and motivate;
- Able to span boundaries;
- Able to provide and apply integrative management techniques to unique, complex organizational ventures characterized by interdependent efforts, a variety of specialists, multiple sites, multiple languages, and multiple cultures.

The process areas of the CMMI® certainly expound on the project planning and project monitoring and control functions that make up the basics of project management but the concerns of the project manager are scattered among many process areas in multiple categories in the CMMI®. We will examine a reasonable subset of them here.

Estimation

Estimation is still a critical component of project management and one of the most troublesome areas for companies pursuing process improvement. The project manager is responsible for making sure the estimates described next are made and used in project planning: First, the project manager needs to work with his/her lead engineers to estimate the scope of the project by establishing a top-level work breakdown structure (WBS). The WBS defines the work or tasks to be performed and is the primary planning and analysis tool used in almost all projects. It should answer two questions: "What is to be accomplished?" and "What are the necessary hierarchical relationships of the work effort?" The WBS process also serves as a tool for the project manager by:

- Providing a complete list of the software, hardware, services, and information technology work tasks that must be completed during the development and production of a product;
- Defining the responsibility, personnel, cost, duration, risk, and precedence of each work task;
- Providing an easy-to-follow numbering system to allow hierarchical tracking of progress.

From the WBS, the project manager can direct his/her project members to develop work packages that describe what must be performed, by whom, and in what time duration. In short, they allow the project manager to start to establish project control.

The project manager is responsible for ensuring that standard estimations are made by those who will be responsible for development and testing of the product or product components. These standard estimations include:

- Size and complexity;
- Effort and cost;

- Schedule;
- Risks;
- Knowledge and skills;
- Stakeholder involvement;
- Critical computer resources;
- Technical activities;
- Quality.

Project Planning

Of course, the project manager must oversee the development of the project plan as the basis for managing the project to completion. The project plan is accompanied by a number of "supporting" plans or "plans that affect" the project. These plans may be incorporated into one physical or logical document called the *project plan* or they may be separate plans that are referenced within the project plan itself. Some of these supporting plans may be developed together with representatives of the supporting groups such as Quality Assurance or Configuration Management, but the project manager is responsible for the content in the supporting plans and for ensuring that all of the supporting plans fit together to support the project's successful completion. If asked about the contents and direction of the project's quality plan, the project manager should never answer "Go ask the quality engineer," even though the quality engineer assigned to support the project may have had a strong hand in developing the project quality plan along with other project members including the project manager.

Plans that affect the project plan that are typically developed together with the project plan for each engineering discipline include:

- Project quality plan;
- Project configuration management plan;
- Risk management plan;
- Knowledge and skills plan;
- Stakeholder involvement plan;
- Data management plan;
- Requirements management plan;
- Integration strategy;
- Verification strategy;
- Validation strategy.

Before the project plan and all other plans that affect the project are "cast in concrete," the project manager is responsible for ensuring that all supporting groups or disciplines are committed to the concepts, budget, and schedule that are contained in the total project planning package. This may involve several rounds of negotiation until total agreement is reached and the project manager is confident that all relevant stakeholders are committed to the project's success.

Criticality

The project manager should assume responsibility for working with the customer, the organization's senior management team, and outside groups such as regulatory agencies in order to determine which product components or subsystems should be treated as "critical." Criticality definitions must be defined upfront at the organizational level. These criticality definitions are then translated into quality functions such as peer reviews, tests, and audits for each level of criticality. Using the concept of "criticality," the project manager can better direct the use of the project's resources and focus the application of the project's activities including the quality functions on the project's critical components that will help optimize the factors of cost, schedule, performance, and quality.

Monitoring and Controlling

Once the project plan has been developed and commitment to it has been achieved, a major portion of project management is monitoring and controlling. The project manager is responsible for seeing that these monitoring and control activities are carried out and that the results are used to manage and control the project's activities in the future. These activities include:

- Tracking actual results against the planned estimates;
- Conducting weekly or periodic project meetings with project members and other representatives such as quality assurance and testing;
- Participating in milestone review meetings to report on progress against major milestones and resetting the project direction as needed;
- Attending senior management oversight review meetings on a periodic basis to discuss project progress, processes, and resulting product quality;
- Taking corrective action, as necessary, to keep the project on track according to plan, or by making adjustments to the plan and establishing a new commitment from all relevant stakeholders.

Requirements Validation, Functional Architecture, and Alternative Solutions

Although the initial requirements elicitation may be carried out by a multidisciplined team that may or may not include the project manager, the project manager is responsible for the evolution of those requirements and their validation with the customer or end users. This also means that the project manager must have the authority to interface with the customer with or without other organizational representation. Staying with this theme, the project manager is responsible for his/her team developing the functional architecture that will be used to guide the construction and testing of the product or product components, and for ensuring that alternative technical solutions are considered in accordance with predefined selection criteria before the final set of requirements has been established and the technical solution implemented.

Peer Reviews and Unit Testing

It is also the responsibility of the project manager to ensure that peer reviews and unit tests are planned for the life-cycle work products that are identified based on the product life cycle chosen for the project. For peer reviews, this means that the project manager is responsible for ensuring that:

- Peer reviews are planned, especially for critical components.
- Peer reviews are led by trained peer review moderators.
- Peer reviews are conducted by trained peer review team members who have an appropriate level of knowledge and skills, the right application domain experience, and the right level of project knowledge to provide useful and effective input so that they are able to help detect and remove major defects.
- Peer reviews are not considered complete until defect analysis and correction have taken place in accordance with the exit criteria established by the peer review moderator.
- Peer review data is placed into a database and analyzed for different categories of defects and trends per life-cycle phase.
- Peer reviews are used to improve product and life-cycle work product quality and for process improvement and not for employee evaluation.

For unit testing the project manager is responsible for ensuring that:

- Unit tests are planned.
- Unit tests follow project or organizational standards and templates for unit tests.
- Each unit test contains a section describing expected results that are to be compared against actual results, and corrective action is taken as necessary.
- A sampling of unit tests is observed and evaluated by quality assurance.
- Unit test results are taken into consideration as part of the transition criteria from the project environment to the integration and systems test environment.

Configuration Management

The project manager is responsible for ensuring that developmental control is carried out on the project. Whether this is supported by a centralized Configuration Management (CM) process area group or is a collaboration of the project's CM specialists working together with the organization's CM engineers, it is the responsibility of the project manager to ensure that the basic CM functions of identification, baselining, configuration control, status accounting, and configuration auditing are carried out according to the standards and guidelines developed for the organization. This responsibility includes:

- Ensuring that modules or product components are baselined only after they have been peer reviewed and unit tested with documented results;

- Ensuring that communication and cooperation exists between the project's CM specialists and the CM group engineers who are assigned to support the project's CM needs;
- Serving as the head of the project level change control board (CCB), perhaps along with the quality engineer and test representative in analyzing change requests that do not affect the agreed-on requirements;
- Ensuring that all life-cycle work products and associated plans are updated, if necessary, whenever a change request is approved at the project or organizational level. This is especially critical when it is the requirements that are changed. This ensures systems integrity.
- Using the information provided by the function of CM status accounting to manage and control the project better.

Quality Assurance

Besides working with quality engineers to develop the project's quality plan, the project manager is responsible for reviewing and responding to noncompliance reports that are the result of objective evaluations carried out on the processes, procedures, standards, guidelines, templates, and checklists that have been identified to be followed in the project's quality plan. The project manager should also support the escalation procedure that a quality engineer may follow if the project does not respond to the noncompliance reports in a timely fashion.

Supplier Management

Project managers who must also manage suppliers are responsible for their involvement with:

- Determining supplier selection criteria;
- Developing the requirements to a sufficient level to determine which requirements would or should be designed and implemented by a supplier;
- Developing the project plan to a sufficient level to determine if the supplier's estimations are in line with project expectations;
- Helping to develop the request for proposal (RFP)
- Helping to select the supplier based on the supplier selection criteria;
- Leading the orientation meeting with the supplier's team to ensure complete understanding of what is expected and who is responsible for what part of the development;
- Managing the supplier through specialized project management activities that keep track of supplier's progress and performance;
- Ensuring that the supplier's capability level is maintained through periodic review.

Risk Management

The basics of risk management are developed in Project Planning and Project Monitoring and Control and evolved with the Risk Management and Integrated Project Management process areas. Risk management is an important tool for project managers who must deal with the uncertainty of the decisions that they are required to make on a daily basis. By using the risk management parameters for categorizing or describing risks, the project manager has a better chance of being prepared when indicators or thresholds show that the risk is becoming a problem and can initiate the risk mitigation plans that were developed according to the risk management strategy that satisfies both organizational and project business objectives.

Those parameters for categorizing or describing risks include:

- Probability, which is the likelihood that the consequence will occur;
- Consequence (impact or potential loss) levels, which are generally related to cost, schedule, environmental impact, or human measures (e.g., labor hours lost and severity of injury);
- Time frame, which refers to when the risk will or might occur during the product life cycle.

Practitioners

Practitioners is a general term for those project members who participate in the life-cycle activities including requirements gathering, architecture definition, detailed design, coding or construction, unit testing, integration and systems testing, and acceptance testing and delivery. They also include members of groups such as quality assurance, configuration management, engineering process group, measurement, database, and so forth. Practitioners do not normally serve as project managers although they may serve as a lead engineer for their project.

Practitioners are not only expected to "do," but they are also expected to participate in a variety of project activities including:

- Requirements analysis;
- Estimation;
- Making commitments;
- Development;
- Tracking daily work progress;
- Developing status reports;
- Participating in project meetings and milestone meetings as required;
- Identifying risks;
- Carrying out risk mitigation activities;
- Participating in objective evaluations for process compliance;
- Participating in training, mentoring, and coaching;
- Providing training, mentoring, and coaching for other project members in their areas of expertise;

- Conducting peer reviews;
- Conducting unit tests;
- Following configuration management guidelines especially at the developmental control level;
- Providing inputs for process improvement;
- Understanding and carrying out the senior manager's vision;
- Understanding and supporting the organization's business objectives;
- Embracing changes in technical processes and organizational development and helping colleagues deal with the changes;
- Aligning personal career development goals with those of the project and organization;
- Studying and acquiring knowledge and skills that will increase individual competence levels that support the organization's business objectives.

Process Group

Organizational Focus

Organizational process improvement should be based on a complete understanding of the strengths and weaknesses of the organization's set of standard processes and the project's defined processes that are tailored from them. Process improvement occurs within the context of the organization's business objectives, but process improvement does not happen by itself or because the senior management team issues a memo proclaiming it to happen.

Facilitating the Organization's Process Improvement Activities

The responsibility for facilitating and managing the organization's process improvement activities is normally assigned to a process group. Figure 5.1 shows a sample process improvement infrastructure needed by an organization for successful process improvement along with the relative position of the process group, that is, the SEPG. This process group is normally seen as the focal group for action planning, process improvement, technology insertion, training, and awareness and expectation setting. Process groups are frequently viewed as a channel for institutionalizing the organization's knowledge of process methods, practices, and technology. Process groups are the organization's "champions of change" and its members "change agents."

Collectively the members of the process group need to be able to demonstrate their ability to manage, develop, coach, and guide process improvement and its accompanying cultural changes. First and foremost, they must understand senior management's vision and the organization's business objectives in order to be able to efficiently and effectively guide the process improvement effort. Without this explicit knowledge, the organization's process improvement effort may demonstrate compliance with a model such as the CMMI®, but will not be supportive of the organization's business objectives at all.

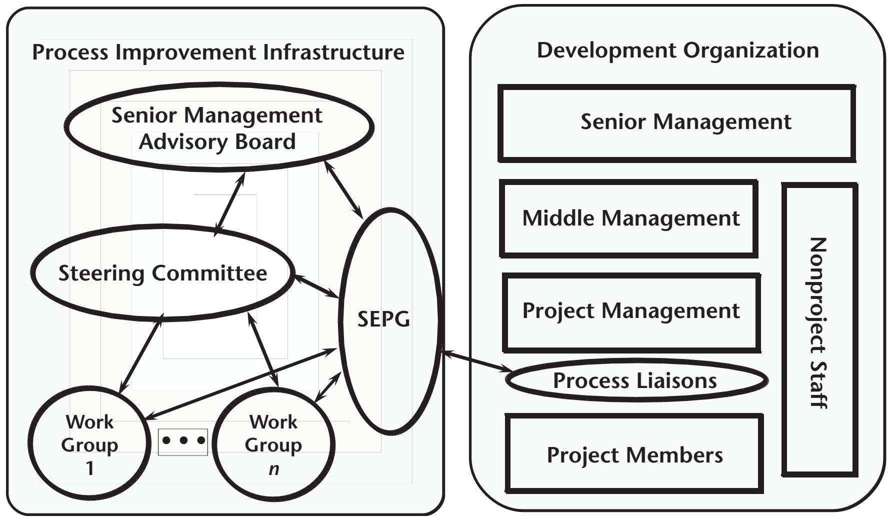


Figure 5.1 Sample process improvement infrastructure.

Engineering Background

Process group members should have a solid engineering background. They should have a general knowledge of the organization's application domains and knowledge of modern engineering techniques and methods. They must be up to date on accepted engineering standards (i.e., DoD, MoD, IEEE, ISO, IEC ESA, NASA). They should have documented experience that they have or had implementation skills in one or more engineering disciplines. In other words, they are or have been capable of "doing the job." They should also have a general understanding of the other engineering disciplines that the projects must integrate and the processes must support. They must also have a good understanding of the project management support functions such as quality assurance and configuration management. Managers and engineers alike must respect them. They must have a strong knowledge and good experience in project management and a working knowledge of metrics to help the project managers manage and control their projects better.

Process group members must be people oriented with superior communication skills and the willingness to perform most of their work in the project developers' offices, not their own. They should always be ready to provide "handholding" support for the managers and practitioners on various projects when the process ideas are being introduced.

Organizational Development Skill

Although process group members must have the technical background to maintain credibility with the product or product component developers, they must also be knowledgeable in organizational development skills as well (i.e., managing technological change, team building, collaborative consulting) to effect successful technology transition.

Process Group Responsibilities

Although the many tasks attributed to the process group are important ones for its members, it must be stressed that the job of the process group is to be the champion of the process improvement effort. This group is expected to facilitate the process of change—*not be responsible for the process change.* Process group responsibilities include, but are not limited to:

- Coordinating the process improvement initiative up, down, and across the organization:
    - Participating in the senior management advisory board reviews;
    - Participating in the steering committee reviews;
    - Facilitating the activities of the working groups, which means staying on top of what is going on, what difficulties are being encountered, and what successes are being realized;
    - Promoting technical awareness and education about process improvement.
- Managing/facilitating the process improvement initiative:
    - Facilitating the definition/improvement of the technical and managerial processes, methods, techniques, and tools for developing and maintaining product and product components;
    - Assisting in the evaluation of new tools and techniques based on their understanding of the existing processes;
    - Facilitating the definition and maintenance of organization policies and standards for processes and products;
    - Discovering "good practices," getting them adapted for general use on the projects throughout the organization, and baselining them as "best practices";
    - Overseeing and facilitating pilot projects and implementation of improvements into the projects and across the organization;
    - Directing the definition of process metrics, initiating the collection of data, and assisting the working groups and projects in the analysis and use of the resulting information.
- Ensuring that the processes are "living":
    - Maintaining a dialogue with project personnel regarding the application and performance of developing processes;
        - Sharing good ideas from other parts of the organization;
        - Listening to issues/ideas from the practitioners;
    - Initiating periodic process improvement progress checks and reassessments;
    - Initiating practitioner-driven review of specific processes.
- Maintaining a library of "process assets":

- Overseeing the process database (process asset library) for product and process assets used across the organization;
- Facilitating the development and retention of tailoring guidelines for specific use of the assets in the process asset library.

Process Group Manager

Each process improvement infrastructure should have an identified process group manager. This individual is a senior person with most of the attributes listed in the section on what process group members should know. The process group manager is responsible for coordinating all of the process improvement activities throughout the organization. He/she has direct access to the senior manager and serves as the link among the senior management advisory board, the steering committee, and the working groups (see Figure 5.1). The process group manager serves as the link between the organization's process improvement initiative and the organization's line, function, and project management.

To ensure that a process improvement initiative moves forward, the process group manager must function as a project manager for the process improvement initiative. This means that he/she must utilize basic project planning and tracking tools and must set milestones, hold meetings with agendas, record minutes, track action items, and so forth. Without strong project management skills and follow-through, the initiative will lack leadership and will not progress in a timely fashion.

The process group manager is also the spokesman for the process improvement initiative. This involves continuous oral and written communication about the process improvement effort, its successes, and its failures. The process group manager must be willing to give a presentation on any aspect of the process improvement initiative many times before the rest of the organization starts to accept that message. The process group manager is the lead organizational champion for the process improvement effort.

Quality Assurance

The classic responsibility of quality assurance is to provide visibility into the processes being followed on the projects for the management team at all levels and to determine if they are efficient, effective, and producing the necessary product quality to satisfy customer, competitor, organization, or project quality goals. Figure 5.2 shows quality assurance participating in three distinct but related directions. The first direction is to management as just described. The second is to developers. This responsibility is inadequately executed worldwide based on my experience of assessments and process improvement consulting. Quality Assurance Responsibles should provide feedback to the individual projects and project members on the efficiency and effectiveness of the processes that the project members are required to follow so that they can be improved at the project level as necessary. Third, when the quality assurance engineers find that project members are indeed faithfully following the organizational processes, but they are not efficient or effective or they are not producing the required product quality, the quality assurance engineers are

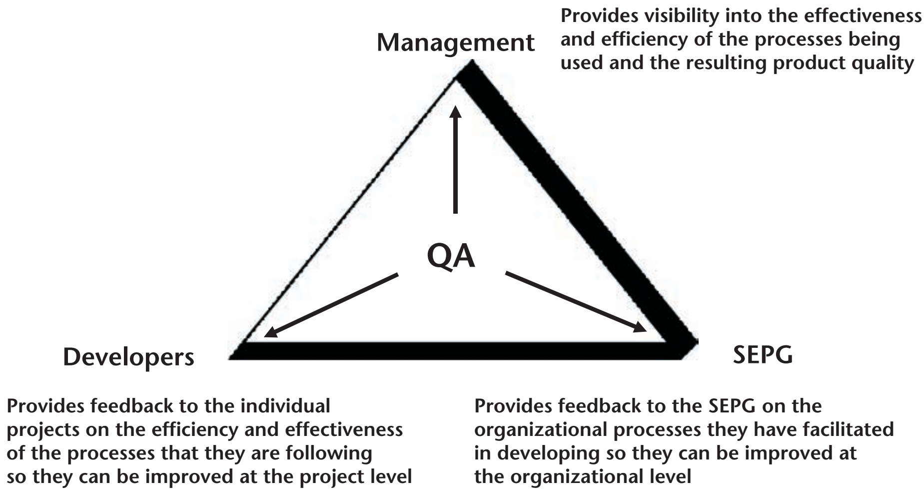


Figure 5.2 Agent for process improvement.

responsible for providing this feedback to the process group so that these processes can be improved at the organizational level.

In addition to the process compliance and process improvement aspect of quality assurance, some basic quality assurance roles and responsibilities include:

- Assisting the project manager in developing quality goals;
- Assisting the project manager in creating the project's quality plan;
- Supporting the project in selecting an adequate set of standards, practices, and procedures;
- Identifying all deviations for the agreed-on processes as early as possible;
- Ensuring that all identified deviations are recorded, corrected, or escalated to the appropriate level of management, up to and including the senior manager;
- Negotiating criticality levels for the product components and subsystems, with the various engineering disciplines and project managers;
- Ensuring that all detailed technical activity plans are made available and reviewed for conformance with process definitions;
- Performing ad hoc process compliance evaluations to ensure the quality functions are being carried out along with the normal technical project activities;
- Interfacing with the appropriate customer representatives on process and/or product quality problems;
- Evaluating the supplier's quality plan and resulting implementation to ensure that any required policies and procedures are being adhered to;
- Evaluating the supplier's quality assurance activities to ensure that the supplier's product or product component will not downgrade the product quality required for the integrated system or subsystem;
- Evaluating the project's and organization's configuration management activities to ensure the integrity and consistency of the work products and the bidirectional traceability of those life-cycle work products back to the requirements;

- Ensuring that the customer receives a complete and correct description of the delivered product or product components and accompanying documentation;
- Ensuring that the development tools and utilities used to develop and test the product or product components fit the development process;
- Serving as the internal "customer representative" to keep that point of view in front of the development team.

Configuration Management

Those individuals who participate in the configuration management function either at the project level, organizational level, test level, or release level are responsible for helping the projects to preserve the results of the hard work that has been done. At the same time, they provide for a controlled, systematic way of making necessary changes.

As described in detail in Chapter 8, configuration management engineers and managers, and other individuals such as the project manager, the quality assurance engineer, and the test manager, work together to enable the proper execution of the basic configuration management functions including:

- Identification;
- Baseline of the release and documentation;
- Change control;
- Configuration management status accounting;
- Configuration auditing;
- Managing the configuration management system;
- Interface control (both technical and organizational);
- CM tool support;
- Supplier control;
- Migration from developmental change control to organizational change control.

The organizational CCB is responsible for controlling any and all changes to the requirements that represent the agreement or contract with the customer. In that capacity, the organizational-level CCB takes on the responsibility of representing the interests of all groups and disciplines that may be affected by the proposed changes to existing documentation baselines. Even in a situation where there is a project with a definite beginning and definite end only focused on one engineering discipline such as software, CM control at various levels is required. Some organizations have also established a requirements CCB (RCCB) that includes the chief architect. All requirements and requirements changes including those that originate from trouble reports are first funneled to this RCCB to determine if the requested change would have an impact on the established architecture and strategic business direction. If it is determined that no significant impact would be realized to the architecture or strategic business direction, the requirements change request would be passed on to the organizational CCB.

The CCB, at the organizational level, authorizes the establishment of organizational-level baselines such as the functional, allocated, or product baselines; authorizes changes to those baselines; and authorizes the creation and release of products and product components from the product baseline. Documentation control at the organizational level will also require a CCB function.

As described in the roles and responsibilities section for the project manager, the project-level CCB is responsible for the change control activities for all project developmental baselines that are under the control of the project manager. The basic configuration management functions must be performed, but may be done so with slightly less formality than if those functions were focused on customer requirements. Project-level change control together with organizational change control is responsible for the smooth transition from developmental or project control to formal systems or organizational control as the product components move through the product life-cycle stages to become an integrated, verified, and validated product.

Integration and Systems Testing

To maintain clear and obvious objectivity during the integration and systems testing phase of the product life cycle, many organizations establish an independent test team that focuses on those functions. Such a testing group is expected to have application knowledge and understanding, a test methodology, testing experience, and knowledge of the test environment and be able to exhibit creativity, insight, determination, and analytical ability.

Although often not the fault of the integration and systems test group, integration and systems testing engineers typically are not involved enough in the early phases of a product's life cycle. This is unfortunate, because their activities must also be planned—and the earlier, the better. Independent test representatives from the integration and systems test group should also be involved in peer reviews of product or systems requirements specifications, and the engineering discipline allocated requirements specifications such as the software requirements specification or the hardware requirements specification. This will eventually form the basis for bidirectional traceability.

Independent test representatives should be involved in peer reviews of interface specifications and architecture specifications. Experienced test engineers can best answer questions such as "Is this requirement testable?" Even before the development of the various specifications, independent test representatives can be very effective by participating on multidisciplined requirements elicitation teams together with systems engineers, software engineers, marketing, program management, design experts, and so on, to help the stakeholders better understand what the testing implications are of their wants, needs, constraints, and interface requirements.

To support integrated project management, test group representatives should work with the project manager and lead designers to develop an integration strategy and determine the integration sequence and testing environment that will support the project's needs.

During the actual integration and systems testing activities, the integration and systems test group is responsible for carrying out:

- Stress, load, and boundary testing;
- Functional testing;
- Quality factor testing such as reliability, maintainability, and portability;
- Documentation testing to ensure that the user and maintenance documentation matches the system to be delivered.

Prior to performing the integration and systems testing, the integration and systems test group may work closely together with the Configuration Management and Quality Assurance PAs to ensure that the verified and validated product components are indeed ready to be integrated and tested.

When the project uses suppliers, the integration and systems test group is responsible for developing the acceptance tests and performing all necessary acceptance testing activities to confirm that the supplier's confidence in the delivered system was justified.

In many IT shops, the customer expects the development organization to provide a partial or full test team to work together with that organization's test team to carry out acceptance testing. Although not normally considered the development organization's responsibility, it is often a good means to understand more of the customer's capabilities for future design, development, and testing of deliverable systems.

Measurement Team

Most organizations have at least one person who has an interest in and an ability to understand metrics and measurements. Few organizations have a designated measurement group. While it may not seem worthwhile for an organization to form a separate measurement group, it has been the author's experience that having a measurement expert or two supporting the organization's metrics needs is quite valuable. The author has typically seen one of two scenarios:

- The "metrics guru" collects a lot of data that is hidden in his/her desk on flash drives and CD-ROMs and is not used or shared by anyone else.
- The "metrics guru" collects data and sends out reports combined with awe-inspiring supporting graphics that few, if any, project managers or project members can understand, much less use.

The CMMI® does not demand an organizational measurement group, but it is recommended that middle and large sized organizations designate a measurement team to facilitate and support the organization's measurement needs. Even small organizations are encouraged to have a team member who can, from time to time, provide guidance on establishing a measurement program. Still and all, it should be remembered that the CMMI® provides management with visibility that is needed in order to make better decisions. Therefore, the importance of effective communication is extremely important. The development of a metrics guide to aid in the explanation and interpretation of metrics trends as soon as possible is invaluable! This helps to ensure that members of the management team receive information that they can interpret and effectively use in their decision making.

The measurement team is expected to support organizational measurement needs at every management and practitioner level that supports the organization's business objectives and the project's information needs. The use of multilevel Balanced Scorecards can assist in this linkage of business objectives to measures. Team members are responsible for assisting the organization and its projects in defining measures based on business objectives. In addition they are responsible for:

- Helping to determine data collection schemes;
- Helping to determine data storage and retrieval schemes;
- Determining appropriate analysis techniques in advance of the data collection and analysis;
- Analyzing the data based on the agreed-on analysis techniques;
- Calculating derived measures;
- Providing information to the projects based on the analyzed measures;
- Coaching project managers to use measurement results to manage and control the project better;
- Ensuring that stored measures also includes the information or context needed to understand and interpret the measures (i.e., a measurement guide).

In addition, the measurement team, in cooperation with quality assurance and the process group, helps to develop measures that will determine the effectiveness of the processes the projects are following.

Measurement is critical for every project's success. A measurement team will prove quite useful for supporting that critical informational need.

Systems Engineering

Systems engineering provides a "cradle to grave" view of the evolving system. Systems engineers help to define the total technical and managerial effort required to transform the set of customer needs, expectations, and constraints into a life-cycle balanced solution. This includes the definition of technical performance measures, the integration of engineering specialties toward the establishment of a product architecture, and the definition of supporting life-cycle processes that balance cost, schedule, performance, functionality, quality objectives, risk, and customer satisfaction.

Specific systems engineering activities include:

- Serving on a multidisciplined team to elicit requirements from the identified stakeholders;
- Transforming those "customer" requirements into product and product component requirements that can be used by project members to refine and build the product;

- Allocating the technical requirements to the various disciplines such as software, hardware, mechanical engineering, hydraulics, manufacturing, people, and processes;
- Supporting the definition of the overall systems architecture and definition of functionality;
- Defining interfaces among systems components;
- Analyzing requirements change requests to ensure the chosen optimal alternative technical solution is not adversely affected;
- Supporting integration and systems testing;
- Supporting the project manager by providing a total systems view throughout the entire product life cycle.

Summary

Process improvements that support business objectives require the cooperation and coordination of all levels of management and practitioners. The short descriptions of the roles and responsibilities of the various levels of management and practitioners that were provided have been taken from various places throughout the CMMI® and were combined with the author's own ideas and interpretations to provide the widest and most flexible but usable definitions.

Reference

[1] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI® Second Edition, Guidelines for Process Integration and Product Improvement,* Reading, MA: Addison-Wesley, 2007.

PART II
Project Management

CHAPTER 6

Enabling Project Managers to Manage and Control Better Through Project Planning and Project Monitoring and Control

This chapter through Chapter 10 focus on the inclusive topic of project management. The topics of project planning and project monitoring and control are covered in this chapter. This chapter also serves as the beginning of the description of project management. The scope of project management encompasses activities that help the project manager to *manage and control* his/her project better. It covers the traditional project management activities such as project planning and project monitoring and control. It will also cover risk management, quality assurance, configuration management, supplier management, and finally integrated project management.

Risk Management has been placed in the CMMI® as a separate process area. Placing Risk Management as a separate process area in the CMM® or CMMI® reference model was debated for years. It was finally placed in the CMMI® v1.1 as a separate process area in order to call attention to its importance in managing both successful projects and successful businesses. However, in the author's opinion, risk management should not be implemented as a separate function, but as a critical part of project management! The Risk Management PA will be covered in detail in Chapter 7.

A representation in CMMI® is analogous to a view into a data set provided by a database. Both representations provide ways of implementing process improvement to achieve business goals. Levels are used within CMMI® to describe an evolutionary path for an organization to follow that wishes to improve the processes it uses to develop and maintain its products and services. Within the continuous representation, capability levels are used to determine an organization's process improvement achievements in individual process areas such as Requirements Development or Configuration Management. Within the staged representation, maturity levels are used to determine an organization's process improvement achievement across multiple process areas, which are predefined for each of the five maturity levels.

Although the continuous representation of the CMMI® chose a categorization scheme that placed CM and QA in the category of support, it is my experience that effective use of the engineering principles of CM and QA are best realized by thinking of them as project management functions. Chapter 9 on quality management will explain that quality assurance can not only ensure that defined and agreed-on processes are being followed on projects, but that quality managers or quality engineers can and should act as advisers to the projects they support and provide quality reports that provide information on which project managers can take action.

The Configuration Management PA, also described in Chapter 8, covers a full set of functions that provide a project manager with insight into the progress and quality of the evolving product or product component. For these reasons, these functions will be discussed as part of project management.

The Supplier Agreement Management PA addresses the increasingly important topic of supplier management. Although much has been written on management of suppliers, it is believed by many that effective supplier management means that a project or business unit must have established and maintained effective requirements engineering, project management, and quality management processes in order to properly and effectively apply them to their suppliers. Supplier management will be presented in Chapter 9.

The Integrated Product Management PA can be thought of as the carrying out of the project management functions discussed in Chapters 6 through 9, but based on the organization's set of standard processes. Integrated project management will be discussed in Chapter 10, which will also act as the conclusion to the overall discussion of project management.

Project Planning

Project management can be thought of as the establishment and maintenance of an environment that enables work to be performed efficiently and effectively. To effectively manage and control a project, the project leader must be able to identify the customers, define and manage the requirements, understand the system that must be built, establish the necessary project roles, understand the project factors that must be managed, establish the project management life cycle, and choose a product life cycle. These and more roles are defined in Figure 6.1.

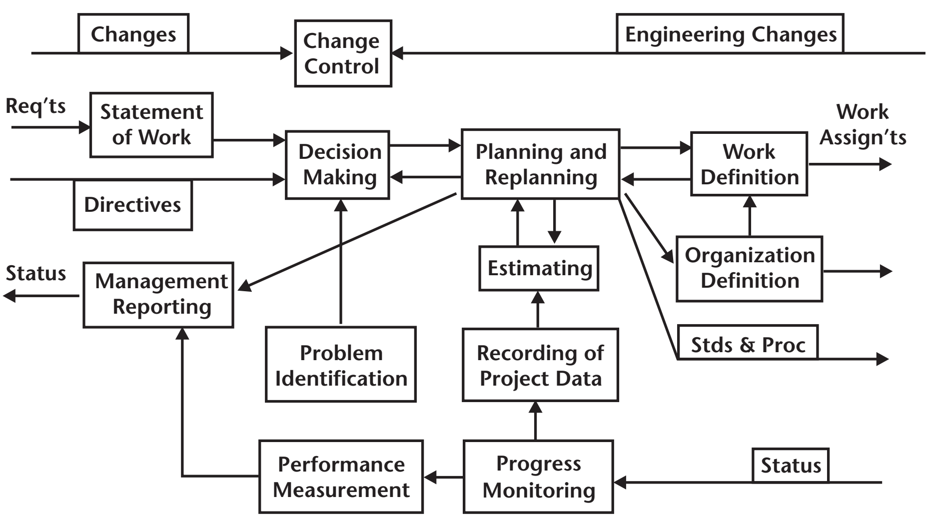


Figure 6.1 Project management.

The purpose of project planning is to establish and maintain plans that define project activities and activities of other relevant stakeholders that affect the project. This involves estimating, developing the project plan, interacting with the relevant stakeholders, resolving conflicts, getting commitment to the plan, and then maintaining it throughout the project management life cycle. The project plan will inevitably need to be revised as the project progresses. During the project's life cycle, it will become necessary to address changes in requirements, commitment changes, inaccurate estimates and assumptions, necessary corrective actions, and process improvements. Figure 6.2 shows the iterative nature of planning.

In addition to the functional and nonfunctional requirements, project planning is driven by the system overview, system objectives, and system constraints. The system overview provides insight into why the system is being built: to provide research and development to a specific customer to meet an existing or anticipated market need, or possibly to upgrade an existing system to newer technology?

Effective project planning should also provide the project with insight into whatever problem is to be solved. Is the problem a known one? Is it an anticipated problem such as the loss of production support or support for existing technology? In addition, the system overview should describe the interactions between the proposed system and its environment. For example, project planning should describe the new system interfaces with other existing systems. Project planning should mention the protocols that must be used to interface with them. It should describe the end user profiles and in what manner they will use the system. And certainly project planning should describe any expected or anticipated changes in the environment over its lifetime.

Constraints

System constraints would include both design constraints as well as functional constraints. Design constraints usually include:

- Development standards and processes;
- Compilers;
- Operating systems;
- Operating environment.

Figure 6.2 Iterative nature of planning.

Functional constraints usually include:

- Performance;
- Efficiency;
- Response times;
- Capacities;
- Safety;
- Security;
- Quality factors such as maintainability, expandability, reliability, and portability.

The final deliverable may not necessarily be the actual “best” solution to the customer’s problem, but it is usually the one that optimizes the constraints of cost, schedule, performance, quality, functionality, risk, and customer satisfaction! These constraints are a good starting point for defining what is considered to be a “major defect,” when projects are conducting peer reviews and testing. It is also a good starting point when attempting to set the selection criteria for choosing alternative solutions. We will refer to these constraints, as appropriate, throughout this book.

Scope Description

The scope description is a description of what will be produced. The scope must be defined and understood in order to effectively develop a scheme for the identification and the organization of the work that must be managed. The scope serves as a communication tool to ensure that what is being produced is properly described. It should also describe what the customer and sponsor want the project to produce. It should describe the critical features and functions that are related to the final deliverable. The scope description includes:

- The restated definition of the customer’s needs;
- A detailed description of the final deliverable, including the features and functions that will be included;
- The customer’s criteria for acceptance in measurable terms;
- A description of the scope boundaries;
- A description of the end point for the final deliverable.

Inputs for defining the project scope include:

- Project roles and responsibilities;
- Initial work breakdown structure (WBS);
- Product and product component requirements;
- Operational scenarios;
- Approved system/software life cycles.

The major tasks that must be performed on each project for defining the project scope are:

- Assignment analysis and boundary determination;
- Establishing and maintaining the WBS;
- Choosing and defining the appropriate systems/software life cycle.

The scope description should be reviewed before it is finalized. A checklist such as the one shown in Table 6.1 will help.

Work Breakdown Structure

The work breakdown structure serves to help estimate the total scope of the project. The WBS is developed and used to divide the overall project into an interconnected set of manageable components. It should then evolve with the project. The WBS is typically a product-oriented structure that helps to identify and organize the logical units of work, the work packages, to be managed.

The WBS answers two questions: "What is to be accomplished?" and "What is the necessary hierarchical relationship of the work effort?" The WBS is structured in accordance with the way the work will be performed and reflects the way in which the project costs and data will be summarized and eventually reported. The WBS is normally represented as a treelike structure. The most common WBS is the six-level indented structure shown in Figure 6.3.

The work breakdown structure normally contains:

- Scope of the work based on the requirements:
    - Technical goals and objectives;
    - Identification of customers and end users;
    - Imposed standards;
    - Assigned responsibilities;

Table 6.1 Scope Checklist

Yes/No	*Checklist Question*	*Rationale*
	Have you identified the problem that the final deliverable is supposed to solve?	
	Will the final deliverable help to resolve the problem the customer is expecting in its intended environment?	
	Does the scope description accurately describe what will be produced by the project?	
	Is there a consensus on what will be produced by the project?	
	Has the list of customer requirements been properly elicited?	
	Are the customer's acceptance criteria written from the customer's point of view?	
	Is the end point for the project clearly defined?	
	Have all relevant stakeholders been identified that will be affected by the project or that can affect the project?	
	Have any project overlaps been identified?	
	Did the project team actively participate in defining the scope of the project?	

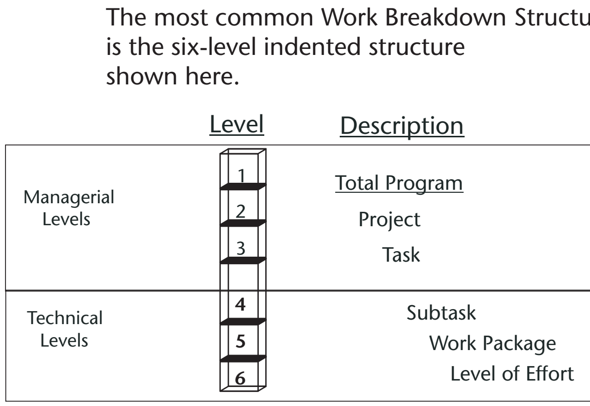


Figure 6.3 Work breakdown structure.

- Cost and schedule constraints and goals;
- Dependencies between the project and other organization;
- Resource constraints,
- Other constraints for development or maintenance;

- Initial identified risks;
- Breakdown of activities:
  - Development activities;
  - Activities to acquire required skills and knowledge;
  - Activities for integration and life-cycle management of nondevelopmental items;
  - Supporting activities and associated plans (configuration management, quality assurance, supplier management, project management);
- Work products:
  - Deliverables;
  - Nondeliverable work products;
  - Work products that will be externally acquired;
  - Work products that will be reused;
  - Nondevelopmental work products that will be integrated;
  - Work products that will be placed under configuration management control.

Estimation

Some of the factors to consider when estimating project planning parameters include product requirements, identified tasks and work products, the technical approach chosen, the selected project life-cycle model, the size and complexity attributes of the work products, the models or historical data for converting the estimates into labor hours and costs, and the methodology used to determine needed materials, skills, labor hours, and costs.

The estimating rationale and supporting data should be documented for the review and commitment of stakeholders to the plan and for maintenance of the plan as the project progresses. Rationale is demanded throughout the CMMI® to support decision making. Why is it so important to document the rationale at the time the decision is made?

For one thing, it makes clear what was going on in the minds of the participants when the decision was made. If something goes wrong 6 months down the road, the original decision makers or even those who have been authorized to take over can review the rationale and perhaps determine what caused things to go wrong, or even why they may have made a wrong decision. It the rationale is not documented, many people can spend hours and even days trying to reconstruct the thinking that went into the decision making.

The Project Life Cycle

The project life cycle consists of phases that are predefined or need to be defined depending on the scope and nature of the project. Larger projects may contain multiple life-cycle phases such as concept exploration, development, production, operations, and disposal. Development phases (for example, for software engineering) may include:

- Subphase descriptions for requirements analysis;
- Design and construction;
- Integration and verification.

Intermediate phases may require prototypes, increments of capability, or spiral model cycles. Project life cycles with defined stages include:

- Evolutionary;
- Incremental;
- Spiral;
- V-model;
- Waterfall;
- Overlapping waterfall.

Figure 6.4 provides an illustration of an incremental life-cycle model.

Size Estimation

Size is the most commonly accepted attribute used to estimate effort, cost, and schedule. However, other attributes are also very important and some of them can and should be used with size to ensure greater success in estimation. These attributes include complexity, connectivity, and structure. For example, a program could be very small in terms of lines of code, but actually contain very complex algorithms that could result in lower than expected productivity. A relative level of difficulty or complexity should be assigned for each size attribute.

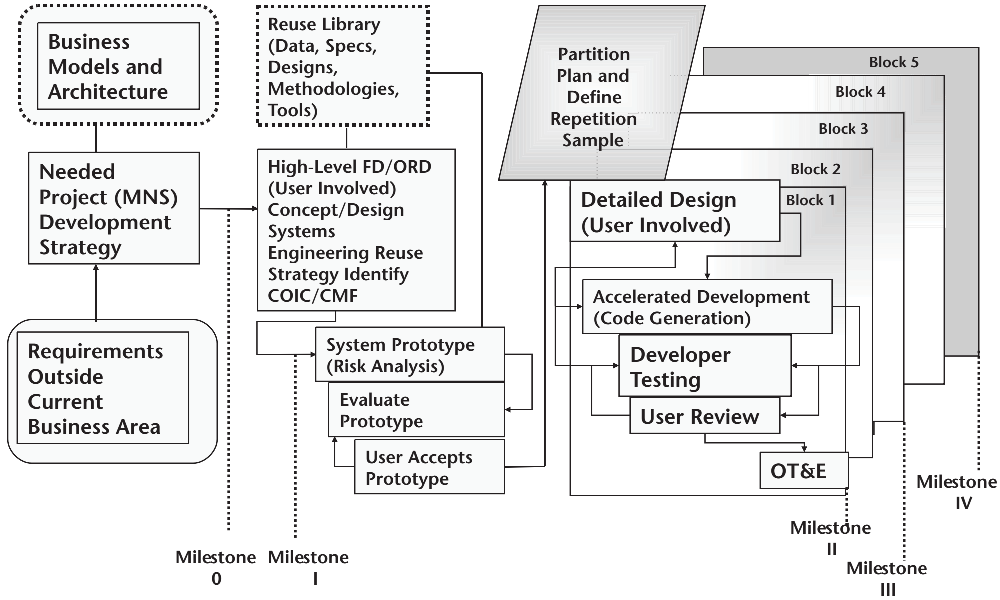


Figure 6.4 Incremental development processes.

Examples of size measures include:

- Function points;
- Source lines of code;
- Number of classes and objects;
- Number and complexity of interfaces;
- Number of pages;
- Number of inputs and outputs;
- Number of technical risk items;
- Volume of data;
- Number of logic gates for integrated circuits;
- Number of parts (e.g., printed circuit boards, components, and mechanical parts);
- Physical constraints (e.g., weight and volume);
- Number of computer screens that must be created.

One popular estimation technique is the Delphi estimation technique, which is illustrated in Figure 6.5. The Delphi approach includes recognizing that there is always uncertainty in estimating. The Delphi method focuses on utilizing the most knowledgeable and experienced people and asking those people to estimate the size of the project using three estimates:

- *Optimistic:* best case, smallest estimate;
- *Expected:* most probable, middle estimate;
- *Pessimistic:* worst case, largest estimate.

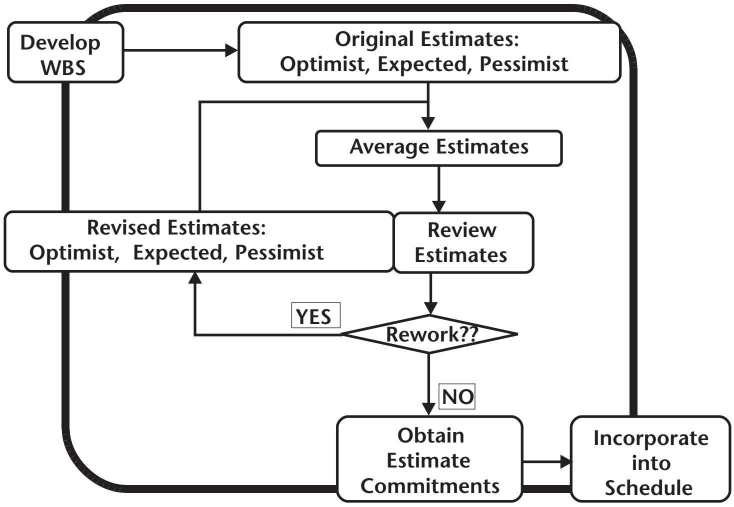


Figure 6.5 Delphi technique.

Taking these three values and putting them into an equation, such as the one shown next, takes into account the "best guess" and shifts the estimate in the direction corresponding to the most likely value, considering the uncertainty of the estimate.

$$E = \frac{\text{Optimistic} + 4\ \text{Expected} + \text{Pessimistic}}{6}$$

This may be viewed as a statistical approach with an approximate standard deviation of SD = (Pessimistic – Optimistic)/6. Multiplying the expected value by 4 approximates a central tendency and a normal distribution. The value of E is a weighted average that takes into account both the most optimistic and pessimistic estimates provided, and SD measures the variability or uncertainty in the estimate [1].

To produce an estimate, the project manager:

1. Decomposes the project into a list of estimable tasks, that is, a work breakdown structure.
2. Estimates each the E value and SD for each task.
3. Calculates the E value for the total project work as E (Project work) = Σ E (Task).
4. Calculates the SD value for the total project work as SD (Project work) = $\sqrt{\Sigma}$ SD (Task).

We then use the E and SD values to convert the project estimates to confidence levels as follows:

- Confidence level in E value is approximately 50%.
- Confidence level in E value + SD is approximately 70%.
- Confidence level in E value + 2 × SD is approximately 95%.

- Confidence level in E value + 3 × SD is approximately 99.5%.

Effort and Cost

Estimates of effort and cost are determined from the size and complexity estimates. Historical data and/or models are applied to the planning parameters to result in the estimates of effort and cost. Cost planning parameters include:

- Risks;
- Critical competencies;
- Allocated requirements;
- WBS;
- Cost of externally acquired work products;
- Knowledge and skills training, mentoring, and coaching needs;
- Capability of the tools in the engineering environment;
- Travel required;
- Level of security required.

Schedule

The project's schedule is also established and maintained based on the size and complexity estimations. Some activities that are normally used to establish a project's schedule include:

- Determining the time phasing of the work activities;
- Determining the inch-pebbles and milestones to support progress measurement;
- Defining activities of appropriate duration;
- Availability of resources;
- Skill level of work team;
- Critical dependencies on suppliers of hardware and software;
- Defining milestones of appropriate time separation;
- Using historical data for schedule verification.

Risk

Although Risk Management is a separate process area in the CMMI®, risk management is an activity that is integral to successful project management. Risks should be identified early in the project life cycle starting with the WBS, developed during the estimation process, and should take into consideration the cost, resources, schedule and technical aspects of the project. Risks should be analyzed to determine the impact, probability of occurrence, and time frame in which the problem(s) are likely to occur.

Data Management

Data includes various forms of documentation required to support a project in all of its areas. According to the definition of data found in the CMMI®-DEV v1.2 Glossary, data is defined as [2]:

> Recorded information, regardless of the form or method of recording, including technical data, computer software documents, financial information, management information, representation of facts, numbers or datum of any nature that can be communicated, stored, and processed.

Data are the various forms of documentation required to support a program in all of its areas including:

- Administration;
- Engineering;
- Configuration management;
- Finance;
- Logistics;
- Quality;
- Safety;
- Manufacturing;
- Procurement.

Data may be reports, manuals, engineering notebooks, charts, drawings, files, e-mails, and project correspondence. E-mails are especially important to estimate and control because they are increasingly used as legal documents showing intent and decisions made. Data requirements for the project should be established for both the data items and their contents. Data may be deliverable or nondeliverable, and may include such items as informal data, trade studies and analyses, internal meeting minutes, internal design review documentation, lessons learned, and action items.

From the project manager point of view, a data management plan must be developed that will clearly define the following: What data must be managed to support the project's successful completion, when is this data going to be collected and used during the project life cycle, and who is going to support the project manager? It should indicate who will be collecting the appropriate data, analyzing it, interpreting it, and then providing feedback and coaching to the project manager in how to use it in the management of the project.

By now it should be clear to the reader, however, that the project manager is not solely responsible for collecting the data or for analyzing it. Different people in different departments may be handling the required data according to the data management plan and supporting the project manager in its use. Figure 6.6 shows some examples of data management stores that might be controlled by the departments supporting the project and some of types of data that might be controlled by the project manager.

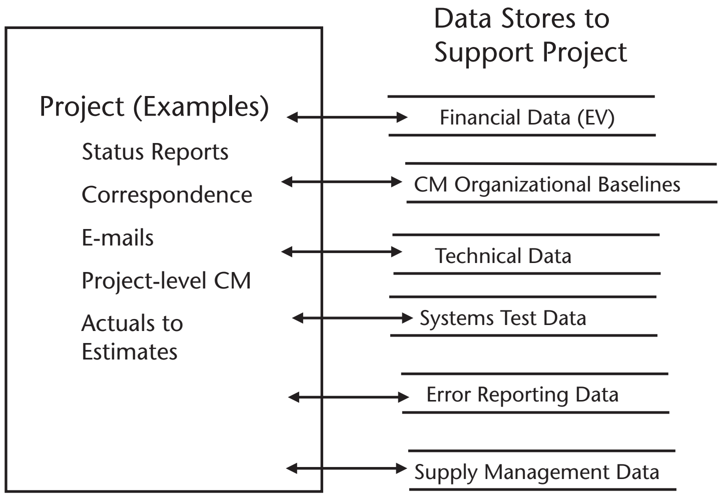


Figure 6.6 Data management stores.

Knowledge and Skills

The project team member's knowledge and skills must be compared against the proposed or assumed knowledge and skills base that is used for the estimation of effort, cost, and schedule. Too often projects are estimated based on fictitious project members with intermediate or high-level project team member skills and experience. The actual allocation of human resources frequently results in the project manager acquiring a large percentage of project members with less knowledge and skills than those planned for during his/her initial estimation. This should result in reestimation before the project plan is built. It is strongly recommended that parametric estimation be considered here as well. If the project manager accepts initial estimates that are based on 20 people with 20 years of experience, and he/she in fact receives 5 people with 20 years of experience and 15 people with 7 years of experience, then some adjustment should be done. A risk management plan should be developed to address how he/she will mitigate the effect of three-fourths of the project team having only 7 years of experience. This is now a constraint that will impact the project. In addition, some of the project planning attributes or parameters, such as effort and/or schedule, might have to be adjusted. This adjustment could be made by applying a weighting factor, perhaps as high 1.5 times the original estimate, in order to adjust for the lack of experience. Note that parametric estimation along with risk management should be applied to most of the project's estimates, not just the knowledge and skills estimates that we are discussing here.

Stakeholder Involvement

For each major activity, the stakeholders who are affected by the activity and those who have expertise needed to conduct the activity should be identified. The stakeholder list normally changes as the project moves through the product life cycle. Stakeholders in the later phases of the life cycle should have early input into the requirements and design decisions that affect them. Stakeholders may include:

- Senior managers;
- Project functional managers (e.g., systems engineering, software engineering, mechanics);
- Support management (e.g., quality assurance, configuration management);
- Financial managers;
- Subcontractors/suppliers;
- Customers and end users;
- Project members or developers in the various disciplines;
- Testers;
- Database administrators;
- Marketers;
- Maintainers;
- Regulatory agencies;
- Manufacturers;
- Logistics personnel.

Many projects develop a matrix that includes the life-cycle phases, and a list of relevant stakeholders that must be involved, or consulted, or informed during each phase. What is most important is that stakeholder involvement is tracked in addition to other project planning parameters in order to ensure that the relevant stakeholders are truly involved at the times planned for and at the right level of involvement. I have found that many companies have the stakeholder matrix clearly defined. But when relevant and significant stakeholders are not able to show up, or for one reason or another decide not to participate when they have agreed to do so, they will often send delegates to act in their stead. That would be quite acceptable if those delegates also have the authority to make decisions and the responsibility to attend the meeting and perform the designated function. This, then, is the most frequent problem that organizations face. Technical leaders and middle to high-level managers are more than happy to delegate their responsibility to someone with a lower title, but are frequently unwilling to give them any authority to actually do something!

However, developing a stakeholder matrix covers only part of the intent of this activity within project management. A stakeholder plan must be developed that gives credibility and clarity to the stakeholder matrix. This plan for stakeholder interaction should include some or all of the following:

- List of all relevant stakeholders;
- Rationale for stakeholder involvement;
- Expected roles and responsibilities;
- Relationships among stakeholders;
- Relative importance of stakeholder to project success by phase;
- Resources needed to ensure relevant stakeholder interaction;
- Schedule for phasing of stakeholder interaction.

Project Resources

Project resources, including labor, machinery, equipment, materials, and methods, and their required quantities build on the initial estimates and provide additional information that can be used to expand the WBS. Infrastructure needs must also be considered such as critical computer resources for software engineering. The infrastructure resource needs in the development environment, the test environment, the production environment, the target environment, or any appropriate combination of these when estimating effort and cost must be considered. A few examples are shared here:

- Memory, disk, and network capacity;
- Processor power;
- Communication channel capacity;
- Workstation power;
- Peripheral capacity;
- Host computers;
- Software test computers.

Establishing the Project Plan

The plan generated for the project defines all aspects of the effort. It ties together project life-cycle considerations, technical and management tasks, budgets and schedules, milestones, data management, risk identification, resource and skill requirements, and stakeholder interaction.

Plans from other process areas may provide additional detailed guidance and should be compatible with the overall project plan in order to indicate who has the authority, responsibility, accountability, and control. These plans may be incorporated into the project plan as separate chapters or they may exist as stand-alone documents. Candidate plans that affect project success include:

- Quality assurance;
- Configuration management;
- Data management;
- Risk management;
- Measurement and analysis;
- Knowledge and skills building (training);
- Stakeholder involvement;
- Integration strategy;
- Verification strategy;
- Validation strategy.

Commitment Process Overview

The commitment process is a process that is established and maintained to ensure that commitments are made with the involvement and agreement of those who will do the work. This is a vastly different concept from what is normally stated by senior managers when they indicate they are "committed" to process improvement and product and service quality. The commitment process includes these steps presented in high-level form:

- Work breakdown structure is defined.
- An estimate is made of the magnitude of the commitment.
- The disciplines involved participate in developing the implementation plan including estimating the size, effort, and schedule.
- An independent review of the plan is held.
- Agreement is reached on the commitment through negotiation.
- Senior management approves the commitment.
- A mechanism is provided to renegotiate the commitment in the event of a requirements change.
- A postimplementation review is held to discover what went right and what went wrong to learn how to improve future commitments and to compare actual performance with original estimates. Sometimes this is referred to as a postproject review in which lessons learned are captured.

The commitment process is on the critical path for three process areas as shown in Figure 6.7.

To obtain commitment from relevant stakeholders, differences between the estimates and resources must be negotiated and reconciled. Project teams and supporting teams must have confidence that the work can be performed within cost, schedule, and performance constraints. Internal and external commitments or recommitments must be reviewed with senior management to ensure necessary support for this commitment together with all of the other existing and pending com-

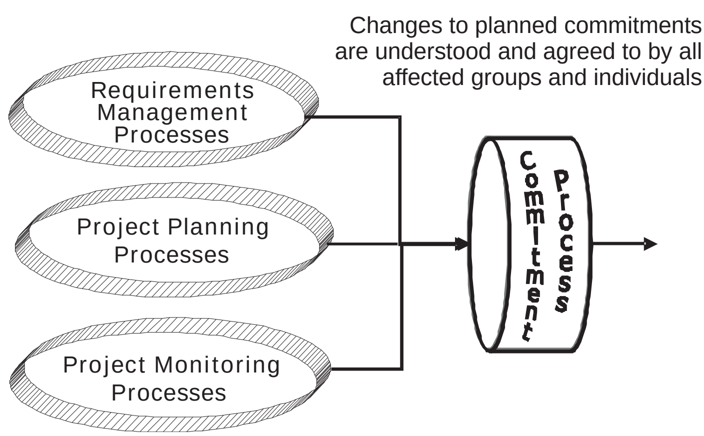


Figure 6.7 Commitment process.

mitments that have been made throughout the organization. This elaboration with senior management was expanded on in Chapter 5.

Project Monitoring and Control

The documented project plan is used as the basis for monitoring activities, communicating status, and taking corrective action. Recording actual project progress and performance must also include recording associated contextual information to help understand the measures. The purpose of project monitoring and control is to provide an understanding into the project's progress so that appropriate corrective action can be taken when the project's expected performance deviates significantly from the plan.

Criteria must be established for determining just what does constitute a significant deviation from the project plan. The corrective actions may require replanning, which might even include revising the original plan and establishing a new agreement with the customer.

Actual values of the project planning parameters together with associated contextual information must be recorded including:

- Attributes of the work products:
  - Size;
  - Complexity;
  - Weight;
  - Form, fit, or function;
- Cost and expended effort;
- Schedule;
- Technical performance (completion of activities and milestones against the schedule);
- Staffing profiles;
- Resources:
  - Physical facilities;
  - Computers and peripherals;
  - Networks;
  - Security environment;
- Knowledge and skills acquisition of project personnel (training needs);
- Estimates and actuals to be placed in an organization-wide historical database for use by ongoing and future projects.

In addition, project monitoring and control must track commitments, risks, data management, and stakeholder involvement. When the actual values recorded deviate significantly from the estimated values, some corrective action must occur. Examples of potential actions include:

- Modifying the statement of work (SOW);
- Modifying the requirements;
- Revising estimates and plans;
- Renegotiating commitments;
- Adding resources;
- Revising understanding of project risks.

Summary

Some years ago during an assessment, a gentleman being interviewed stated, "Why should we develop a project plan? It is only going to be changed anyway." In fact, one of the major reasons we develop project plans is to modify or change them when tracking indicates that we should take corrective action or when we learn something about the system we are developing that causes us to replan in some way.

Enabling the project manager to manage and control his/her project better means:

- Understanding all of the work that must be done by the project or a support group or by a supplier;
- Estimating the attributes of the system through the use of historical data, models, and project team estimates whether they are lines of code, function points, number of interfaces, or number of gates on a gate array;
- Establishing the budget and schedule that corresponds to the work and its complexity;
- Ensuring that the knowledge and skill of the project team is sufficient to produce the products or product components that have been tasked to the project;
- Ensuring that all project data, regardless of form or type, is identified and the project manager knows where the data is being stored, who will help him/her store and retrieve the data, who will perform any data analysis, and will help the project manager understand the data so that it can be used to support project management decisions;
- Ensuring that all relevant stakeholders have been identified through the project life cycle along with what their importance is and how they should be involved, and that they are involved according to plan. The project is getting from them what is needed;
- Producing all necessary plans from all involved engineering disciplines, support groups, and testing;
- Negotiating all estimates until all relevant stakeholders can commitment to them;
- Making the project commitment that also fits into the overall business commitment that the senior manager can support;
- Monitoring and controlling the plan to ensure that tasks, milestones, and commitments are met by all relevant stakeholders;

- Conducting management reviews or progress checks to ensure a successful project.

Project Planning and Project Monitoring and Control are the basic project management functions that must be understood and carried out on every project. In the Chapter 7, we examine the details of risk management.

References

[1] University of Edinburgh, “Estimation, Project Planning and Resource Booking,” section on Project and Task Estimation, Estimation Technique 1, Three Point Estimation, February 2008, http://www.projects.ed.ac.uk/methodologies/Full_Software_Project_Template/EstimationGuidelines.shtml.

[2] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI Second Edition, Guidelines for Process Integration and Product Improvement,* Reading, MA: Addison-Wesley, 2007.

CHAPTER 7

Enabling Project Managers to Manage and Control Better Through Risk Management

Risk as a science was born in the 16th century Renaissance, a time of discovery. The work *risk* is derived from the early Italian *risicare,* which means "to dare." Today, risk is defined as the possibility of loss. Unless there is a potential for loss, there is no risk.

Risks are future events with a probability of occurrence and a potential for loss. Many problems that arise in systems/software development efforts were first identified as risks by someone on the project staff. Many organizations and projects worry about risks only during "risk management season." At the beginning of a project or a new business year, management encourages the projects to identify all of the risks that they can. Risks are brainstormed without regard to type, probability, or potential loss, placed in a file, and then promptly forgotten until the start of the next "risk management season." Even when risks are identified in a brainstorming session and tracked on the project, many of the risks are not risks at all, but are already known problems.

For instance, when a project manager knows that the number of people and their level of knowledge and skills are not sufficient to satisfy the goals of the project, this is not a risk; it is a problem. Problems must be addressed. Problems are realized risks. Risks have the possibility of being managed.

Making Decisions Under Conditions of Uncertainty

Risk management is decision making under conditions of uncertainty. Robert Charette[1] described risk management as a project management activity that does not deal with future decisions, but with the future of present decisions.

Risk management (Figure 7.1) involves:

- Risk assessment through:
  - Identifying potential problems before they occur;
  - Analyzing their probability, potential impact, and time frame;
  - Risk prioritization;

1. Robert N. Charette is a Cutter Consortium Fellow, Director of the Risk Management Intelligence Network, and Senior Consultant with the Agile Software Development and Project Management Practice. With more than 25 years of experience in a wide variety of international technology and management positions, Dr. Charette is recognized as an international authority and pioneer in the areas of information systems, technology, and telecommunications risk management.

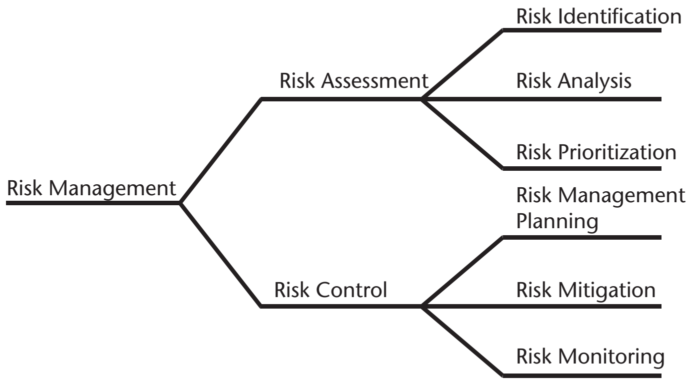


Figure 7.1 Elements of risk management.

- Risk control through:
  - Determining a risk management strategy;
  - Planning risk-handling activities to mitigate adverse impacts on achieving project and organizational objectives;
  - Determining acceptable contingency plans;
  - Implementing risk mitigation plans and possibly contingency plans when the risks approach the predefined thresholds;
  - Proactively tracking and managing the risks.

Another way to view risk management is through the risk management cycle taken from Shewhart's PDCA cycle (Figure 7.2). Risk management is not done once per project and then forgotten; it is a continuous activity that changes as the challenges facing the project are met or lost, or as the status of known risks changes, or as new risks appear that affect past risk mitigation planning.

Project managers *do* manage risks, but they tend to:

- Manage the risks they see, and don't often see all of the risks, or fail to recognize the critical risks;

Figure 7.2 Risk management cycle. (*After:* [1].)

- Manage only those risks for which they have domain expertise;
- Manage cost and schedule—the symptoms, not the risk itself.

In addition, project managers are usually selected and rewarded for their "crisis management skills."

Although the Risk Management PA is new process to the CMMI®, risk management is certainly not a new topic. Its benefits to the project and organization and to the individual people who are allowed to share their ideas of risk are significant. Risk management:

- Facilitates a closer look at strategy and decision-making criteria;
- Creates a shared product vision;
- Facilitates upward communication of risks;
- Facilitates downward communication of strategy and decision-making criteria;
- Helps project members to:
    - Take a closer look at their plans, procedures, and processes compared to the business needs and possible problems;
    - Think of alternative approaches to problem solving;
    - Determine backup plans in the event of problems;
    - Prevent problems before they occur.

To prepare for risk management, a project leader and his/her project members need to determine risk sources and categories or "bins" for collecting and organizing the risks, define risk parameters, prioritize those risks, and then establish a risk management strategy.

Sources of Risk

There are many sources of risks both internal and external to the project. Many of these sources of risk are "just accepted" without adequate planning to reduce the probability or impact of the risk. Typical sources of risks include:

- Poorly defined requirements;
- Interface requirements;
- Performance requirements;
- Requirements stability;
- Unavailable technology;
- Unrealistic schedule estimates;
- Inadequate knowledge and skills of staff;
- Inadequate test and evaluation opportunities;
- Inadequate supplier capability;
- Safety, security, and health issues.

One of the first places for a project to look for hints of risks is to examine the WBS and then branch to the individual activities. It should be clear that one of the most important sources of risk to recognize and control is an organization's suppliers. I found myself in a situation once where my client stated another process improvement consultant had advised them that because they were only trying to control 20 suppliers, they really did not have to worry about risk management. I advised this client to gain as much capability in risk management as it could if it truly had to control 20 suppliers and ensure that all of the subsystems delivered could indeed be integrated and function properly.

Risk Parameters

It is important for the project team to have a clear understanding of the parameters used to categorize risks in order to develop a mitigation strategy and manage these risks throughout the project life cycle. Standard parameters for categorizing risks include:

- *Likelihood or probability* that the consequence will occur;
- *Consequence, impact, or potential loss* (consequences are generally related to costs, schedules, environmental impact, or human measures such as labor hours lost and severity of injury);
- *Time frame* that the risk might occur during the project/product life cycle;
- Thresholds or control points to trigger management actions (define the early warning signals that the risk is becoming serious).

It is very important for any project to understand risk in terms of quantitative data. For example, it would be the best if we could determine that the probability of a risk occurring is 80% or the business impact is $2 million. It would also be great if we could say with certainty that the risk will occur in the sixth month of the project. Unfortunately, projects do not normally have such quantitative data when a risk management culture is being established throughout the organization. To get things started, it is recommended that relative terms be defined and used for those parameters.

Examples for likelihood are:

- Remote;
- Unlikely;
- Likely;
- Highly likely;
- Near certainty.

Examples for consequences are:

- Low, medium, high;
- Negligible, marginal, significant, critical, catastrophic.

Examples of time frames are:

- Short term;
- Medium term;
- Long term.

Although absolute values may not be available in the early stages of establishing a risk management function throughout an organization, it is important for all projects to either use organizational definitions or develop project definitions that point to a range of values or attributes that can be assigned to these relative risk terms. It is important for all relevant stakeholders to be able to effectively plan for both what and how to handle identified risks.

For example, if we use the categories of risk parameters stated earlier, we might start with providing percentage intervals for likelihood such as:

Remote	0% to 20%;
Unlikely	21% to 40%;
Likely	41% to 60%;
Highly likely	61% to 80%;
Near certainty	81% to 100%.

With a focus on consequences; we might have to provide scenarios to describe the "ranges of consequences." For example, what is a "catastrophic" risk? In one company, it might mean making an important client angry. In another company it might mean losing a share in the market place, while for a third company, a "catastrophic" risk might mean losing human life. As long as the relative definition has an agreed upon substance behind it, it can work to get risk management institutionalized throughout an organization.

To be complete we should examine time frames. What does short term, medium term, or long term mean? In some companies, short term could mean a month or even weeks. One company, with whom I consulted years ago, had projects defined in terms of 8 hours. A company that thinks of short term as 1 month might think of medium term as 5 or 6 months and long term as 1 year. But for companies that are used to 12-year projects, short term might be thought of as 3 years or less; medium term, 6 to 9 years; and long term 10 to 13 years. The effectiveness of using relative terms as risk parameters is dependent on a common understanding of what they mean until the organization can turn those relative terms into quantitative terms.

Time frame might deserve a few more words in order to encourage companies to factor this parameter into its risk management thinking. Time frame is normally thought of as when the risk might occur during the defined and planned project life cycle or even during product life cycle. We can take a real-life example that might make this concept a bit sharper. Figure 7.3 shows a picture of a tornado that was photographed in Salt Lake City during an SSTC conference being held there.

Perhaps the reader has had the opportunity to see the Hollywood movie, *Twister*. Besides the cow and the house rolling across the street, the movie depicted people and events that were very real. I grew up in Lawton, Oklahoma, which is part of the tornado belt. There is a real tornado season that occurs every year when the warm air from the Gulf of Mexico comes into conflict with the cooler air from Canada. This season is approximately from late March until late May or early June.

Figure 7.3 Tornado sighting.

If a person from the East Coast were to move to Oklahoma 1 month after tornado season were over, the probability that a tornado would appear in the next year would be 100%, the consequence, possible death, but the time frame would cause the priority assigned to the possibility of a tornado appearing to be very low. If, on the other hand, a person were to move to Oklahoma 1 month before tornado season were to start, the probability a tornado would appear in the next year would be 100%, the consequence, possible death, and as folks say in those parts, "You are in a heap of trouble, boy!" The time frame can have a dramatic effect on calculating the priority of when and how to mitigate the risk.

Prioritizing Risks

One of the most common ways to prioritize risks that I have seen is the practice of allowing the individual with the most seniority and/or loudest voice to decide what the priority of the risk should be. Many companies identify numbers for both probability and consequence, and then multiply those numbers together to arrive at a value that they use to prioritize the risks. Using numbers as relative terms is acceptable as long as some description of that relative term is documented, as described earlier. Another way is shown in Figures 7.4 and 7.5. The technique is to use a three-dimensional matrix multiplication of the probability, consequence, and time frame. First, we multiply the probability values by the impact. This could yield a matrix result as shown in Figure 7.4. We then multiply those values by the time frame as shown in Figure 7.5.

Some clients use criticality to break the tie. This method is not meant to be taken as absolute, but it has proven effective on three continents and is used at the univer-

Probability: / Impact:	Probable	Possible	Unlikely
Catastrophic	Top	High	Serious
Critical	High	Serious	Medium
Marginal	Serious	Medium	Low
Negligible	Medium	Low	Bottom

Figure 7.4 Prioritizing risks 1: probability multiplied by impact or consequence.

Time Frame:	Immediate	Short	Medium	Long
Top	9	8	7	6
High	8	7	6	5
Serious	7	6	5	4
Medium	6	5	4	3
Low	5	4	3	2
Bottom	4	3	2	1

Figure 7.5 Prioritizing risks 2: results multiplied by time frame.

sity level in Europe. This is also an example of a decision analysis and resolution (DAR) method.

Risk Management Strategy

A risk management strategy needs to be established and maintained to ensure that any risk mitigation activities that are put in place will be cost effective and support business objectives as well. The risk management strategy should include objectives, constraints, and alternatives. For those risks that exceed established thresholds for risk exposure, the project team needs to develop a risk mitigation plan (i.e., a set of activities to reduce the probability of the risk occurring) or reduce the impact of the occurrence of the risk. Some risk mitigation activities include:

- Change control mechanisms to monitor risk areas;
- Consider alternative designs;
- Provide additional training;

- Provide cross-training to ensure each function is backed up;
- Involve users, such as focus groups, more;
- Increase peer reviews such as inspections and structured walkthroughs;
- Develop and use a traceability matrix;
- Increase the level of testing and auditing of testing results;
- Provide additional time and budget;
- Use prototyping;
- Use simulation;
- Search for higher performance hardware;
- Follow an incremental development or evolutionary development approach.

The risk management strategy is necessary in order to support the project manager to determine risk mitigation activities that will:

- Provide the greatest reduction in risk;
- Require the fewest resources;
- Require available resources;
- Have the least impact on the schedule.

The question that requires an answer is "What set of strategies best manages the project's risk?" Example strategies commonly used include:

- *Acceptance.* Accept the consequences of the risk (do nothing). Make a conscious decision to live with the risk, having determined that the mitigation effort would be more expensive than the problem.
- *Avoidance.* Eliminate the risk altogether in order to avoid a "lose–lose" situation (e.g., decision not to bid on a request for proposal). Change or lower the requirements while still meeting the user's needs.
- *Diversification.* Do not rely entirely on one supplier, one product, or one point of failure.
- *Reduction.* Decrease the risk through mitigation, prevention, and anticipation. Reduction can be applied to either the probability or the consequences.
- *Elimination.* Eliminate the risk completely whenever possible based on the cost of the problem and the cost of the solution.
- *Protection.* Employ redundancy to mitigate the risk (e.g., two systems backing up each other).
- *Reserves.* Use contingency funds and build into the schedule slack time to cover uncertainties.
- *Transfer.* Shift the risk to another person or group better able to act on it.

Along with the risk management strategy, other risk management concerns to manage the project risks include defining the scope of the risk management effort; identifying the methods and tools that will be used for risk identification, risk analysis, risk mitigation, risk monitoring, and communication; determining how the risks will be organized, classified, bounded, and consolidated; and identifying the risk mitigation techniques to be used such as prototyping, simulation, or evolutionary development.

Risk Management Plan

Risk management planning is the function of deciding what, if anything, should be done about a risk or set of related risks. A risk management plan is prepared by the project manager and the project team to foresee risks and create response plans to mitigate them. The risk management plan should contain the analysis of the risks including likelihood, potential impact, and time frame. Mitigation strategies, such as those described earlier, must be defined and agreed to in order to help the project avoid being pushed off course should the risk become a problem.

Most importantly, any risk management plan must include the risk strategy that is acceptable to the project and organization so that the rest of the elements of the risk management plan can be built on it.

Based on the risk management strategy adopted by the project or business unit, a risk management plan should be developed for the most important risks to the project. Suggested risk plan content includes:

- *Risk description or risk statement*. This should be written in clear risk management language to make it obvious to all readers why this risk is an important risk to list and manage for the project and possibly organization.
- *Source or cause of risk*. Both external and internal. Ensure that all suppliers are examined as potential sources of risk.
- *Point of contact for details of identified risk*. Who is responsible for identifying the risk and formulating it so it can be managed?
- *Date identified as a risk*. Calendar date with a clear indication of when in the project life cycle the risk was identified.
- *Risk probability*. The likelihood the impact or consequence will occur if the risk becomes a problem. Remember, initially the probability may have to be expressed in relative terms such as remote, unlikely, likely, highly likely, or near certainty along with what the meaning is for those terms for the project and organization.
- *Consequence or impact*. This is the potential loss or impact to cost, schedule, human measures, and so forth. The impact may also have to be expressed in relative terms at the start, but a range of impact values must be agreed on and documented for all to have the best understanding of why this risk must be managed.
- *Time frame in which the risk might occur during the project/product life cycle*. Again, this may start out to be a relative term such as short term, medium term, or long term. Ranges of values should be associated with these relative terms until more quantitative data can be obtained.
- *Thresholds or control points*. These trigger management actions (define the early warning signals that the risk is becoming serious).

- *Classification of risk*. Is the risk associated with management, technical issues, process, suppliers, the environment, or so forth?
- *Person or team assigned to handle risk*. Which person or persons from the project team or support groups have been assigned to put the risk mitigation in place if the risk nears the threshold and starts to become a problem?
- *Product components and parameters*. Product components that may be affected by the risk as well as the project management parameters.
- *Possible mitigating actions*. The mitigating activities that will be put in place in an attempt to reduce the probability or impact within the time frame in which the risk is supposed to occur. Remember this does not mean that the risk must be mitigated to zero—that may be too expensive and time consuming. Each alternative mitigation action along with its possible reduction effect should be included in the mitigation actions. Remember as well that many risk mitigation actions need not be complicated engineering feats but actions that help the risk avoid becoming a problem or issue that must be dealt with. It may be necessary to use a formal decision process when developing a risk mitigation plan.
- *Contingency plan*. Contingency plans are normally developed for selected critical risks in the event that if the risk cannot be mitigated with the techniques chosen, a Plan B or alternative course of action can be taken. The reader may want to consider two popular definitions for contingency planning. The first one is that a contingency plan suggests activities that must be done when the risk has crossed the risk threshold and has become a problem or issue. The second definition of a contingency plan is a second, but normally more expensive mitigation plan to try to reduce the risk before it becomes a problem. It must be understood that this second definition of a contingency plan is not simply a second chance, but one that has been shown to be cost effective albeit with a lower profit margin or a necessary choice due to strategic organizational or project reasons.

Figure 7.6 shows the relationship between risk strategies, risk mitigation and contingency plans, and the overall risk management plan.

Risk Monitoring

Risk monitoring should be included with the other standard project management monitoring and control activities that were discussed in Chapter 6. It is imperative that the thresholds that define when a risk becomes unacceptable and triggers the execution of a risk mitigation plan or a contingency plan be determined and used during risk monitoring. Figure 7.7 illustrates the establishment of a risk management strategy, risk identification, determination of risk mitigation techniques that will be used, determination of contingency plans prior to project start, and risk monitoring using those established thresholds.

Note that risk monitoring will probably result in the identification of new risks, the setting of new thresholds, and challenges to the previously agreed–on risk mitigation activities. It is the author's recommendation that a Top 10 risk list be devel-

Figure 7.6 Risk management plan.

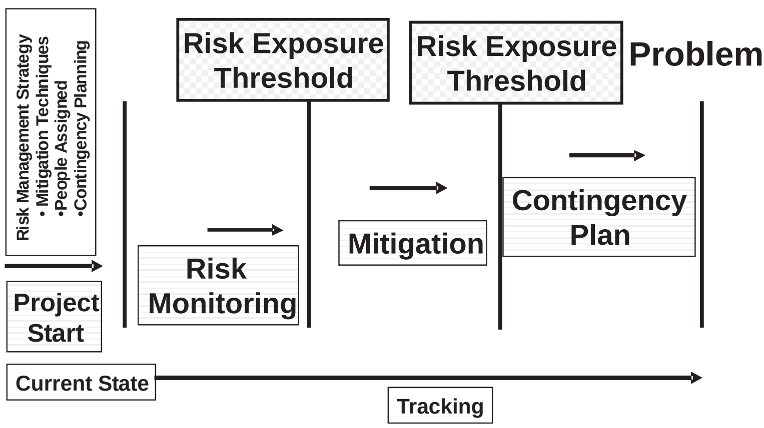


Figure 7.7 Establishing risk thresholds.

oped for each project that is constantly monitored and updated as necessary. As the project moves through the project life-cycle phases, this Top 10 risk list should reflect the success or failure of the risk mitigation and contingency planning made so far and result in an updated set of risk items with adjusted priorities and possibly new risk mitigation and contingency planning techniques.

The risk status report may be a part of the standard project management status report or may be a separate reporting item. Suggested contents include:

- Top 10 risk items;
- New risk items since last report;
- Number resolved (i.e., successfully mitigated, avoided, or had impact reduced to lower priority);
- Number of contingency plans that had to be invoked;

- Risks that became problems;
- Time histories of above;
- Status to existing risks:
    - Risk statement;
    - Old and new priority;
    - Old and new probability and impact;
    - Old and new responsibility;
    - Reason for change.

Summary

Risk management is about making informed decisions under conditions of uncertainty. Risk management involves:

- Identifying potential problems before they occur;
- Analyzing the probability, potential impact, and time frame;
- Determining a risk management strategy;
- Planning risk-handling activities to mitigate adverse impacts on achieving project and organizational objectives;
- Determining and evaluating contingency plans;
- Proactively tracking and managing the risks.

Effective risk management depends on open communication with all of the project's relevant stakeholders throughout the project life cycle. Communication is an essential part of project management. Without effective communication, process improvement cannot be realized. The SEI has recognized this and has provided the exposure in the CMMI® model that it deserves.

Reference

[1] Shewhart, W. A., *Statistical Method from the Viewpoint of Quality Control,* Washington, D.C.: Graduate School of the Department of Agriculture, 1939.

CHAPTER 8

Enabling Project Managers to Manage and Control Better Through Quality Management

As mentioned in Chapter 6, the concepts of project management are being presented in Chapters 6 through 10. The quality management process areas of Process and Product Quality Assurance and Configuration Management are now described in this chapter as project management functions that provide input to project managers to help them manage and control better and not simply go through the motions to satisfy audit or assessment criteria.

Process and Product Quality Assurance

Quality assurance should begin in the early phases of a project to establish plans, processes, standards, and procedures that will:

- Add value to the project;
- Satisfy the requirements of the project and organizational policies.

Quality Assurance Responsibles should participate in the establishment of those plans, processes, standards, and procedures to ensure they will fit or can be tailored to fit the project's needs and they can be audited or evaluated against throughout the project life cycle.

There are two types of representations in CMMI® models: staged and continuous. A representation in CMMI® is analogous to a view into a data set provided by a database. Both representations provide ways of implementing process improvement to achieve business goals. Levels are used in CMMI® to describe an evolutionary path for an organization that wants to improve the processes it uses to develop and maintain its products and services.

Within the continuous representation, capability levels are used to determine an organization's process improvement achievement in individual process areas such as Requirements Development or Configuration Management. Within the staged representation, maturity levels are used to determine an organization's process improvement achievement across multiple process areas that have been predefined for each of the five maturity levels. Process and Product Quality Assurance (PPQA) and Configuration Management (CM) appear in the special category identified as

Maturity Level 2 that makes up the staged representation of the CMMI-DEV v1.2. From the continuous representation point of view, PPQA and CM are part of the category called Support [1].

Although the continuous representation of the CMMI® depicts a categorization scheme that places CM and QA in the category of Support, it is the author's experience that effective use of the engineering principles of CM and QA are best realized by thinking of them as project management functions. This section of Chapter 8 focuses on the support a project manager can and should expect from the functions of process and product QA.

Quality Control

Process and product quality assurance is often misunderstood or purposefully equated to testing. Perhaps it is important to first distinguish between *quality control* and quality assurance. Quality control evaluates or checks the quality of the products and life-cycle work products during their development. Quality control functions or activities help to determine if the product is within defined tolerances and of acceptable quality. Tools and techniques used for quality control include peer reviews, such as inspections or structured walkthroughs, and the different levels of testing. Peer reviews and most testing techniques are described in the process areas of Verification and Validation.

Quality Assurance

Quality assurance, in contrast, evaluates or checks to see if the process is working. Is the process being followed? Are the quality control checks being applied with the proper rigor? Are the quality control checks efficient and effective? Is the process causing quality problems? Is the process working for the organization? Tools and techniques used by QA Responsibles include process reviews to verify whether the processes and procedures developed are both usable and are in fact actually used!

The purpose of process and product quality assurance is to provide management at all levels and practitioners with objective insight into the processes that are in place and identified to be used on the projects. Objective evaluations are used to determine if the processes are indeed being followed on the projects and, if they are, to answer these questions: Are they efficient, are they effective, and are they enabling the project members to produce the required product quality?

Quality Functions

To be compliant with the requirements and guidelines of the Process and Product Quality Assurance PA, each project is expected to develop a project quality plan to document the quality functions that will be needed to support the project throughout the life cycle. Quality functions are very important, and yet are typically overlooked in their relative value. These quality functions include, but are not limited to:

- Setting quality goals for the project that support the organization's business objectives;

- Conducting peer reviews and technical reviews throughout the product life cycle;
- Performing multiple levels of testing such as unit testing, integration testing, systems testing, and acceptance testing;
- Designing in quality factors such as maintainability, expandability, reliability, safety, and security;
- Conducting objective evaluations with respect to product quality;
- Conducting objective evaluations with respect to process quality;
- Conducting objective evaluations of customer and maintenance documentation;
- Conducting objective evaluations of CM activities;
- Providing visibility into the process and product quality for management and practitioners through quality reporting;
- Getting noncompliance issues resolved before the product is delivered to the customer;
- Identifying measurements that support the information needs of the project and organization;
- Conducting performance evaluations to ensure the system converges to established performance constraints;
- Conducting appropriate Verification PA functions;
- Conducting appropriate Validation PA functions.

These quality functions may be performed by:

- Project managers;
- Product and product component developers (SW, EE, ME, optics, and so on);
- Quality managers or QA Responsibles;
- Configuration managers;
- Organizational-level quality assurance group;
- Systems engineering;
- Integration and systems test;
- Documentation;
- Database;
- Others.

Project Quality Plan

In summary, given that a project development plan exists, a project quality plan must answer these questions:

- What quality functions will be performed?
- Who will perform them?
- During what phase of the product life cycle will they be performed?
- Who has approval authority?

- How will conflicts over nonconformance be resolved?

Additional questions should be asked and answered in the project quality plan:

- What peer reviews will take place and when?
- How will the data from the peer reviews be utilized?
- Which tests will be conducted and by whom?
- Which tests will a QA Responsible observe?
- What objective evaluations will a QA Responsible conduct?
- What metrics will be used for the capture and analysis of identified defects?
- How will the correction of the discrepancies be assured?
- What are the criteria for the acceptance of the product from a quality point of view?

Because the project manager is ultimately responsible for the product quality produced by his/her project members, it is important that the project leader work with the QA Responsibles supporting the project to develop and manage this project quality plan.

Note that the project quality plan is *not the same* as the quality assurance plan that may be developed by the quality assurance group documenting how the QA Responsibles will support the project with their advice and quality evaluations.

The consequences to a project and ultimately the business objectives of an organization if projects do not develop and follow a project quality plan are as follows:

- Qualify functions may be left out.
- Quality criteria will be forgotten or ignored.
- Interfaces may not work.
- Process steps may be ignored.
- The product or product component may not match the customer's requirements and expectations when it is delivered.
- Problems take a long time to discover and fix.
- The resulting *rework* is expensive.
- Development and production times lengthen.
- The delivery date is delayed.
- The business unit can incur financial penalties.
- There is risk of losing business and even the customer.

Quality Assurance Responsibles

The QA Responsibles should be providing consultation and objective evaluation of the project's plans, processes, standards, procedures, guidelines, templates, forms, and checklists with regard to:

- Compliance with the organizational policies;
- Other standards;

- Compliance with externally imposed requirements, standards, and procedures required by the customers;
- Processes, standards, and procedures that are appropriate for use by the project;
- Required knowledge and skills of the staff;
- Training needs;
- Historical data.

Project leaders should be able to expect the following support from quality assurance to help them manage and control their project better:

- Knowledge about the product life-cycle processes;
- Input as to the efficiency of the process being used by project members;
- Assistance in creating an executable and successful project plan;
- Assistance in creating and developing the project's quality plan;
- Assistance in choosing the right standards for the project's needs;
- Assistance in tailoring the standards and processes for practical use by the project based on the organization's tailoring guidelines;
- Assistance in setting up peer reviews for the product life-cycle work products;
- Assistance in identifying and monitoring risks;
- Assistance in putting together the right quality plan to match the criticality of the life-cycle work products;
- Performance of objective evaluations (process and work product compliance);
- Performance of requirements traceability audits to ensure that the quality goals are being met and the system's integrity is being maintained;
- Assistance in verifying that configuration audits have taken place.

Objective Evaluation

Objective evaluation in quality assurance evaluations is critical to the success of the project. Objective evaluation:

- Provides the QA Responsibles with the organizational freedom to be representatives of management on the project;
- Protects the QA Responsibles from adverse actions by the project managers such as loss of job, pay, or position;
- Provides management with the confidence that the objective information about the activities and work products of the project is being accurately reported;
- Ensures that everyone performing the quality assurance activities is trained in quality assurance/quality management concepts;
- Ensures that those designated to perform the QA activities are separated from those directly involved in developing or maintaining the work products;

- Provides an independent reporting channel to the appropriate level of organizational management to allow noncompliance issues to be escalated to the appropriate levels of management as necessary.

Objective evaluation applied to performed processes as defined or referenced in the project plan against applicable process descriptions, standards, procedures, and so on, should be conducted based on clearly established and maintained criteria and business needs and answer the following questions:

- What will be evaluated?
- When or how often will a process be evaluated?
- How will the evaluation be conducted?
- Who must be involved in the evaluation?

Objective evaluation applied to the project's work products and services looks for compliance against the applicable process descriptions, standards, and procedures. The definition of a work product in the CMMI® Product Suite is a tangible artifact that is a useful result or by-product of executing a process. This can include:

- Files;
- Documents;
- Products;
- Parts of a product (components);
- Services;
- Process descriptions;
- Meeting minutes;
- Agendas,
- Specifications;
- Peer review records;
- Test records;
- Invoices.

The purpose of these objective evaluations is to provide the project manager and project members with information that will help the project manager to look into process and product noncompliances, and then to implement corrective action as soon as possible in order to reduce the quality management and project management risks.

Quality reports in the form of evaluation reports, corrective action reports, and quality trends should be tracked, openly communicated to all relevant stakeholders in a timely manner, and resolved. Noncompliance issues must be resolved at a level as close as possible to the source of the issue. Quality assurance loses perceived value instantly if the QA Responsibles immediately report their noncompliance findings to higher level management without giving the project members or project leader a chance to respond to the issues. Noncompliance findings should be first addressed at the lowest levels capable of resolving them. Noncompliance issues should be analyzed to determine if there are any quality trends that should be discussed with the project leader that might motivate preventive actions being put in place or, if need be, escalated to senior management.

To be complete, trends discovered from analysis of quality reports and documentation of the process and product quality assurance activities should be recorded in sufficient detail so that the results can be made available to and understood by all relevant stakeholders that are concerned with product quality.

Quality Assurance Group

There are many possibilities to setting up a quality assurance or a more overarching quality management organization. One in particular that has proven popular for many different types of organizations and in many different countries is described here:

- A centralized quality management group is established at the organizational level and is headed up by a middle-senior manager. This is a very important concept because too often companies focusing on achieving CMMI® ML 2 overlook that there are QA activities that must be performed outside of the project!
- The quality engineers that serve in this organizational quality management group are individuals that have between 10 and 20 years of experience, including development and project management experience.
- There are normally about 1.5% to 2% of these highly qualified quality engineers compared to the total development staff. One financial organization in the Netherlands had approximately 8 senior quality engineers compared to 600 software developers.
- Each project of medium to large size is required to nominate at least one project quality assurance (PQA) coordinator. This person does, in fact, report to the project manager but is only responsible for ensuring that the necessary quality functions for the project are completed.
- The quality engineers mentor and coach the PQA coordinators on a regular basis.
- The quality engineers also support the quality directives of the organization by representing the independent and objective point of view on process and product quality.
- When necessary, the quality engineers will confront the project manager and escalate any serious noncompliance up to the highest management level in the organization.
- Once per month the quality engineers meet with all of the PQA coordinators to discuss quality processes and procedures. Presentations are made on a selected quality topic. Approaches to dealing with difficult project situations regarding quality are discussed. Expert consulting is brought in periodically to address this forum and provide CMMI® interpretation and quality management guidance.

- Once per month, the quality engineers meet with the project managers to discuss what quality support they need, the responsiveness of the PQA coordinators as well as their own responsiveness, and process improvements that could be made to assist the project in producing higher quality products and services.

Figure 8.1 illustrates the type of quality management organization that I have set up in the United States, Europe, and Asia with high success due to the factors mentioned earlier.

Configuration Management

The purpose of configuration management is to establish and maintain the integrity of the work products using configuration identification, configuration control, configuration status accounting, and configuration audits throughout the product life cycle. Configuration management is focused on the rigorous control of the managerial and technical aspects of the work products, including the delivered system.

Integrity

Webster's dictionary defines integrity to be "the quality of state of being unimpaired"; wholeness, completeness, constancy, unfragmented.

To more completely understand the concept of integrity with regard to configuration management, let us explore further. If a system exhibits integrity we can expect to see the following things happen:

- Changes to any configuration item within the system are only made according to an established and maintained process.
- The system is secure from misdirected developers who seek to circumvent the rules.

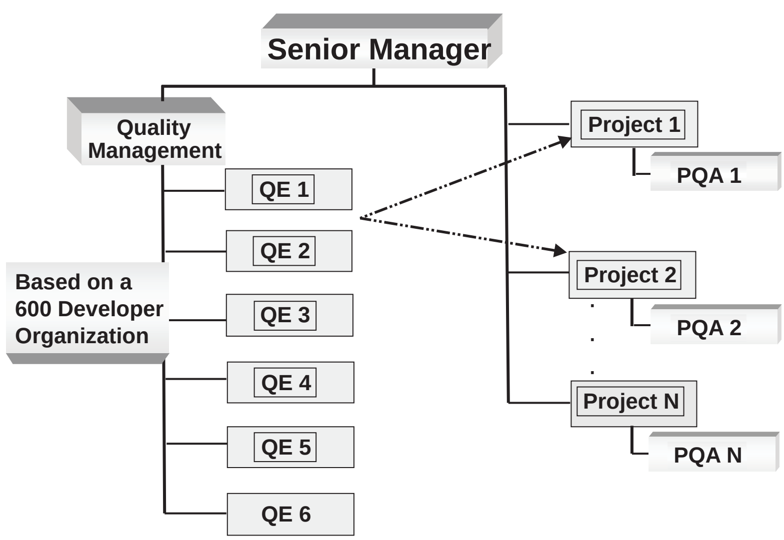


Figure 8.1 Quality management structures with an organizational and project focus.

- The system is secure from hostile attacks that threaten to damage the contents of the configuration items.
- Life-cycle work products are kept consistent when requirements change requests are approved and the requirements specification is modified. All related life-cycle work products are reviewed to determine if accompanying changes to them are necessary.
- Periodic audits are made on the contents of the system to ensure that changes made to product components are complete and correct.
- Regression testing is conducted to ensure that defects are corrected and existing functionality remains.

The need for configuration management of software components can be used to illustrate the importance of this project management support function. The most frustrating software problems are often caused by poor configuration management issues such as these:

- The latest version of source code cannot be found.
- A difficult bug that was fixed at great expense suddenly reappears.
- A developed and tested feature is mysteriously missing.
- A fully tested program suddenly does not work.
- A configuration identification system is not established.
- The wrong version of the code was tested.
- There is no traceability between the software requirements, documentation, and code.
- Programmers are working on the wrong version of the code.
- The wrong version of the configuration items is being baselined.
- No one knows which modules comprised the software system delivered to the customer.

Configuration management functions include:

- Configuration identification;
- Baselining;
- Change control;
- Use of a CM system;
- Use of CM status accounting;
- Configuration auditing;
- Interface control;
- Control of supplier CM functions;
- Release.

Let us examine these functions as they are viewed in the CM process area and in related process areas of the CMMI®.

Configuration Identification

Configuration identification includes the selection, creation, and specification of:

- The products that are delivered to the customer;
- Designated internal work products;
- Acquired products;
- Tools;
- Other items that are used in creating and describing these work products.

Configuration items to be controlled come out of the product life cycle that is chosen for the project and the product architecture as illustrated in Figures 8.2 and 8.3, respectively.

Configuration items that should always be considered for control include:

- Requirements specifications;
- Interface specifications;
- Architectural specifications;
- Design specifications;
- Code modules;
- Test plans;
- Test procedures;
- Project plan;
- Quality plan;
- Configuration management plan;
- Risk management plan;
- Data dictionaries.

Baselining

Change is a fact of life in product development. Customers want to modify requirements. Developers want to modify the technical approach. Management wants to modify the project approach. New technological developments introduce new and better materials. Modification is necessary because, as time passes, all parties know more about what they need, which approach would be best, and how to get it done and still keep within the range of constraints. This additional knowledge becomes the driving force behind most changes.

The fundamental success of any development effort depends on well-defined reference points against which to specify requirements, formulate a design, and specify changes to these requirements and the resulting designs. The term *baseline* is normally used to denote such a reference point. A baseline is an approved snapshot of the system at appropriate points in the development life cycle such as the functional baseline, allocated baseline, and product baseline commonly used for systems engineering. A baseline establishes a formal base for defining subsequent change. Without this line or reference point, the notion of change is meaningless.

A baseline could be:

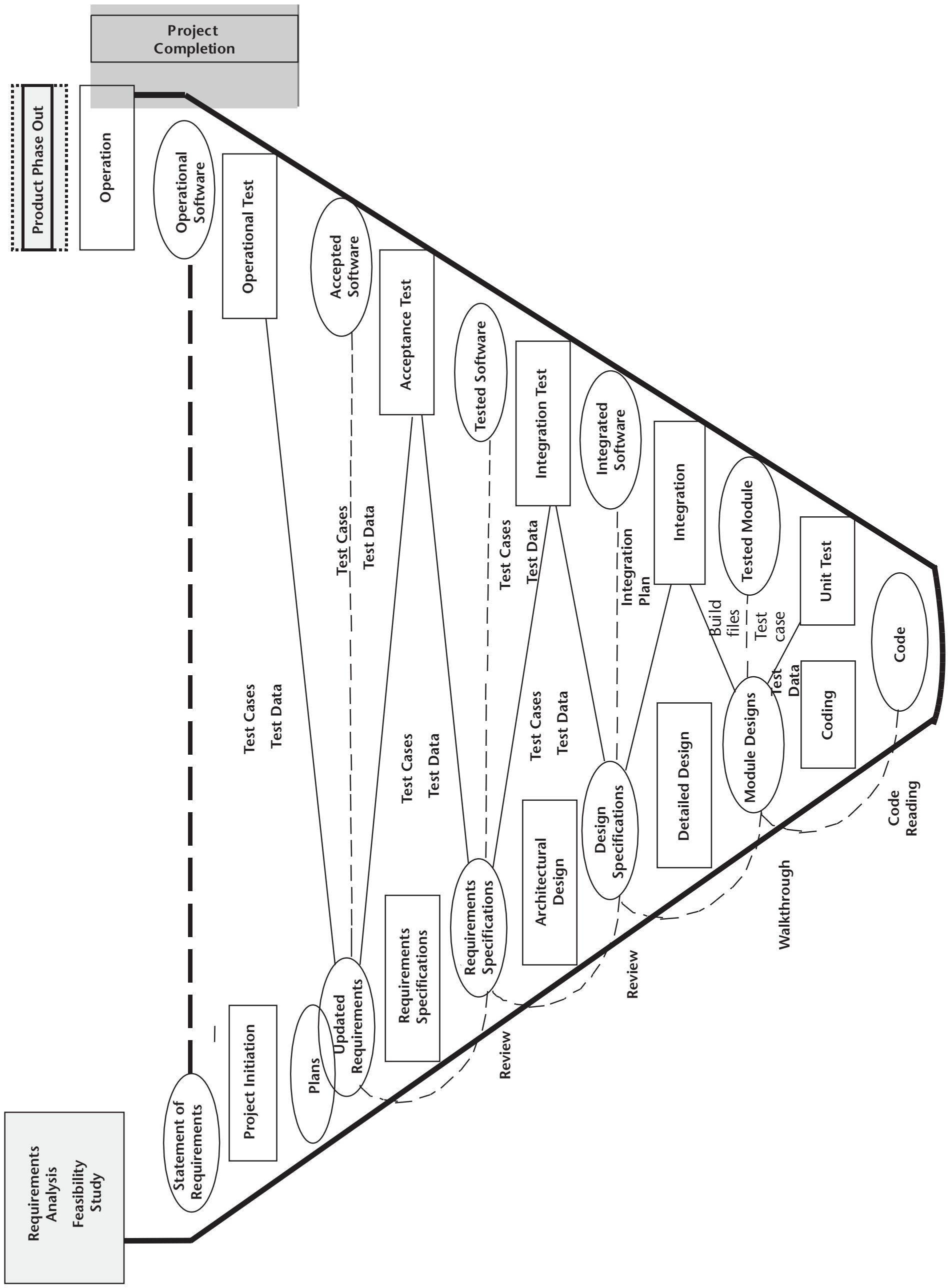


Figure 8.2 V-model development life cycle.

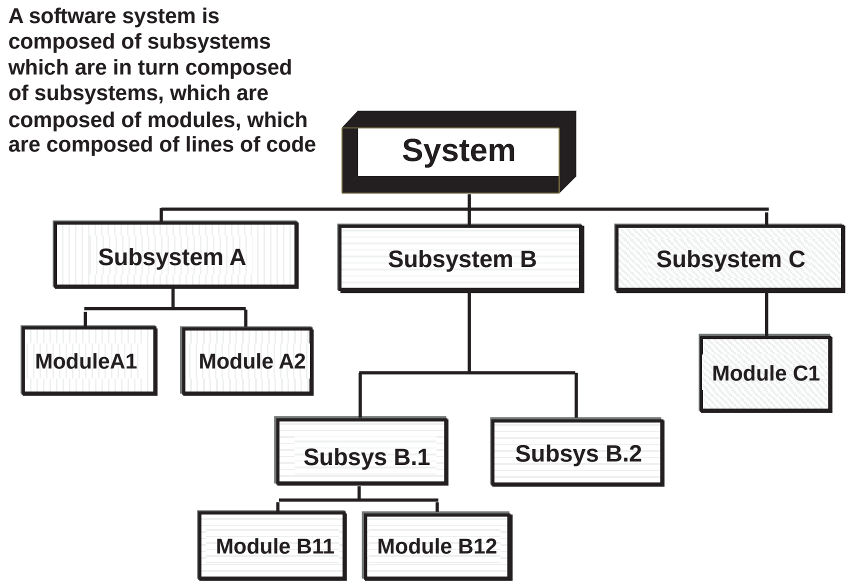


Figure 8.3 Product structure.

- A specification (e.g., requirements specification, design specification);
- A document that has become approved and released as version 1;
- A product that has been formally reviewed and agreed on;
- A partial system.

A baseline is a "record of a contract" and serves as the basis for further development. It should be changed only through an agreed-on change procedure. A baseline helps a project to control change without seriously impeding justifiable change. It will help a project to control the identified configuration items but not constrain early development excessively from the aspects of time, money, or resources. Before a baseline is established, change may be made quickly and informally. Once a baseline is established, change can be made, but a specific, formal procedure must be applied to evaluate and verify each change. The items in the baseline are the basis for the work in the next phase of the development cycle. The items of the next baseline are measured and verified against previous baselines before they become baselines themselves.

An example of a baseline is an approved description of a product that includes:

- Internally consistent versions of requirements;
- Requirements traceability matrices;
- Discipline-specific items;
- End-user documentation.

Multiple baselines may be used to define an evolving product during its development cycle. Figure 8.4 shows a common set of baselines that includes the system-level requirements, system-element-level design requirements, and the product definition at the end of the development/beginning of production. These are com-

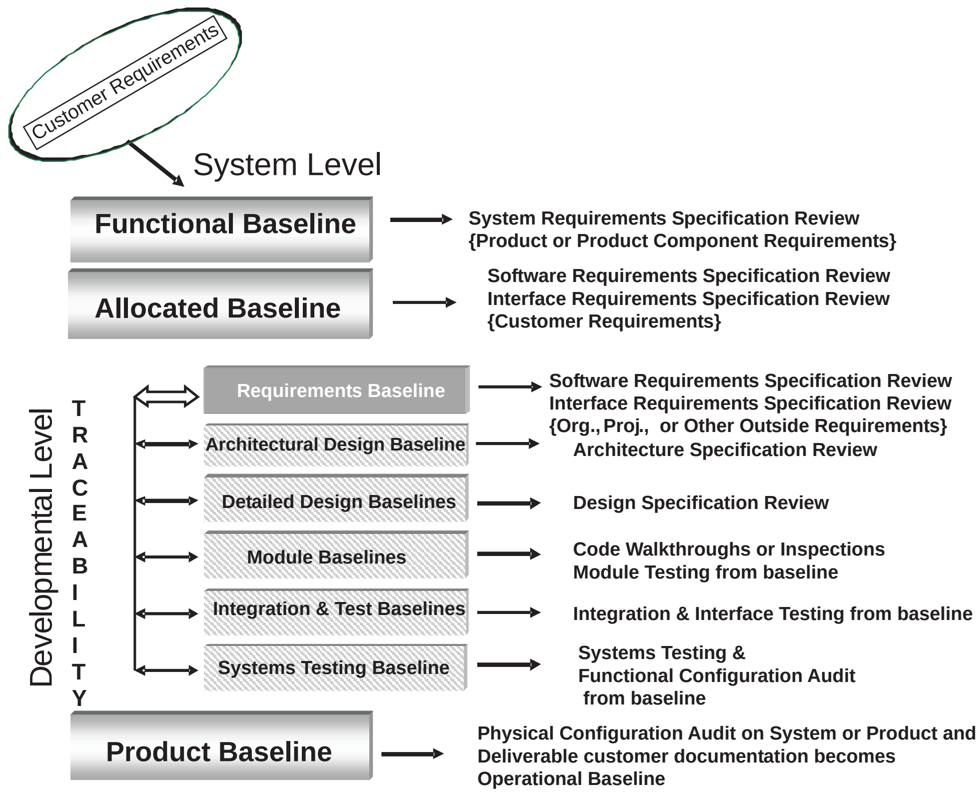


Figure 8.4 Mapping of system and developmental baselines.

monly referred to as the functional baseline, allocated baseline, and product baseline. They are also referenced as organizational baselines. Baselines that are created under the auspices of the project manager during the development of the product components are referred to as developmental or project baselines.

A baseline that is delivered to a customer is typically called a *release,* whereas a baseline for internal use is typically called a *build.*

Baselines of configuration items should be created or released from the CM system only with authorization from the configuration control board at the organizational level. Change control boards for organizational and developmental baselines along with the authority to create or change those organizational or developmental baselines are discussed following the next section.

Change Control

In an ideal world, once a configuration item was fully approved there would be no need to change it. In the real world, new versions of a configuration item are needed for a variety of reasons. The requirements for the system may change as the customer and the project obtain a better understanding of what the system is supposed to do. The boundaries between items in the design hierarchy may change, and the precise interface between those items may need to be renegotiated. The specification of an item may be incomplete or could have been wrongly interpreted. An error may be found that was not detected during the review of the configuration item. A software/hardware environment may evolve in a way that necessitates a change to the existing system. In each case, a new version of configuration items is needed that supersedes the earlier version. Replacing one version of a configuration item with a better one is the objective of all change.

Without effective change control, an engineer (software, hardware, systems) could make an important change to a configuration item or its interfaces without a lot of extra work and red tape, but unfortunately no record would be kept to answer the following questions:

- What was the change was and why was the change requested?
- Who wanted the change made?
- Who approved the change?
- Who made the change?
- Who verified the change?

Change requests apply not only to new or changed requirements, but also to system failures and to defects in life-cycle work products. Changes at the organizational or system level or even the developmental or project level should follow a change request process. The change request process typically contains the following steps:

- The change request is recorded.
- The impact the change will have on the work product, related work products, schedule, and cost is determined.
- The change request is reviewed and an agreement is reached with those affected by the change request (the relevant stakeholders).
- The change request is tracked to closure.

Change control involves tracking each configuration item, approving a new configuration, and updating the baseline of configuration items. Check-in and check-out procedures should be used to maintain the correctness and integrity of the configuration. All changes and the reasons for the changes must be recorded in sufficient detail to provide a change history and support configuration management status accounting, which is discussed later.

Peer reviews, unit testing, and regression testing should be applied to changed configuration items to ensure that the changes have not caused unintended effects on the baseline. Regression testing has been often called "configuration management's best friend" because it shows that the changes have indeed fixed the identified problem, and that the functions that worked before still work. Authorization from the appropriate level of configuration control board must be obtained before the changed configuration items are reentered into the configuration management system. Changes are not official until they are released and all relevant stakeholders are notified.

The change control process should specify who can initiate the change requests and what the criteria are for placing the software/hardware components under formal or system change control. The "change impact" analysis must be conducted for each requested change even if it is informally carried out by the project team members. Check-in and check-out procedures should be used and should support how the revision history should be kept. The change control process should indicate how change requests will be linked to the trouble reporting system. The process for tracking and resolving change requests should be clear. It should specify the steps that will be followed to update all affected product life-cycle components. This process should require certain reviews and/or the regression tests that must be performed to ensure that changes have not caused unintended effects on the baseline. It should also spell out the process the configuration control board at the appropriate level should follow to approve changes. A sample change procedure is illustrated in Figure 8.5.

During the preparation for making a change, a clear understanding of the change and its impact to the existing system must be understood. To this end, the following questions must be answered:

- What is the perceived value of the change?
- What is the priority of the change?
- Which baselined elements are affected by the proposed change?
- What is the extent and impact of the change?

Following the impact analysis, an evaluation must be made to determine whether or not the change is to be implemented. Other questions include these: Does the manpower exist or can it be rescheduled? What is the effect on current development projects? What is the effect on current interface descriptions? What effect would this change have on related products currently in the product line? What support is necessary and available or must be purchased or built such as machines or tools? Can the change be accomplished given the required time constraints? What is the customer or market priority?

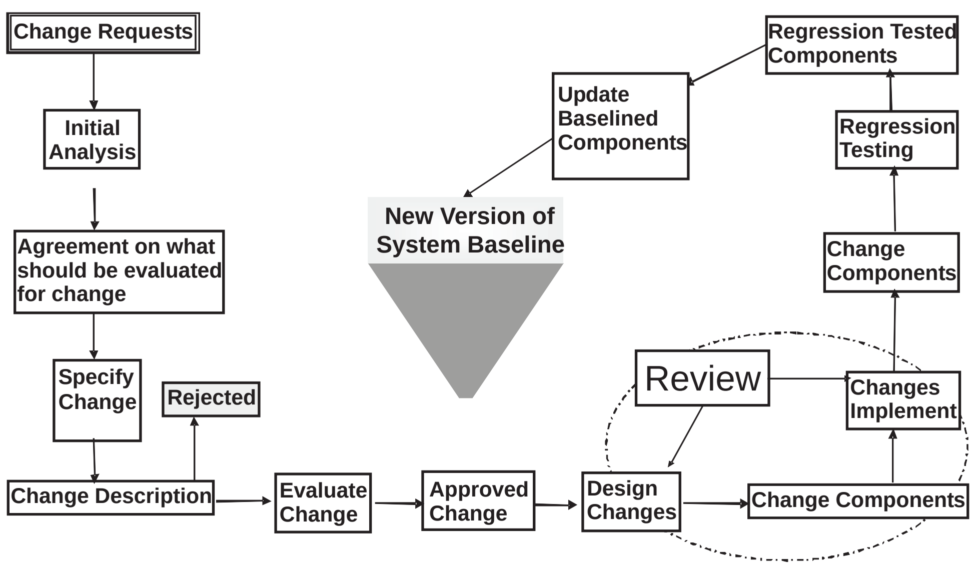


Figure 8.5 Incorporating a change into the current configuration baseline.

Input Needed to Decide on Proposed Changes

An organizational configuration control board typically needs the following information on each proposed change:

- *Software size*. How many new or changed lines of code are likely to be required?
- *Hardware interfaces*. How many interfaces will need to be changed? How many components will have to be revised?
- *Alternatives*. What, if any, are the alternatives to making the change and why was the proposed one selected?
- *Complexity*. Is the change within a single component or does it involve multiple components?
- *Schedule*. What are the required and planned dates?
- *Impact*. What are the future product consequences of this change?
- *Cost*. What are the potential costs and savings from making this change?
- *Severity*. How critical is this change?
- *Relationship to other changes*. Will other changes eliminate or invalidate this one, or does this change depend on other changes?
- *Test*. Are there any special test requirements?
- *Resources*. Are the needed people available to do the work?
- *System impact*. What are the memory, performance, or other system consequences of this change?
- *Benefits*. Can any special advantages be derived from this change?
- *Change maturity*. How long has the change been under consideration?

Configuration Control Boards

Configuration management is an accepted part of the infrastructure that is necessary in the support of systems development [2]. The goal of configuration management is to minimize confusion so that work can get done on a project. This is done by eliminating mistakes, which can be accomplished through the identification, organization, and control of modifications of product components in the development effort.

In the preceding section, we discussed the change control procedure and what inputs are required in order to make a decision to modify a baselined configuration item or not. In this section, we discuss the various levels of configuration control boards and the authority to approve change requests that are often associated with them.

Organizations frequently refer to the change control board at the organizational level as the configuration control board (CCB). Change control at other levels, especially the developmental or project level, is often ignored. Institutionalization, as defined by the CMMI®, includes placing designated work products of the process under appropriate levels of control. Different levels of control are appropriate for different work products at different points in time. For some work products, it may be sufficient to maintain version control at, say, the individual or project level. Sometimes, however, it is critical that work products be placed under formal change control. Additional levels of control are possible at any point in the product life cycle [1]. Using different levels of change control is a project management tool that can be used by the project manager whenever a system that seems to be approaching stability starts to become unstable again.

We will refer to Figure 8.4 for the remainder of this description of CCBs and their authority to approve change requests.

The configuration control board at the organizational or system level may contain representatives from different departments and disciplines as suggested here:

- Program management,
- Systems engineering,
- Software engineering,
- Hardware engineering;
- Software quality assurance;
- Hardware quality assurance;
- Configuration management;
- Integration and systems test;
- Documentation;
- Customer representative.

Product line CCBs may be established for organizations with distinct product lines. Project-level or developmental CCBs may be established and may include the project leader, the QA Responsible, CM Responsible, and an Integration and Test Responsible. The project-level CCB is typically given authority to approved change requests if they do not have any effect on the baselined requirements specifications.

The following scenarios are presented to assist the reader with some practical insights into these different levels of change control and their respective authority. If there is a hierarchy of CCBs, the configuration management plan should describe the procedure that ensures that a change request is considered by the appropriate CCB.

Using Figure 8.4 again, the following scenarios show the necessary level of CCB authority to approve a change request:

Scenario 1. The project is a software-only project with a software project manager. If the change request is to a design document and that design document is totally within the control of the software project, and there is no effect to the existing requirements specification at the organizational level, the project manager together with the QA Responsible and sometimes the CM engineer and even test manager become the de facto CCB at the developmental or project level and have the authority to make the change request approval.

Scenario 2. If the change request is to a design document that happens to be an interface document to a hardware device, the de facto CCB becomes the software project leader and the hardware project leader together with their respective QA Responsibles, CM engineers, and test managers as long as there is no effect on the existing requirements specification at the organizational level. If they cannot agree, the organizational CCB will be required to meet and make the final decision about whether or not the change request is to be accepted and thus implemented.

Scenario 3. If the change request to the design document does cause a change to the baselined requirements specification and that affects the "contract" with the customer, then the CCB approval must come from the organizational CCB.

These possibilities must be documented in this part of the project's CM plan so there is no confusion once the project gets started. It also serves as a guide for the quality assurance group, in its audit role, to ensure that change request approvals are being made properly with the correct level of approval authority.

The goals of the various change control boards at the developmental and organizational levels or any other defined level necessary for project or product line control are to ensure that change requests are handled complete with impact analysis, informal or formal, and that new versions along with change history are created for the baselined items.

What happens if multiple divisions in multiple countries develop product components that get integrated into systems and those systems get integrated into systems of systems? It should be clear that a hierarchy of change control boards must be in place with very clear guidelines as to who can make a change request and the process that must be followed for each division to determine the impact of the change request on its work and then to the systems and finally to the system of systems that will be installed at the end-user site itself.

Before we end this portion on change control boards, I would like to share an implementation scheme that was developed by Postbank OBV in Amsterdam, which is part of the ING Group. I was privileged to work with this division of ING in the late 1990s. They made their chief architect the head of a requirements configuration control board. This was a special CCB that examined change requests that were related to the requirements, the system architecture, and the business strategy. If a change request did not have any effect on the architecture or business strategy and seemed to be a normal requirements change request, it was passed down to the standard organizational CCB and processed per standard change control procedures. However, if there seemed to be a possible impact to the architecture or business strategy, senior managers and customer representatives were called together to assess the impact and discuss alternative solutions.

Configuration Management System

The configuration management system stores the configuration items created during the product life cycle or references to them and prevents unauthorized changes to the baselined items. The configuration management system can be viewed as a repository where changes to the baselines and releases of products and product components take place in a controlled fashion. A well-functioning configuration management system includes the following functions:

- Managing multiple levels of configuration management;
- Storage and retrieval of configuration items;

- Sharing and transferring of configuration items between the different control levels;
- Storage and recovery of archived versions of configuration items;
- Storage, update, and retrieval of configuration management records;
- Supporting the tracing of the requirements, design, code, or hardware components, and test cases to the source from which they were derived;
- Creating and disseminating configuration management status reports;
- Helping to ensure correct creation of products from the release baseline with authorization from the configuration control board;
- Ensuring the integrity of the baselines through baseline configuration audits, functional configuration audits, and physical configuration audits;
- Coordinating product releases through the various manufacturing departments such as manufacturing engineering, reliability testing, software reproduction, and product documentation;
- Preserving the contents of the configuration management system:
    - Backup and restoring of configuration management files;
    - Recovery from configuration management errors;
    - Archiving of configuration management files;
    - Disaster recovery.

The configuration management system section of the CM plan should describe the storage media, the procedures for accessing the CMS, the tools used for accessing the CM system, and the CM person or CM Responsible who is skilled in the use of the tool, the storage media, and the format of the information kept in the CM system to support the project manager's requests and to support periodic status accounting reporting.

Configuration Management Status Accounting

Configuration management status accounting is used to maintain a continuous record of the status and history of all baselined items and proposed changes to them. It should be able to answer these questions: What changes have been made to the system? What changes remain to be implemented?

The information required for a comprehensive status accounting includes:

- The time at which each baseline was established;
- When each configuration item and change was included in the baseline;
- A description of each configuration item;
- All change requests;
- The description of each product or product component change.

Note that many organizations use a CM tool to control the check-in and check-out procedures. Check-in procedures normally require the change description to be input along with the changed configuration item such as a detailed design document. However, these self-provided change descriptions should be peer reviewed and checked periodically to ensure that correct and usable information is being entered into the system for later reporting or status accounting.

My CMMI® assessment experience has shown that often the change history does not match the current baselined configuration items. In some cases, change descriptions were:

- I like peanut butter.
- Bug fixed.
- Updated.
- . (Period)
- {Blank Space} (This one wins the prize as the CM tool only required one character and a blank space satisfied the tool.)

Configuration management status accounting for systems, hardware, and software engineering items follows the concepts of financial accounting. Imagine going to an automated teller machine and using your bankcard to find out your balance. Suppose you felt your balance was around $2,000. When you query the automated teller system you find out your balance is $1,000. Normally in this situation, you would want to find out what transactions had taken place to result in that unexpected low balance. Most automated teller systems would be able to print out your buying activity for the past 15 or 30 days.

You may find out that your spouse had purchased a new set of golf clubs for $1,000 and that was the reason your balance was not what you expected. Configuration management status accounting operates in the same way. Status reports on additions and changes to the evolving configuration should be readily available. To support this critical function:

- Configuration management actions must be recorded in sufficient detail so that the content and status of each configuration item is known.
- All relevant stakeholders, especially project leaders, should have access to and knowledge of the configuration status of the baselined configuration items.
- Previous versions should be able to be recovered.
- The difference between successive baselines must be able to be clearly described.
- The current status and history of each configuration item must be maintained and updated as necessary.
- All relevant stakeholders must have access to and knowledge of the configuration management system through defined and understood request procedures that may involve a human being or may be provided by a CM tool.
- The reports must be detailed enough to support each project's management needs.
- Standard configuration management status reports should be sent out to affected groups and individuals on a periodic basis.

Configuration Auditing

Configuration audits should answer the following questions:

- Does the system satisfy the requirements?
- Does the documentation represent the system as built?
- Are all changes incorporated in this version?
- Are only approved changes incorporated in this version?

Configuration audits verify that the product is built according to the requirements, standards, or contractual agreements. The audit verifies that all product components have been produced, correctly identified, and described and that all change requests have been resolved.

Baseline audits should be conducted at phase end or some other designated point in the product life cycle to continuously ensure that the completeness and correctness of the baselined configuration management system contents are verified based on the requirements as stated in the project plan and approved change requests. Baseline audits can be carried out by examining the contents of peer review reports, test reports, the version description document, release note content and directories, simulation and emulation results, and differences between an original document and its updated version.

The product's functionality and performance should be compared to the requirements. In addition, the documentation that is baselined for maintenance activities (architectural specification and design specification) as well as for operational use (user manuals, operations manuals, installation manuals) should be compared to the requirements.

When the product is ready to be packaged and delivered, this final configuration auditing is often referred to as function configuration auditing (FCA) and physical configuration auditing (PCA). Functional configuration audits verify that the delivered product or product component satisfies the requirements and all approved change requests and nothing more. Physical configuration audits provide an independent evaluation of the system configuration items to confirm that each configuration item that makes up the "as-built" system maps to its specifications; in other words, that all the specified parts and code in the correct version are in the product. These formal audits are held to verify that the product and its documentation are internally consistent and ready for delivery to the customer or end user. Appropriate customer deliverable documentation includes installation manuals, operating manuals, maintenance manuals, and release notes or version description documents.

If the FCA and PCA are based on ongoing baseline configuration audits, shown in Figure 8.6, these audit activities, at the end of the project, are often no more than a confirmation and a spot-check of those baseline configuration auditing reports.

Interface Control

Control of interfaces is becoming one of the most critical tasks in building systems today. This is especially true as many projects are being built at multiple sites, in multiple countries, and within multiple cultures. In addition, suppliers are being

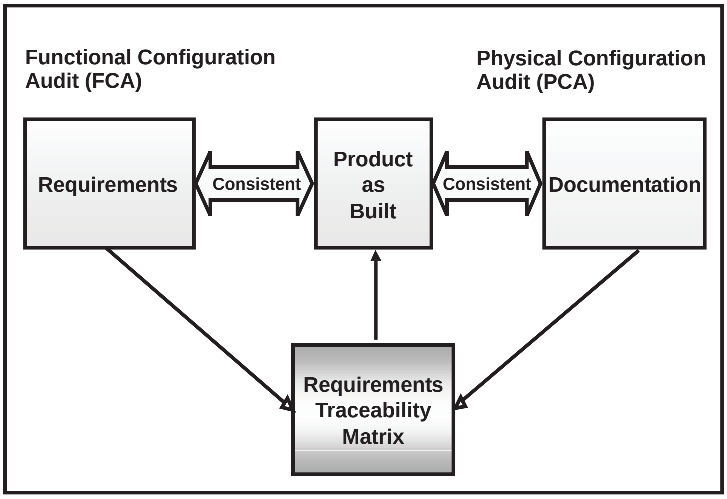


Figure 8.6 FCA and PCA audits.

used more than at any other point in history. This simply means that the identification of interface requirements, the establishment of interface descriptions, and the configuration management control of those interface descriptions is important to the success of many of today's projects. The CMMI® describes the identification of interface requirements in the Requirements Development process area, the establishment of interface descriptions in the Technical Solution process area, and the control and use of those interface descriptions in the Product Integration process area.

Interfaces are often classified into two major categories:

- Organizational interfaces where the transfer of configuration items is controlled between individuals, the project and support groups, and the customer. Organizational interfaces include interfaces between various organizations or groups involved with the product:
  - Vendor to customer;
  - Project to project;
  - Codeveloper to codeveloper.
- Technical interfaces, which include:
  - *System interfaces*. The explicit interfaces between the system and the software configuration items whose functionality must accomplish the system requirements,
  - *Life-cycle phase*. Transition interfaces between those life-cycle phases of the product,
  - *User interfaces*. Logical characteristics of each interface between the product and its users;
  - *Software interfaces*. The agreements shared between software modules and other software components;

- *Hardware interfaces.* Agreements shared between other product component and any hardware component in the environment with which it must interface;
- *Communication interfaces.* The interfaces between the software modules and communication hardware and software such as local-area network protocols.

Interfaces may be internal to the system or external. An example of this is the telephone. The internal interfaces are the software and components internal to the device. The keyboard and the screen are examples of external interfaces. These are examples of technical interfaces.

Control of Supplier CM Functions

If a portion of a development project is to be subcontracted to a supplier outside of the project's boundaries, the responsibility for the configuration management of the overall system belongs to the buyer organization. The supplier is only responsible for the portion of work that has been tasked according to the supplier agreement. Therefore, it is very important for the buyer organization to thoroughly understand the configuration management capabilities of its suppliers.

Questions such as the following should be asked and answered:

- What supplier life-cycle work products must be placed under configuration control to ensure consistency with the main development effort?
- What CM concerns need to be added to or removed from the contract with the supplier?
- What audits and procedures need to be established for the supplier?
  - What are the supplier's obligations?
  - What are the contracting organization's obligations?
- Are the supplier's configuration identification schemes compatible with those of the buyer?
- Does the supplier have an effective means for managing the interface descriptions?
- Has the frequency and format of the supplier's configuration management status reports been agreed on?
- Does the supplier understand the importance of configuration audits and understand what the buyer expects of it regarding them?

It is recommended that the RFP and supplier agreement contain the stipulation that the buyer organization has the option and authority to conduct audits on the supplier's configuration management activities before, during, and after contract award. Informal CM-to-CM representative communication should take place on a regular basis during any given month so that the monthly project management review of suppliers will not result in any surprises, especially regarding configuration management.

Release

Products from the configuration management system–approved baselines should be created and released according to a documented procedure. Basic Release activities to be performed include:

- Ensuring that the Product Package is complete and accurate before it is sent to the customer;
- Ensuring that the FCA results show that the product about to be shipped meets the requirements and approved requirements change and nothing more;
- Ensuring that the maintenance documentation is accurate;
- Ensuring that the installation instructions are accurate;
- Ensuring that the operator's manual is accurate and helpful;
- Ensuring that each customer's constraints are known before delivery;
- Ensuring that the customer or end user's site is prepared to receive and accept delivery;
- Ensuring that acceptance testing support is available in the event of problems being discovered during acceptance testing;
- Development of the Version Description Document to describe the product that is being delivered accurately along with any other necessary Release Notes, including:
  - New functional capabilities—A brief summation of the released system's new capabilities;
  - Closed problem reports—A list of all of the change requests that were incorporated into this release;
  - User considerations—Any special actions the users must take in using the new release: the use of a new function key;
  - Demands that the new release makes upon its environment—a new release may require a math coprocessor where the previous release did not;
  - Limitations—A list of the known faults and limitations of this version of the system;
  - Inventory—A list of the source and executable modules and data objects that are contained in this release;
  - Installation instructions—How to read the magnetic media, discs, or CD-ROMs; install the system; and then run tests to verify that the system has been installed correctly.

Summary

The quality management functions that are expected to be implemented on any project interested in developing products that satisfy the requirements of product and service quality include, but are not limited to:

- Setting quality goals for the project that support the organization's business objectives;
- Conducting peer reviews throughout the product life cycle;
- Performing multiple levels of testing such as unit testing, integration testing, systems testing, and acceptance testing;
- Designing in quality factors such as maintainability, expandability, and reliability;
- Conducting objective evaluations with respect to process quality;
- Controlling the integrity of the evolving product or product component through configuration management;
- Providing visibility into the process and product quality for management and practitioners through quality reporting;
- Getting noncompliance issues resolved before the product is delivered to the customer.

CM is one of the most important processes that project leaders can use to evolve and deliver their product in a controlled manner. Knowing the state of the product that a project is developing and knowing that it satisfies the customer's requirements is of the utmost importance for any project leader. Because many of the most frustrating product or product component problems are often caused by poor configuration management, proper CM is not only appropriate, but is a critical part of the infrastructure necessary to build systems.

References

[1] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI Second Edition, Guidelines for Process Integration and Product Improvement,* Reading, MA: Addison-Wesley, 2007.

[2] Sage, A. P., and W. B. Rouse, *Handbook of Systems Engineering and Management*, New York: John Wiley & Sons, 1999.

CHAPTER 9

Enabling Project Managers to Manage and Control Better Through Supplier Management

Working with Suppliers

Working with suppliers is now a fact of life for most companies. For those companies that are international and have projects developed at multiple sites and in multiple companies, the challenge to manage and control all of these various types of suppliers is greater than ever, and in some cases overwhelming. Too often companies do not have their own requirements engineering, requirements management, project management, or quality management activities under their control. Attempting to manage their project's suppliers in addition to their own project activities is difficult at best.

To set the stage, it is important to define *buyers* and *suppliers*. Clear terminology is very important to understanding the importance of each role. A company may be a buyer for its suppliers and a supplier for its prime contractor.

- *Buyer*. The project or organization that is setting up an agreement with an entity outside of the project's or organization's boundaries to develop a product or product component for delivery. Outside of the project's boundaries indicates the buyer normally has no control over the supplier's resources.
- *Supplier*. A project inside or outside of the buyer's business unit or organization that agrees to do the necessary product or product component development according to the requirements of the buyer and deliver within specified constraints such as cost, schedule, quality, and performance.

Myths About Suppliers

With those rather simplistic definitions of buyer and supplier, let us take a look at the most popular myths that surround supplier management today and that are often offered as excuses for projects and organizations not to properly manage their suppliers:

- *You must trust your suppliers or you would not have chosen them*. This statement implies that the instant an organization selects a supplier, trust automatically comes with the contract or agreement. In such cases, the buyer has made an erroneous assumption in that it assumes that all of the risks have now moved from buyer to supplier and, therefore, have "vanished." Companies that have built long-term relationships with preferred suppliers know that developing trust in a relationship is an incremental process. A better approach would be that first a supplier is given smaller, less risky assignments, and then, on successful completion of those assignments, more and more complex, demanding, and critical projects are offered. Other factors, such as the supplier's willingness to share entrepreneurial risks, may be one of the most critical factors for the buyer to offer the supplier the ultimate trust.

- *You must never allow your supplier to suggest a change to the requirements or to the contract.* It is the author's experience that many buyer organizations feel that suppliers will always try to cheat them; as a result, they write RFPs and supplier agreements explicitly stating that the supplier is not allowed to suggest any change to the requirements specified in the supplier agreement. If a buyer truly does not trust its supplier, then why would it enter into a relationship in the first place? I am sure that many readers will have instant answers to this question, but I suggest that it is a high-risk approach that your organization takes with supplier management.

  If you have a supplier that your organization feels has the technical capability and the integrity to work with, then a well-thought-out suggestion for improvement to the requirements by the supplier may indeed be worthy of consideration. In today's fast-paced economy, ensuring open communication and dialogue may be crucial to the development of the product that will actually be wanted at the time of delivery. In fact, the CMMI® does strongly suggest that revisiting requirements is actually more desirable in the long run. The buyer can always "just say no," but at least it has reviewed the input from the experienced supplier with whom it has contracts.
- *If you do not have much expertise, it is all right to have your supplier define your requirements for you and also the acceptance testing criteria. It is also acceptable for the supplier to then conduct the acceptance testing for you.* This point has always bothered me when I heard a buyer make a strong point of it. I would invariably push back and state that the responsibility of establishing acceptance criteria and conducting acceptance testing belongs with the buyer. True, the supplier might assist the buyer in conducting the acceptance testing, but the responsibility still remains with the buyer. Too often, the buyer's representatives would tell me that they did not have the expertise to conduct acceptance testing and, therefore, had to trust the supplier. This would take me back to the first point. I have suggested that the buyer could hire an outside test organization to help it with the acceptance testing. One organization told its supplier that it needed an outside organization for acceptance testing and asked the supplier to suggest one. Unbelievable? This is not an uncommon tale.
- *Using a supplier is always cheaper than developing a product or product component in-house.* This myth may be one of the most misleading as well as costly myths about suppliers—and this myth invariably involves the fallacious concept mentioned earlier about shifting of risk to the supplier.

Supplier management may involve all of the topics listed here (the author is by no means suggesting that this list is complete or exhaustive):

- Developing/eliciting the customer's requirements,
- Deciding to use a supplier;
- Selecting a contract management team;
- Determining the desired buyer–supplier relationship;
- Determining an acquisition strategy;
- Setting supplier evaluation criteria;
- Developing the contract management approach;
- Preparing the statement of work (SOW) and the RFP;
- Conducting supplier capability evaluations;
- Evaluating the responses to the RFP and selecting the supplier;
- Final negotiating of a contract;
- Establishing the supplier agreement;
- Expectation setting—getting started;
- Monitoring progress and performance;
- Conducting process improvement progress checks;
- Acceptance testing;
- Transitioning from supplier to buyer.

Many of these topics are mentioned in the CMMI®-DEV v1.2, others are not. It is the author's intent to touch on most of these topics although not at the level of a book written strictly about supplier management.

- *If you choose a Maturity Level 5 supplier, it greatly reduces the work for the project, because the supplier will "manage itself."* This may be one of the more popular myths about suppliers that the author has encountered. It is the author's opinion that many buyers do realize the amount of work that it takes to manage a supplier, but do not feel they have the resources to do a proper job. Let us examine this statement a bit more closely.

  The buyer's organization is identified as one that does not have enough resources to do a proper job of in-house development. The buyer has either decided to use a supplier or has been ordered to use a supplier in order to become more of a "project management" company. Members of the organization soon realize that quite a bit more work is required to get prepared to issue an RFP, much less develop a supplier agreement, and then to actually manage a supplier that may be, in reality, 4,000 miles away in another country, speaking another language, and having another culture altogether. It is a daunting task at the very least!

  The answer that is often reached by management is to hire an organization that has declared itself CMMI® Maturity Level 5. If the supplier is a CMMI® ML 5, it will surely do a better job on any assignment given to it and will automatically manage itself. Why anyone would believe this myth is not understood by this author, but it happens, with the end result too often being an inadequate product delivered without all of the desired functionality or quality—and this sometimes leads to large money loss and even loss of market share of the buyer.

- *If you need multiple suppliers, ML 5 suppliers will be more willing to cooperate to support your business objectives.* This myth is closely related to the preceding one with a slightly different implication. CMMI® ML 5 suppliers will be so confident in themselves that they will admit to other suppliers they have made a mistake and help the buyer to find and fix it. They will do this regardless of the possible loss of face in front of the other competing suppliers with which they are being forced to cooperate. I suggest that this is not human nature, nor is this practice considered good business by most suppliers. If total cooperation is required by all suppliers involved on the buyer's project, the contract must be written to enforce the necessary cooperation among the suppliers in order to ensure that the buyer's objectives will be accomplished.
- *ML 5 suppliers are able to demonstrate expected performance regardless of the application domain.* This myth may be harder for the readers to believe, but this author has heard this statement made. It is as if the magic maturity level makes companies capable in areas in which they previously had no experience. Imagine a database developer being called on to develop software for a missile system! This would be ludicrous. Of course, we realize this myth is not true, but it is worth guarding against this myth if it comes up in your company.
- *ML 5 suppliers are able to support the culture of any buyer they are supporting.* Although similar to the points made earlier, this one is more subtle. The concept of culture being an important selection criterion does not receive the importance this author thinks it should. It is important to have a good relationship with your suppliers. Suppliers on whom you depend should not only have the technical capability, but the ability to work with and understand the buyer's culture and be willing to share its culture to ensure the buyer and supplier have a good chance of working well together. Culture may have greater ramifications than most buyers anticipate. What happens if one culture cannot say "no" because they deem this to be rude? The other party might assume that silence means assent, and yet it may instead mean disagreement. Or one party may say that it can perform the task asked, although it knows it cannot, but will agree anyway because it cannot state outright that it cannot do the task—for fear of violating a cultural concept of being rude.

When and Why Do We Use Suppliers?

So when and why do we use suppliers? Let us examine some of the reasons that organizations choose to use suppliers:

- When we do not know how to develop the product or product components;
- When we know how to develop the product or product components, but don't have the technology to do so;
- When someone else knows how to develop the product or product components better;

- When someone else knows how to develop the product or product components cheaper and with higher quality;
- When we have insufficient in-house resources;
- When it is desirable to create a strategic alliance because the supplier adds to the buyer's competitive position;
- When an organization wishes to pursue new business opportunities;
- When someone else already has a prototype or similar product;
- When using a supplier will result in a more cost-effective solution;
- When the supplier is willing to share some of the contract risk or entrepreneurial risk.

Sources of Problems Using Suppliers

Using suppliers is a form of risk and should be treated as so. (A more complete focus on risk can be found in Chapter 7.) Although supplier organizations and projects provide buyer organizations and projects with risks to be identified, classified, prioritized, and managed, it is often the buyer that contributes to that risk. Buyers often set unreasonable completion dates. The philosophy given is that if the buyer sets an unreasonable completion date for the supplier, the supplier will "try harder" to meet it. The buyer realizes the supplier cannot meet the unreasonable date, but believes that it will get a bit closer and thereby allow the buyer to get a little more out of the supplier.

In like fashion, the buyer sets and enforces a budget limit without understanding the full scope of the problem. This issue is addressed in subsequent sections of this chapter, but for now let us just accept the given statement as fact. Buyers expect a "cast-in-concrete" initial project cost and plan that provides a clear definition of the requirements to the supplier.

There are many reasons for this behavior, some of which have already been explored. It does seem reasonable that one would ask for a supplier's best initial estimate, but it does not seem reasonable for the buyer to expect the supplier to live with that initial estimate especially when the requirements have not yet been fully identified. However, if one looks to one's own organization, perhaps we impose these unreasonable constraints on our suppliers because we impose them on our own projects! Some buyers exhibit an "us against them" attitude right from the very start. That "I am the buyer and you are not" seems to be the attitude that is presented to suppliers. Just from a human nature point of view, this can only cause resentment and an uncooperative attitude. We must always remember that "We want our suppliers to succeed!" This tip is often ignored and buyer–supplier conflicts exist from the very start of the project.

My final reason for buyer–supplier conflict presented in this chapter is that the level of buyer participation throughout the life cycle is negligible at best. Many buyer organizations seem to feel that once the contract is in place and the requirements provided to the supplier, that this is sufficient. The idea that progress and performance checks should be provided to the supplier on a periodic basis is either not considered or is ignored for reasons already provided in the discussion on myths.

Problems and risks arise when a buyer organization decides to use suppliers. They are not insurmountable, but it does take work on the part of both the buyer and the supplier to resolve them.

We have discussed a few of the problems of using suppliers from the point of view of the buyer. Let us now examine the issue from the supplier's point of view. Suppliers, wanting to convince the buyer to choose them for the contract, often fail to identify and specify assumptions and contract performance constraints in plans/proposal. The supplier tends to agree to the buyer-imposed limits too readily and, therefore, fails to effectively negotiate a win–win solution for both parties. Suppliers often contribute to the buyer's risk by backing into an imposed impossible schedule in order to win the contract. To compensate, the suppliers reduce their focus on quality over the life of the project. Why? The suppliers then believe that too strong a focus on quality will further impact the already impossible-to-achieve schedule objective. The suppliers then fail to fully identify and manage risks either from their own risk sources or when dealing with the buyer. Either way, problems can result and will inevitably cause conflicts and contract failures for both parties in the future.

Regardless of the source of the problems of using suppliers, the typical results include:

- Products not delivered on time;
- Reduced insight into progress of the supplier,
- Delivery dates that seem to slip continuously;
- Quality of products or work products that is far below expectations;
- Weak insight into configuration management and into the integrity of the products and work products that the supplier is building;
- Weak insight into quality control processes such as reviews and tests;
- Costs that seem to increase or even skyrocket.

Different Forms of Suppliers

Suppliers may take many forms as indicated below and illustrated in Figure 9.1:

- In-house vendors;
- Other projects;
- Subcontractors (software and hardware);
- Sister divisions;
- Commercial-off-the-shelf (COTS) products;
- Customer-furnished equipment (CFE);
- Government-furnished equipment (GFE).

From my experience, having led more than 100 assessments and appraisals, I feel that it is virtually impossible for any organization to declare it has no suppliers to manage. Figure 9.1 illustrates the author's understanding of supplier sources. Most of these are mentioned in the CMMI®. Others have been contributed by cli-

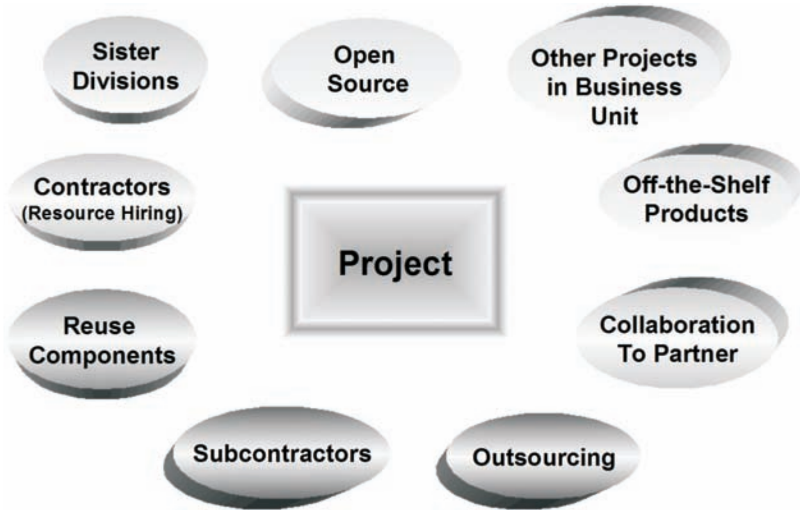


Figure 9.1 Supplier agreement management overview.

ents during CMMI® classes, supplier management classes, or assessments as being important to consider as suppliers as a direct result of their personal experiences. We will expand on some of these possibilities to give the reader the best understanding possible.

Subcontractors

This is the most classic form of supplier. Imagine a company that is located 1,000 miles away from your organization's site. Your project determines what the project's requirements are and allocates a subset of those requirements to be developed by the subcontractor. The subcontractor completes the WBS that your project started, helps to better understand the requirements, plans the project, builds the product or product component, verifies and validates it, and transfers it to your environment for acceptance testing.

Sister Divisions

Based on my experience, if there is one type of supplier that could be considered a major difficulty, it would be a "sister division." A sister division is another division or business unit of an organization that typically resides some geographical distance from the buyer organization (although there are many examples of large companies with sister divisions that are colocated in the same area that have the same classic sister division problems). The issues of dealing with a sister division as a supplier are endless. Let's examine a few of the more prevalent ones. The author has been repeatedly told that a sister division cannot or should not be considered as a supplier because it:

- Is part of the same company;
- Is on the same team;

- Is a codeveloper;
- Cares about the company first, therefore, would help any other division in the company that needs help;
- Is not a competitor;
- Has extra resources and is just helping out the buyer division of the company.

Of course, these words are nice to hear, but I prefer the philosophy of "Take care of business first, and keep your friendships." This philosophy also applies to sister divisions.

The use of sister divisions as suppliers is most vulnerable when a company is experiencing difficult financial times. A buyer division is ordered to use a sister division as a supplier to keep the money inside the company. On the face of it, that makes sense. The difficulty comes, however, when the sister division is not required to sign a supplier agreement and commits to the deliverables and schedule necessary for the buyer division's project success. Because the sister division is "part of the family," even progress and performance checks are often not conducted. Then when a critical delivery is expected and the buyer division calls the sister division and asks when the product or product component will be delivered, the answer typically given is "Oh, sorry about that! Shortly after you gave us the requirements for the product component you needed, good times came back to our division. We received some significant contracts that forced us to focus all of our resources on them and we just have not had time to focus on what you asked us to do." This results in much wailing and gnashing of teeth. Vice presidents of the respective divisions have gotten involved in shouting matches and name calling and it frequently takes the CEO of the company to straighten things out. This could have all been avoided or at least mitigated with a straightforward supplier agreement signed and committed to by both the buyer division and the sister division acting as the supplier.

Other Projects in the Same Division

Even if it is another project in the same division or business unit, and that project is in a room right across the hall from the project that is acting as the buyer, it should be treated as a supplier. If a subset of the requirements is given to the supplier project across the hall and they go through the standard product life cycle to design, develop, test, and deliver the product component, that project must be considered as a supplier and treated as such.

Off-the-Shelf Products

Off-the-shelf-products, or COTS products as they are often called, have been one of the most debated forms of suppliers since the CMM® for Software was released. After all, one cannot "negotiate" with a product that is shrink-wrapped can one? COTS products are often thought of by management as a source of a product that only costs whatever is on the price tag. This runs close to the myth that suppliers will manage themselves. This assumption may be the furthest from the truth. Before COTS products can be purchased, the project's requirements must be developed and used to help evaluate any supplier including a COTS product. Rarely does a COTS product satisfy 100% of a project's requirements needs. Therefore, the project must build some part of the product or bridge between the existing product components and the COTS product. This cost is not for free. For example, if the COTS product was an off-the-shelf database, every product component that uses that COTS database must be modified to ensure it works well with the interface defined for that database. This could mean that extensive changes would need to be made due to the volume of product components that make calls on that COTS database.

In the event that off-the-shelf products are desired, care in evaluating and selecting these products and the choice of vendor may be critical to the project. Candidate products should be evaluated against the associated requirements considering:

- Functionality;
- Performance;
- Quality;
- Reliability;
- Terms and conditions of warranties for the products;
- Supplier's responsibilities for ongoing maintenance and support of the projects.

In addition, the impact of candidate COTS products on the project's plans and commitments must be considered. The supplier's past performance and ability to deliver should be examined. Risks associated with the selected off-the-shelf products must be documented in the project planning stage. Benefits and impacts that may result from future product releases should be estimated. Other factors that should be considered in dealing with COTS products include:

- Discounts for large quantity purchases;
- Coverage of relevant stakeholders under the licensing agreement, including project suppliers, team members, and the project's customer;
- On-site support, such as responses to queries and problem reports;
- Additional capabilities that are not in the product;
- Maintenance support, including support after the product is withdrawn from general availability.

Reuse Components

Reuse components can be viewed in a similar manner as COTS products. The reuse library must have been developed for reuse. The project must still have knowledge of its requirements in order to determine if the reuse component library can indeed match most of those requirements and how much work project members will have to do in order to satisfy all of its requirements. Of course, some issues such as warranties may not come into effect with a reuse library, but treating a reuse library similar to a COTS product should prove beneficial to any project that is considering its use.

Contractors/Resource Hiring

Strictly speaking, contractors or temporary employees who are hired to fill a resource need when the company is not in the position to hire full-time employees or has chosen not to do so are not under the auspices of the Supplier Agreement Management PA according to the CMMI®. But in the author's experience, there have been multiple instances where the following scenario has played out.

> A company wanted to hire some contract programmers. They located an agent in the country from which they wanted to hire the contract programmers and negotiated a deal where the agency would hire the contract programmers and pay their salary and benefits such as vacation and health. The buyer's country had a law that clearly stated that if a contractor was hired for a certain length of time, the buyer's organization would be forced to treat the contractor as a full-time employee and pay employee benefits accordingly. The contract programmers were hired and moved to the buyer's organization and country. They did a great job and everyone was happy with the hiring choice.

Suddenly, the tax authorities of the buyer's country stated the buyer's organization was in violation of the law regarding keeping contractors too long without paying their employee benefits. Of course, the legal representatives of the buyer's organization protested and explained the arrangement with the agency in the contractor's country. The tax authorities then explained the buyer's country's laws and firmly made the case that the buyer's organization would have to pay substantial financial penalties. The contract programmers packed up and went back to their home country. The agency closed its doors and changed its name. The buyer's organization lost the contract programmers, had significant project delays, and realized millions of dollars of financial penalties.

The reason for bringing up this situation is that the author has found that this way of handling contractors from outside of the buyer's country is true for a significant number of countries. The message is clear: Before you go about hiring contractors from another country, make sure you get sound legal advice on what your country's tax laws are to avoid the situation described.

Outsourcing

Outsourcing involves transferring or sharing management control and/or decision making of a business function to an outside supplier, which involves a degree of two-way information exchange, coordination, and trust between the outsourcer and its client. Such a relationship between economic entities is qualitatively different from traditional relationships between buyers and sellers of services. The economic entities involved in an outsourcing relationship dynamically integrate and share management control of the labor process rather than enter in contracting relationships where both entities remain separate in the coordination of the production of goods and services. Business segments typically outsourced include information technology, human resources, facilities and real estate management, and accounting. Many companies also outsource customer support and call center functions, manufacturing, and engineering. Consequently, there is a great deal of debate concerning the benefits and costs of the practice as well as how to categorize it as a phenomenon [1].

Unfortunately, in many books and papers, and contracts, the term *outsourcing* has taken on multiple meanings in similar fashion to the terms *Xerox* and *Coke*. If you ask someone to Xerox a document, you are, of course, asking the person to make a copy even though Xerox is the brand name of a company that makes copiers. If you ask someone for a coke, you do not necessarily mean a Coca-Cola—you could mean any dark-colored sweet carbonated beverage. In like fashion, outsourcing has come to mean everything from the classic definition of taking over a service such as computer operations for a company to providing subcontractor services.

In general, outsourcing does require as much of the same care as supplier management, but it must be viewed with caution so that the meaning of outsourcing is the same for the supplier as it is for the buyer. And of course, the potential loss of management control must also be factored in.

There is a focus on outsourcing today that was not prevalent in the industry a few years ago. For example, systems testing is a function that is being outsourced to companies that specialize in integration and systems testing. However, "buyer beware" must again remain the slogan. Systems integration and test companies provide their services to other companies that may also be your competitors. Care must be negotiated to ensure that knowledge of the buyer's product does not somehow find its way into competitor's hands. Should the systems integration and testing supplier have an endless supply of security firewalls? Should high security measures be required for privacy of intellectual property rights for any company using the services of the systems integration and test organization?

Another question that is being raised frequently now is this: What rights does the buyer have if it outsources maintenance work to an outside company? Some companies that are trying to obtain outsourcing contracts will ask to place a contingent of project people with one project manager on the site of the buyer, and then tell the same buyer that they do not need to contact or worry about the other 150 people that will be doing the actual work "back home." I totally disagree with that approach and strongly suggest that any organization negotiating on behalf of its projects to an outsourcing supplier ought to ensure that representatives of the project or its organization have the right to make on-site visits and audit selected processes and work products in order to gain confidence that the outsourcing company can indeed provide the level of service to the buyer's customers that they have come to expect.

Open Source

Open source is a set of principles and practices that promote access to the production and design process for various goods, products, resources, and technical conclusions or advice. The term is most commonly applied to the source code of software that is made available to the general public with relaxed or nonexistent intellectual property restrictions. This allows users to create user-generated software content through incremental individual effort or through collaboration.

The open-source culture refers to creative practices that involve the appropriation and/or free sharing of found or created content. Open source defines a culture in which fixes are made generally available. Participants in such an open-source culture are able to modify open-source products, if needed, and redistribute them into the community or other organizations.

The rise of open-source culture in the 20th century resulted from a growing tension between creative practices that involve appropriation, and therefore require access to content that is often copyrighted, and increasingly restrictive intellectual property laws and policies that govern access to copyrighted content.

Although artistic appropriation is often permitted under fair-use doctrines, the complexity and ambiguity of these doctrines creates an atmosphere of uncertainty among cultural practitioners. In addition, the protective actions of copyright owners create what some call a "chilling effect" among cultural practitioners. Critics have said that the term *open source* fosters an ambiguity of a different kind in that it confuses the mere availability of the source with the freedom to use, modify, and redistribute it [2].

Although the use of open-source products is considered to be a "type of supplier," it is also considered a very complex type of supplier that most project managers and members do not know enough about. If an organization decides to use an open-source product as part of the product it is developing, who will be responsible for the maintenance of the version of the open-source product that has been incorporated into the project's product? The reader might feel that this question is relatively easy to answer. But consider this situation. For some open-source products, if an organization decides to make the open-source product a composite part of its own product, the open-source laws might require or force that organization to make the entire product it has built, including the open-source product, open source. What an unpleasant surprise that would be if it were not known in advance! This situation was reported to the author as an authentic possibility.

Collaboration to Partner

Last, but not least, we turn our attention to the supplier type called "collaboration to partner." This is tightly linked to the relationship the buyer has with the supplier, which is discussed later in this chapter. The most important aspect of treating a supplier as a collaborator or partner has to do with the willingness of the supplier to share the entrepreneurial risk. Some suppliers have totally aligned their vision and business objectives with those of their buyer. These suppliers have spent 3 to 4 years of their own resources in parallel to the buyer in order to show loyalty and gain a share of the usually significant contract that might be awarded. These suppliers meet with the buyer's most important customers and offer solutions that are in line with the philosophy of the buyer.

Supplier Agreement Management

Supplier agreement management addresses the need of the project to effectively select and manage those portions of the work that are conducted by suppliers. We examine most of the important concepts that were discussed under the earlier section on myths about suppliers.

Requirements Development by the Buyer

A typical scenario that continually repeats itself throughout the IT world is one in which the buyer's organization does not have a sufficiently competent staff to develop or manage the projects they are being asked to deliver. The senior management team decides to outsource major portions of system development to a supplier. But because of their own lack of knowledge about requirements gathering, project management, or quality management, the buyer's organization asks the supplier organization to not only develop the requirements, but to also develop the acceptance criteria and the acceptance tests. From the author's point of view, this is a very risky way in which to work with suppliers.

While teaching a public supplier management workshop, the author explained why the buyer's organization should develop the WBS, the requirements, and the project plan to a low enough level to be able to put out a RFP that included a statement of work (SOW). A participant asked in an obviously agitated voice:

> Are you saying that we, the buyer organization, need to do some work before we choose a supplier? Why should we decide to select a supplier if we have to do any of the work? That is why we want a supplier so that we do not have to use any of our resources and do any of the work.

Sound unbelievable? This class was taught in 2001. But the scenario and the questions are presented to me again and again. There are many good reasons to use suppliers. One truth must be kept in mind when a decision is made to use a supplier:

Either you control your suppliers or they control you!

Requirements Elicitation

Requirements elicitation is the name usually given to the activities involved in gathering the requirements that will be used to define the system to be developed and obtained through multiple techniques and sources.

Although the supplier is responsible for eliciting requirements from the buyer, the task should be viewed as a joint responsibility. The buyer must understand its customer's needs, wants, constraints, and interface requirements and convey them to the supplier to a low enough level to be able to decide which requirements the supplier will be assigned. The supplier is then responsible for following the requirements engineering processes to elicit the requirements from the buyer, analyze them, and validate them with the buyer throughout the product life cycle. Requirements questions that should be asked to determine what requirements to present to the supplier include these:

- How are the requirements related to the buyer's organizational business requirements?
- What is the buyer's relationship with the supplier?

- What technical ability must the supplier have?
- What technical performance is required and is the supplier capable? What other projects can the supplier discuss and use to demonstrate the actual performance delivered?
- What quality factors is the buyer seeking for the supplier to deliver?
    - Reliability;
    - Usability;
    - Maintainability and expandability;
    - Safety or security.

The eliciting, analyzing, and specifying of requirements is an iterative process. The iterations are performed until the buyer and supplier agree to the requirements. Elicitation of requirements can be broken down into more detailed steps that can be turned into procedures and checklists to ensure these process steps are followed:

1. Identify stakeholders;
2. Set objectives;
3. Understand organization background;
4. Organize knowledge;
5. Gather requirements from stakeholders;
6. Determine document requirements.

Deciding to Use a Supplier

Deciding to use a supplier can occur at many points during the product life cycle. A sister division of the same company may be chosen as the supplier by higher level management of an organization because of economic factors. The project or product requirements may limit the choice of potential suppliers to a small number. Or the senior manager of a business unit may have a favorite supplier that he/she wants to use.

Even under the circumstances in which the supplier is prechosen, the buyer still has the obligation to provide the project details that would normally be developed following the processes shown in Figure 9.2.

Describing System Objectives

The supplier needs to have an understanding of the buyer's organization such as its background, its main business focus, and the leading products and services. The buyer should document the reason for the system it is building or wants built. Is the system being built for research and development? Is it being built for a specific customer? Is it being built to meet existing/anticipated market needs/requirements? Or is it being built to upgrade an existing system to a newer technology? In short, what are its business objectives?

What is the problem the system is trying to solve? Is it a known problem or an anticipated problem? With what other systems must it interface? What protocols must be used for the new system to interface with the existing systems? Who are the

Figure 9.2 Deciding to use a supplier—activities overview.

anticipated users of the system and how will they use it? What are the expected changes in its environment over the product's life cycle? What are its technical objectives? What is its market focus?

Developing a System Overview

The system overview should describe the interactions between the system and its operational environment. Contents of the system overview include:

- Who its users are and in what manner they will use it;
- How long the system must operate;
- How effective the system's performance must be;
- Expected changes in the system's environment over its lifetime;
- Other systems with which it must interface;
- Protocols it must use to interface with them;
- Interaction of the product, the end user, and the environment as it is understood from the buyer's point of view;
- How the system will be:
    - Developed;
    - Produced;
    - Deployed;
    - Trained;
    - Used operationally;
    - Maintained;
    - Refined (supported);
    - Retired (disposed).

Determining and Listing System Constraints

The buyer is responsible for determining and listing all critical system constraints in order to guide the supplier in its attempts to design, develop, and deliver the product or product component. Those constraints could include:

- System performance;
- System response time;
- Safety considerations;
- Security considerations;
- System capacity or throughput;
- Quality factors or user attributes such as maintainability, expandability, reliability, and interoperability;
- Design constraints (e.g., a design constraint may be that the system to be developed must obtain its information from an existing database);
- Interface requirements (critical if the supplier's product must interface with existing systems);
- Project constraints such as personnel or equipment;
- Process constraints such as standards, languages (spoken, written, computer), and regulations that must be followed on the project;
- Qualification requirements that suggest where the product and product components must be verified and validated;
- Environmental constraints stating the physical setting and social and cultural conditions of the system development effort and the setting in which the system will be used;
- Budget constraints;
- Schedule constraints.

Determining Resources Needed

The reason that the buyer should estimate necessary project resources is to help guide the development of the SOW and the RFP. What tools are expected? What level and numbers of project staff are thought to be required? For example, does the buyer believe that the project will take 20 people each with 15 years of experience in the application domain? Will it be a problem if this level of experience or these numbers of experienced people cannot be found? If so, this should be documented and used as part of the supplier evaluation when the supplier responds to the RFP.

- Staff:
  - Engineering resources;
  - Quality assurance;
  - Configuration management;
  - Testing;
  - and so forth;

- Tools:
  - Analysis;
  - Design;
  - Project tracking;
  - CM;
  - Testing;
- Facilities;
- Systems;
- Training.

Developing a Work Breakdown Structure

The buyer is responsible for developing the WBS to at least two or three levels to gain an understanding of the scope of the project and to guide the supplier in the detailed development of the WBS to the level of effort on each work package.

The work breakdown structure must eventually answer two questions:

- What is to be accomplished?
- What is the necessary hierarchical relationship(s) of the work effort?

Creating Initial Estimations

Buyers should conduct the initial project estimation to ensure that they understand the full range of activities that will make up the project. This will enable the buyer to more effectively review the supplier's response to the proposal, and to also be able to select and eliminate extreme outliers. Examples of this can include schedules that are too short or cost proposals that do not make sense based on the buyer's understanding of what it ought to take to successfully develop the project. Initial estimates are often developed for:

- Size and complexity;
- Effort;
- Cost;
- Staffing (experience level and time frame needed and availability of qualified persons);
- Schedule;
- Risks;
- Computer resources;
- Facilities and test environment;
- Critical dependencies.

Identifying Initial Project Risks

Risk identification and management is a joint buyer–supplier responsibility. During the process of identifying and deciding whether or not to use a supplier, the buyer must conduct an initial investigation of the project risks based on its own experience with past projects, the supplier's past performance history, the requirements, the initial development of the WBS, organizational constraints, and the project constraints that were listed. These risks can be presented to the supplier in the RFP, and must be enhanced when placed in the supplier agreement.

It is then expected that the supplier will respond with its own risk list, and will also continue to identify and manage risks throughout the project life cycle.

Determining a System Realization Strategy

This is a function that the buyer will want to perform in order to place itself in a strong position to work with the supplier to determine potential alternative solutions or to be able to evaluate the supplier's own alternative solutions and rationale for selecting the best life cycle balanced solution.

Reviewing the Preliminary Project Plan

The buyer should put together a preliminary project plan in preparation for evaluating the supplier's responses to the RFP and to structure the monitoring of the supplier's project activities after a contract has been awarded.

Selecting the Contract Management Team

The selection of a contract management team is an essential, if not critical, factor in the success of selecting and managing suppliers. Members of the contract management team must work together to reconcile multiple views of the problem to be solved in order to ensure that the most complete understanding of the problem can be presented to the potential suppliers in both the SOW and RFP.

The contract management team will also evaluate the potential supplier's responses to reduce the risk of choosing a poor supplier based on only one point of view. The contract management team or associated representatives will support the project manager in the monitoring and control activities throughout the project life cycle. Figure 9.3 represents a sample contract management team.

Determining the Desired Buyer–Supplier Relationship

Before the acquisition strategy can be solidified, it is imperative that the desired buyer–supplier relationship be discussed and agreed on:

- Is it desirable to build a long-term relationship with the supplier?
- Is the objective to select the best contractor?
- Is the objective to select the cheapest contractor?
- How important is domain knowledge and expertise?

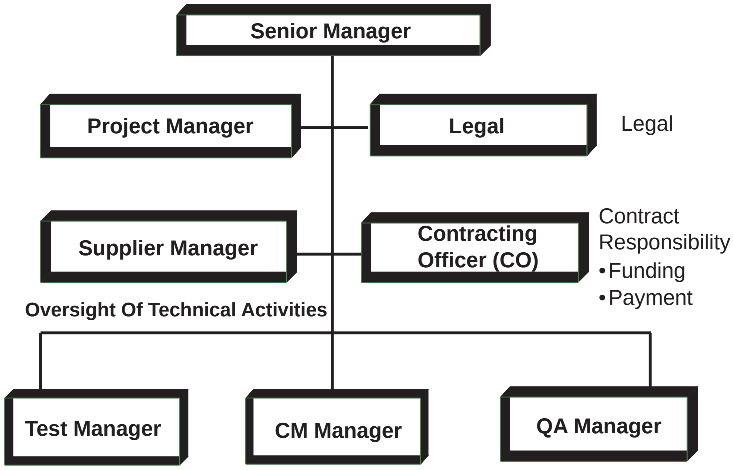


Figure 9.3 Sample contract management team.

- Is the contract for an independent supplier or for merged teams?

In theory, a buyer organization should totally open its doors to the supplier. The supplier should know what business the buyer is in, who its clients are, what its vision and strategic business direction are, and how it intends to realize that vision and strategy. In other words, the buyer should provide as much information to the supplier as possible in order to help the supplier think and act as the buyer would think. Of course, in reality, a buyer is not going to offer such an "open-door" view to just any supplier. The relationship that must be built between and buyer and supplier to justify such an open-door view must be strong and based on trust and knowledge of performance and values. Here is where that past performance history can make a real difference in how the buyer perceives the supplier.

Obviously a spectrum of buyer–supplier relationships is possible, as illustrated in Figure 9.4. Choosing the right relationship to fit the needs of the buyer leads to a more satisfying working relationship and delivery of a product or product component that indeed does satisfy the buyer. Some buyer–supplier relationship examples that may fit on that spectrum include the following:

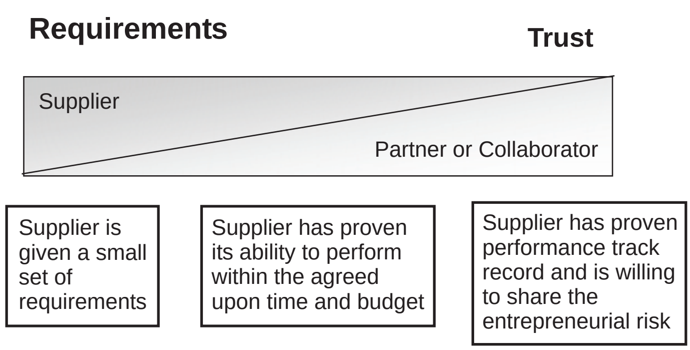


Figure 9.4 Supplier relationship.

- Provide a closely held set of requirements for a supplier to use for proof of concept. The supplier develops the product or product component strictly based on a controlled set of requirements with little to no other knowledge of the buyer. The buyer receives the product and pays the supplier with no further activities planned for in the future.
- Provide a carefully considered set of requirements along with constraints and a start on the operational concept that the supplier will take, elaborate on, validate with the buyer, architect, build, test, and deliver. The buyer will integrate that product component together with others that have been built or purchased, put its logo on the product, and ship it to its customers.
- Provide the open-door view to a supplier that is willing to share business risk with the buyer. The supplier is willing to invest research and development dollars along with the buyer. The buyer may support the supplier with training and equipment as well as insight into its future business direction. The supplier demonstrates it is willing to align its vision and business strategy with the buyer's. Oftentimes this relationship may move to the level of a partnership.

Setting Supplier Evaluation Criteria

Supplier selection implies that there is a set of established criteria that is used to evaluate suppliers. The main objective is to ensure that the evaluation of all proposals or responses to the RFP is consistent, objective, and comprehensive.

Although many criteria have been developed for supplier evaluation, it is important for the buyer to think first about the desired supplier relationship and the potential problems to be solved, and then to develop a specific set of supplier selection criteria that will facilitate the systematic choice of a supplier. This implies that the necessary contract management team members are involved, baselined supplier evaluation criteria checklists are reviewed, debates with the other CMT members about the relative importance of each criterion are held, and that the team eventually reaches consensus.

Major factors involved in proposal evaluation are the supplier's:

- Corporate experience and qualifications;
- Technical approach;
- Management processes;
- Technical processes;
- Personnel experience;
- Ability to deal with financial milestones;
- Risk management processes;
- Ethics;
- Geographical location;
- Working environment (tools and techniques);
- Ability to attract experienced developers and key personnel;
- Understanding of its business, systems, and software processes;

- Reputation for product and service quality;
- Use of its own suppliers or outsourced resources.

Along with the selection criteria, it is important for the buyer to establish the requirements it wishes the supplier to fulfill. Do not forget to let the supplier know about other requirements beyond the function ones such as:

- Internal standards (templates, forms, coding style guides, and so forth);
- External standards;
- Procedures;
- Processes;
- Configuration management system requirements including the CM tool that is preferred;
- Quality assurance expectations such as:
  - Audits of products for compliance with internal and external standards;
  - Reviews of processes and procedures to verify usability and actual usage;
  - Documentation and escalation of corrective action, as appropriate;
- Design criteria, if appropriate;
- Requirements for risk management;
- Deliverables;
- Expected project status reports and frequency;
- Requirements for personal face-to-face visits.

Preparing the Statement of Work and the Request for Proposal

Statement of Work

The statement of work is a document used by the buyer as the means to describe and specify the tasks the supplier is to perform under the contract. The SOW should:

- Clearly specify the work to be done by the supplier;
- Describe the requirements for the system needed;
- Facilitate the preparation of responsive proposals.

Typical contents of an SOW include:

- The work to be contracted out;
- The technical requirements, compliance specifications, and design criteria;
- The nonfunctional requirements including the quality factors, such as maintainability, expandability, reliability;
- The product life-cycle process requirements, which may include:
  - Preparation of a project management plan;
  - Preparation of a risk management plan;

  - Preparation of an integration and systems test plan;
  - Preparation of a quality assurance plan;
  - Preparation of a configuration management plan;
  - Monitoring and control activities;
  - Quality assurance reviews and audits;
  - Configuration management audits;
  - Systems test reports;
  - Standards (corporate, project, international, government);
- The allowable tailoring guidelines;
- The deliverables including the acceptance criteria for these end products; these deliverable might include:
  - Project plan;
  - Risk management plan;
  - Quality assurance plan;
  - Configuration management plan;
  - Integration and systems test plan;
  - Architectural specification;
  - Design documents;
  - Engineering notes;
  - Description of standards followed;
  - Description of methods used;
  - Description of process followed;
  - Technical documents;
  - Project status reports/frequency;
  - Other reports:
    - Configuration management status accounting reports;
    - Functional configuration audit reports;
    - Physical configuration audit reports;
    - Quality assurance audits and noncompliance status;
    - Issues management;
    - Risk management;
    - Test results reports;
    - Milestone reviews;
    - Technical progress reviews;
    - Management oversight reviews;
- The buyer-provided inputs in the form of goods or services that are necessary for the supplier to perform the work;
- The supplier-provided inputs in the form of certain materials that the supplier must guarantee it will provide in order to perform the work.

Request for Proposal

The contract management plan also serves as input to the RFP along with the SOW. The RFP is the heart of the contracting process and provides the basis for the contract and the work to be completed. It clearly explains all of the deliverables expected of the supplier. An RFP outline typically contains:

- Introduction;
- Statement of work;
- Supplier's quality assurance responsibility;
- Supplier's configuration management responsibility;
- Supplier's integration and systems test responsibility;
- Contract administration and clauses;
- Instructions to suppliers;
- Pricing;
- Proposal evaluation.

Conducting Supplier Capability Evaluations

A supplier capability evaluation (SCE) is a method for the independent evaluation of the processes of a supplier in order to gain insight into its management and development capabilities. The SCE investigates the necessary support functions, such as quality assurance, configuration management, systems integration and test, logistics, and even production, provided that this scope has been defined in the source selection evaluation plan and in the initial RFP as one of the criterion used to differentiate suppliers from each other.

What capability does the buyer want its supplier to have? Should the supplier have strong capabilities in software and hardware development, integration, and verification techniques? Should the supplier be aggressive when it comes to using inspections and structured walkthroughs? Does the buyer care if the supplier is competent in configuration management or not? It certainly depends on whether the buyer is intending on performing all of the project-level configuration management functions itself or expects its supplier to have that capability.

As stated earlier in the section on myths about suppliers, trying to find a CMMI® ML 5 supplier may not be the best tactic to take. An ML 2 supplier with the proven capabilities to handle the buyer's requirements and maintain a good relationship with the buyer may be an even better choice. Proof is better than promise!

It is strongly recommended that the buyer have the "option" of conducting capability evaluations on its suppliers before, during, and after contract award. If a buyer does not conduct a capability evaluation to make sure it has proof of the supplier's performance—not just the supplier's maturity level—it runs a great risk of not getting the product it expects at the desired quality and within the cost and schedule constraints that were established.

Here is an example of some questions that might guide the QA portion of the capability evaluation:

- Does the supplier have a quality policy and a QA program that is supported by its management?
- Does the supplier have a standard software/systems development process that can be tailored for a project?
- Does the supplier have an independent QA group whose responsibility includes the subcontracted project?
- Is the skill level of the QA group assigned to the subcontracted project sufficient to do the right job?
- Based on supplied samples, is the supplier's quality plan complete enough to meet contract requirements?
- Did the supplier propose that its QA group conduct project process and product quality audits?
- How do noncompliance issues get resolved in the supplier's organization? Is there an escalation process?
- What does the supplier's QA group report to senior management to assist in making quality risk decisions?
- What is the relationship of the supplier's QA group and configuration management members?
- Does the supplier have criticality levels defined for the organization so that the subcontracted projects can use the definitions to tailor their project's quality plans?

I suggested earlier that the buyer have or gain the "option" of conducting supplier capability evaluations on its suppliers before, during, and after contract award. The top Maturity Level 3 companies that I have had the chance to work with have stated that if a new supplier, especially one that provides hardware components or subsystems, wishes to become a supplier, the buyer will insist that it send a team, normally headed up by quality assurance personnel, to certify or qualify the new supplier. If this is not accepted by the potential supplier, that company will not become a supplier, plain and simple. Even when the supplier reaches "preferred supplier" status, the buyer organization still conducts at least one certification or qualification audit per year to ensure that the supplier is able to maintain its high level of technical excellence and continue to develop and deliver quality products.

Some of the leading Maturity Level 3 organizations also indicated that the procurement office or supply management office will even have its own QA personnel assigned to it to carry out the initial capability assessments or certification audits. These QA personnel will also conduct periodic quality checks on selected processes and selected work products, and conduct quality audits when the performance and progress checks indicate the supplier is not able to meet its normal levels of process performance nor deliver expected product and service quality. Such organizations realize that making such an investment to support supplier management is nominal at best compared to the returns the buyer receives from suppliers that are providing high-quality products that meets the buyer's requirements on time and within budget. To suggest that their buyer organization not conduct those audits, to save a dollar or a day of time, is unthinkable. Their response is often "How else can you ensure your supplier's success as well as your own?" I find this question an interesting question that should be asked more often.

Establishing the Supplier Agreement

Once the supplier is selected, it is necessary to establish the legal and functional way in which the buyer expects to work with the supplier. This is normally documented in the supplier agreement, which should provide the supplier with the project needs, expectations, and measures of effectiveness. The buyer must decide what contract type it wants the supplier to accept to minimize risk. Risk should not be overlooked in the selection of the type of supplier agreement. Before we go into detail about contract types and risks, however, let us first provide a brief overview of standard contract types:

Firm Fixed Price

- Cost for all deliverables is fixed.
- Normally supplier's bid is inflated to offset risk.
- Best used when requirements are well known and it is guaranteed that changes will be kept to a minimum.
- Incremental payments are normally negotiated at selected milestones.

Cost Plus Fixed Fee

- Labor and indirect cost make up the billing rate, with other direct costs being kept separate. You only pay for the actual work performed and the other direct costs.
- Fixed fee is negotiated at the beginning of the contract; for example, 10% of fee for proposed hours. This is meant to prevent suppliers from underbidding.
- Fixed fee can only change if there is an approved scope change, but the percentage does not.
- Requires increased contract monitoring.
- Fees are not tied to any contract performance capability.
- Best used when development risks are high.

Cost Plus Award Fee

- Direct labor and indirect costs make up the billing rate with other direct costs kept separate.
- Used as a way to enforce standards and procedures.
- Award fee is negotiated at the beginning of the contract.
- Fee is usually higher than the fee of a cost plus fixed fee.
- Tendency to get better qualified personnel.
- Consistent buyer appraisal of performance is required.

- The "award" approach puts a major emphasis on contractor performance and product quality.
- All of portions of the award fee could be permanently withheld if the performance or product quality does not meet expectations.
- Best used when development risks are high, but performance and quality are a must.

Time and Materials

- Fixed rate per labor hour including all overhead costs and the fee.
- Consistent appraisal of supplier performance has a higher level of importance.
- Typically used for service-type contracts including training.

Task Order Contract

- Normally a framework agreement is developed between the buyer and supplier.
- Task orders are used for project, training, or consulting activities that are part of the framework agreement.
- A rolling 3-month forecast is typically used so that the supplier can provide the necessary support to the buyer in a timely fashion.
- Better pricing is achieved by the buyer when the supplier is given a guaranteed number of days or tasks in a given time period.
- Best used when the buyer's needs are of varying complexities and risks and the full details are not known in advance.

The supplier agreement typically includes:

- Statement of work;
- Terms and conditions;
- List of deliverables, schedule, and budget;
- Defined acceptance process, including acceptance criteria;
- Identification of buyer and supplier representatives responsible and authorized to agree to changes to the supplier agreement;
- Identification of the process for handling requirements change requests from both sides;
- Identification processes, procedures, guidelines, methods, templates, and so forth, that will be followed;
- Identification of critical dependencies between the buyer and the supplier;
- Identification of the form, frequency, and depth of project oversight the supplier can expect from the buyer (this will also be dependent on the amount of foreseeable risk—greater risk will require greater oversight);
- Identification of the supplier's responsibilities for ongoing maintenance and support of the acquired products;

- Identification of the warranty, ownership, and usage rights for the acquired products.

Expectation Setting: Getting Started

Starting a contract right can prevent a lot of subsequent frustration on the part of both the buyer and supplier. A good start-up allows the team to be established early on and continue to work as a team throughout the project. Communication needs to be established early among all relevant stakeholders in order to optimize the development environment before the implementation starts.

Meetings and face-to-face contacts and discussions are the easiest and best way to set clear expectations and gain a mutual understanding of the requirements and expected performance. It is important to remember that the people from the supplier organization who put together their response to the RFP may or may not be the people who will actually work on the project. It is probable that some of them will indeed be part of the project team that will work on the buyer's project. However, it is not likely that the leader of the supplier's response team will convey all of the details and subtleties to the actual project members who will eventually get assigned to the project. Therefore, it is imperative that the buyer conduct a project orientation meeting at the beginning of the project to assist both the buyer and supplier to work toward a common goal. Continual and open communication is key to successful buyer and supplier relationships.

Habits formed at the beginning are normally carried throughout the project. Lack of clarity of purpose and objectives inevitably results in unsatisfactory performance and deficient products. Roles, responsibilities, resolution authorities, interfaces, and so on, are understood early in order to limit miscommunications.

Frequently asked supplier questions at start-up include the following:

- For whom do I work?
- What is expected of me?
- Why is it expected of me?
- What tools and facilities are available to me?
- How do I do what is expected?
- What training is available to me?
- What must I produce?
- When must it be produced?
- Who do I give it to?
- How will my product be evaluated?

These start-up questions are no different than would be expected from any other member of a project. However, if the supplier is a great distance away and possibly responding to a different culture and way of thinking, any attention paid to this contract at the start-up stage will pay for itself thousands of times over during the execution of the supplier agreement. The real payoff will be realized when the supplier's product or product component is integrated with the rest of the buyer's system and put into operation.

Monitoring Progress and Performance

Monitoring the progress and performance of one or more suppliers takes effort. It is does not happen for free, and a buyer should not rely on a supplier to successfully manage itself, not even a Maturity Level 5 supplier. As a friend of mine, Ingo Tegtmeier, CTO of KT BITS–Business and IT Solutions in Germany, is often fond of saying, "Trust is nice, verify is better!" Thinking of a supplier as a project member, despite extraordinary circumstances, such as being 1,000 miles away and in a different country, is a very good way to view a supplier. The activities of receiving status reports from the project member and holding project meetings or project reviews to discuss progress, risks, problems, and next activities all apply to a supplier (Figure 9.5).

Monitoring a supplier's progress and performance necessitates a look at the management, technical, quality, and process activities. Technical reviews typically include performance of these tasks:

- Visibility into the needs and desires of the project's customers and end users is provided.
- Technical, cost, staffing, and schedule performance are reviewed.
- Technical issues are communicated and resolved.
- Critical computer resources are reviewed.
- Technological risks that may arise based on the technologies employed are reviewed.

Management reviews typically cover:

- Critical dependencies;
- Nonconformance to the supplier agreement;
- Project risks;
- Feedback on supplier performance;
- Changes to the supplier's statement of work, terms and conditions, and other commitments.

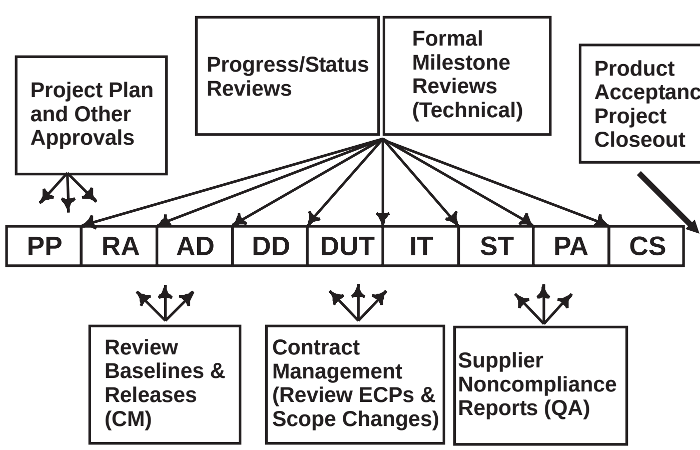


Figure 9.5 Monitoring progress and performance activities.

Quality management reviews typically cover:

- Quality plans;
- Quality goals for the project and organization;
- Quality assurance objective evaluations;
- Trend analyses on noncompliance categories;
- Peer review records;
- Test records;
- Configuration baseline audits;
- Status accounting reports.

Process management reviews typically cover:

- Process improvement initiative progress;
- Noncompliance findings and resolution;
- Effectiveness of processes measures;
- Improvements in process that resulted in measurable business objectives improvements,
- Process improvement suggestions versus how many have been implemented.

In situations where the supplier's processes must be closely aligned with the buyer's, and the supplier deliveries are considered a critical dependency for buyer satisfaction of its customers, the CMMI® suggests a more focused view of the supplier's processes and work products.

Selected processes or subprocesses from the supplier's defined processes for the project should be monitored as an indicator of meeting the project's constraints and the potential to deliver the high-quality products desired. The processes selected for monitoring should include:

- Engineering;
- Project management;
- Support processes critical to successful project performance such as quality assurance and configuration management.

In like fashion, selected work products produced by the supplier should be evaluated to help detect issues as early as possible that may affect the supplier's ability to satisfy the requirements of the agreement. The work products selected for evaluation should include critical products, product components, and work products that provide insight into quality issues as early as possible.

Work products should be evaluated to ensure the following:

- Derived requirements are traceable to higher level requirements.
- The architecture is feasible and will satisfy future product growth and reuse needs.

- Documentation that will be used to operate and to support the product is adequate.
- Work products are consistent with one another.
- Products and product components (e.g., custom-made, off-the-shelf, and customer-supplied products) can be integrated.

It is worthwhile to again share activities of the leading buyer organizations I have had the privilege of working with throughout the world. For most of the buyers' suppliers on whom the buyers are critically dependent, these leading buyer organizations conduct supplier management meetings at the supplier's site at least every 2 weeks and insist that supplier representatives attend management meetings the other weeks at the buyer's site. This was done to ensure that the critical processes were being followed and to increase confidence that the suppliers would be able to keep their commitments and deliver high-quality products and product components. Whenever a buyer determined that its supplier was not reaching its progress and performance goals, the buyer took immediate steps to work with the supplier with the intent of helping it get its performance back up to expectations. The buyer's first goal was to help its top suppliers succeed. One buyer organization even went to the supplier's site and participated with them in a kaizen event to speed up the production process.

Of course, contracts are created and signed to protect both the buyer and supplier. Many leading companies keep detailed records about their suppliers and their past performance history. A supplier whose performance rating is low may be offered assistance. If that performance rating continues to remain low, the buyer may keep the supplier on its list for some predetermined time to see if the supplier can fix the business problems it faces. However, if the negative trend continues, the buyer will often have to exercise the choice of dropping the supplier from its list of preferred suppliers.

Acceptance Testing

The objective of acceptance testing is to demonstrate that:

- The product is complete.
- Design problems encountered during development have been solved.
- The design meets the specifications.
- The system's operational effectiveness and suitability have been demonstrated.
- The system is ready for operational use:
    - Functional requirements;
    - Performance requirements;
    - Operational use profile;
    - Recovery requirements.
- The system is both usable and reliable.
- Installation documentation and user documentation address the actual system that has been developed.

The tests script and cases for the acceptance test should be created by the buyer or at the very least in conjunction with the supplier. Note that the supplier should not develop the acceptance tests in isolation.

- The preparation for acceptance testing begins at the requirements stage of the life cycle.
- Acceptance criteria should be discussed and agreed on in advance of the actual acceptance testing.
- Acceptance “success” criteria must be documented.

Transitioning from Supplier to Buyer

Transitioning from the supplier’s environment to the buyer’s environment for integration and buyer systems testing and/or acceptance testing requires thought and up-front planning. The buyer needs to determine what it expects to receive or have happen in the way of deliverables and service to successfully transition the product or product component from the supplier to the buyer. The following questions provide a high-level summary of the checklist that can be developed to assist the transitioning of the product or product component from the supplier to the buyer. The following is an example of a “transitioning from supplier to buyer” checklist.

1. What do you expect to be delivered?
   - Product and product components;
   - Data;
   - Test cases and test scenarios;
   - Test results;
   - Architectural specifications;
   - Design specifications;
   - Operations documentation;
   - Installation manuals;
   - Operator’s manuals;
   - Maintenance manuals;
   - Quality audit reports;
   - Functional configuration audits,
   - Physical configuration audits;
   - Release notes or a version description document;
   - Key area experts;
   - Competence transfer;
   - Risk lists;
   - In-house tools critical to support system execution;

- Compiler;
- Operating system.

2. What environment must the buyer provide to be able to accept supplier's product or partial system?
    - Space;
    - Simulators;
    - Emulators;
    - Simulation packages;
    - Mock-ups;
    - Diagnostic equipment;
    - Computers, printers, copy machines, scanners, and so on;
    - Networking;
    - FTP servers;
    - Configuration management control systems:
        - Configuration management manager,
        - Library;
    - License for tools for suppliers;
    - Power;
    - Service environment;
    - Special environments (e.g., RF chambers);
    - Environmental chambers (heating and cooling);
    - Tools for reloading software;
    - Target software and hardware for integration;
    - Ability to test interfaces.
3. What people must you provide to gain acceptance of the supplier's product?
    - Systems testers;
    - Requirements engineers;
    - Lead designer;
    - Quality assurance;
    - Configuration management;
    - Operators;
    - Error manager and engineers;
    - Troubleshooter;
    - Representative users.
4. What people must the supplier provide during the acceptance period and for how long?
    - Lead analyst;
    - Systems engineer;
    - Systems test team;

- Troubleshooter;
- Error manger;
- Technical expert;
- CM manager;
- Requirements analyst;
- Quality assurance.

5. What will happen if problems are found?
   - *Critical problems (system does not work)*. Supplier required to fix all critical problems in defined period of time or face penalty. If the supplier cannot solve problem, buyer can use third party and charge back to supplier.
   - *Intermediate problems (system has limited ability)*. Is workaround available? Is the time frame required to actually fix the problem acceptable?
   - *Small annoying problems*. Fixed according to contract.
6. What other support do you expect of your supplier? (Also: How long will the support be required? Will this be covered in a separate contract or is this support covered in this contract?)
   - Training in critical system functions;
   - Training in normal operations use;
   - Training in using user's manuals;
   - Training in troubleshooting;
   - On-site support or real-time availability of technical expertise until system has passed maintenance criteria (MTBF);
   - Cooperation with other suppliers to get total system to run,
   - Joint support for problem solving (1 year);
   - Trouble reporting system;
   - Training for new people or on new features for 6 months to 1 year or technology transfer.
7. What type of maintenance agreement do you seek?
   - On-site support for short answers;
   - Help desk;
   - Help with any critical or intermediate problems discovered within 1 calendar year from acceptance and sign-off by the buyer;
   - Operating system support.
8. What about future enhancements?
   - One free enhancement during the first year following acceptance and sign-off by the buyer,
   - Notifications of enhancements and chance of beta site testing for next 1 to 2 years;
   - On-site support for integration and switchover to enhanced system capability.

Supplier: One of the Project Team Members

It is my opinion that a supplier should be thought of just as if it were one of the project team members (Figure 9.6). True, the supplier may be another company of 1,000 developers about 4,000 miles away, but, like the project member, the supplier is given an assignment with expected milestones and deliverables. And, like the project member, the supplier needs to provide status reports describing accomplishments, issues, risks, actual problems, considerations for the next reporting period, direction from the project manager, a list of what did not get accomplished, and information about what is planned to be accomplished during the next reporting period.

The supplier's management and technical activities and work products should be evaluated on both a periodic and an event-driven basis in accordance with the supplier agreement. The buyer's project manager should be prepared to make recommendations for performance improvement as necessary. The buyer must also be prepared to provide either an award or a penalty, as appropriate, in accordance with the contractual agreement.

Project management reviews should also be set up to discuss any changes to the supplier's statement of work, terms and conditions, and any other commitments. When contracts are involved, changes to the baselined requirements will inevitably involve more money if they are out of scope with the agreed-on requirements. For this reason, it is best to stipulate up front in the contract that there is only one person truly authorized within the buyer's office to suggest changes, and that all other suggestions must be disregarded and are not officially authorized unless that person puts it in writing. This ensures that only one face is presented to the contractor. That person authorized to put in writing actual changes is typically the contract administrator or contracting officer. This is standard policy in government contracting, and well worth adopting to reduce legal issues later.

I mentioned earlier that the requirements should include the buyer's demand regarding configuration management and quality assurance. These project management support areas should not be excluded in the periodic progress meetings and status reports. It is up to the buyer to ensure that the supplier is responsible for

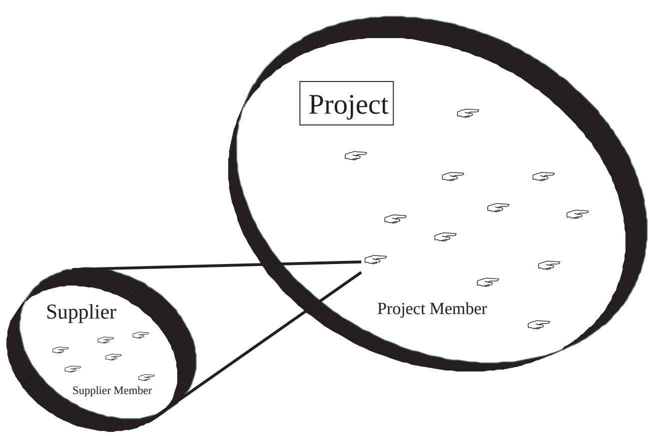


Figure 9.6 Supplier viewed as another project member.

the integrity of the work products or product components that it is building. It is also up to the buyer to ensure that the supplier, through its processes, peer reviews, and tests, is able to product the necessary product quality that is expected by the project.

Although the supplier's progress reports may be very encouraging, remember that it is the buyer that must integrate the product component(s) produced by the supplier, put its brand name on the entire integrated product, and advertise it as quality. Therefore, acceptance testing should be defined and performed by the buyer. It can be done with support from the supplier, but it remains the buyer's responsibility for the development of the acceptance criteria and for the responsibility for independent acceptance testing.

The product that is delivered by the supplier must function in the buyer's operational environment, and it must work as the buyer intended it to work. It is perfectly acceptable for the buyer to ask for systems test results from the supplier along with the results of the functional configuration audit and the physical configuration audit before the buyer even agrees to enter the transition phase, and certainly before the product or product component transitions from the supplier to the buyer.

Supplier agreement management is about deciding:

- What is needed to manage your project's activities?
- What part of your project's requirements would you like to be satisfied outside of your project boundaries?
- What is the effort and cost of acquiring a COTS solution?
- What is the effort and cost to manage a supplier?
- Where outside of your project's boundaries will you find the most balanced solution for your needs?

Summary

The use of suppliers has become a common way of life for many system developers. To be able to guarantee a fully integrated, working, and effective product or product component that the business unit is willing to put its logo on, project managers must be able to control the project's suppliers as if they were just another member of the project. The term *supplier* is used to identify an internal or external organization that develops, manufactures, tests, or supports products being developed or maintained that will be integrated into the buyer's product or will stand alone and be delivered to the buyer's customer. Applying the basic principles of project management, quality management, and engineering addresses the needs of the project to effectively select and manage those portions of the work that are conducted by suppliers.

References

[1] Wikipedia, "Outsourcing," section on "Deciding to Outsource," http://en.wikipedia.org/wiki/Out_Sourcing#Deciding_to_outsource.

[2] Wikipedia, "Open Source," section on "The Open Source Definition," http://en.wikipedia.org/wiki/Open_source#The_Open_Source_Definition.

CHAPTER 10

Enabling Project Managers to Manage and Control Better Through Integrated Project Management

Integrated project management (IPM) can be thought of as project management based on the organization's set of standard processes. The purpose statement in the CMMI® for the Integrated Project Management PA is "Establish and manage the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes" [1]. Integrated project management involves:

- Establishing the project's defined process at project start-up by tailoring the organization's set of standard processes (*project start-up* is an added term to the CMMI® glossary and plays a significant role in organizational process focus process area especially for Specific Goal 3);
- Managing the project using the project's defined process;
- Establishing the work environment for the project based on the organization's work environment standards;
- Using and contributing to the organizational process assets;
- Enabling relevant stakeholders' concerns to be identified, considered, and, when appropriate, addressed during the development of the product;
- Ensuring that the relevant stakeholders perform their tasks in a coordinated and timely manner to:
    - Address product and product component requirements, plans, objectives, problems, and risks;
    - Fulfill their commitments;
    - Identify, track, and resolve coordination issues.

Figure 10.1 illustrates the parallelism between tailoring the organization's set of standard processes from the organizational process assets to make the project's defined process and that of starting with the organization's work environment standards and defining the project's work environment. Both concepts must be understood in order to properly implement them on any given project. We next examine the project's defined process and the project's work environment in more detail.

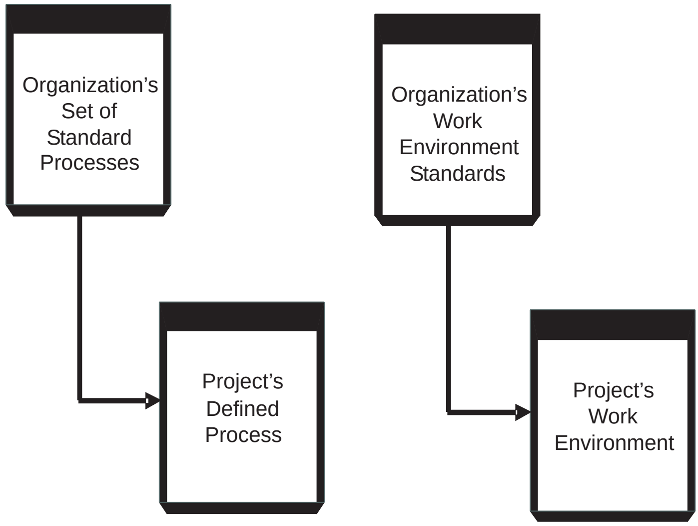


Figure 10.1 A project's defined process and work environment.

The Project's Defined Process

A project's defined process should be based on a minimal set of items including:

- The organization's set of standard processes;
- The customer requirements;
- The product and product component requirements;
- Commitments made both internally and externally;
- Organizational process needs and objectives;
- The organizational support environment;
- The operational environment and the business environment.

The task of managing the project's effort, cost, schedule, staffing, risks, and other factors is tied to the tasks of the project's defined process. Integrated project management also means that all of the activities associated with and supporting the project are coordinated, including the technical activities of requirements development, design, development, and verification and validation. It also includes the project management support activities of quality assurance, configuration management, marketing, and training.

A short story is offered here in support of the next point. A few years ago, I was asked to spend 1 to 1.5 days at an organization to give them a vote of confidence as to whether they should conduct a Maturity Level 3 appraisal in the next 3 months. I was not going to be that lead appraiser. I agreed to the task and then wondered how in the world I could examine enough of the organization's processes and conduct a sufficient number of interviews in 1.5 days to give this organization the vote of confidence that they sought, and still be accurate enough to live with myself.

After some time spent deep in thought, the answer came to me. The answer lies in the description of integrated project management itself. While at this organization, I asked that I be able to interview three or four project managers of the representative projects that would be used in the Maturity Level 3 appraisal. When each project manager came into the interview room, and we had introduced ourselves, I asked them one question. "How do you manage your project in an ML3 environment/organization?" The answer is actually only a 1-minute answer, but it is a clear indicator for ML 3 organizations. The project managers simply needed to state the steps for establishing the project's defined process, using the organization's process assets to help with the estimation. They needed only to indicate that they would examine the organization's measurement repository for any measures that would help guide the project manager, state how the project plan would be developed based on that, and how the project would be managed against the integrated project plan. It turns out that none of the project managers could answer that question, even when given broad hints. My recommendation to the senior management of that organization was to wait 6 months instead of 3 months to ensure that the concepts found in integrated project management would truly be institutionalized.

The suggested steps for establishing the project's defined process include the following:

- Select a life-cycle model from the organization's process assets.
- Select the standard processes from the organization's set of standard processes that best fit the needs of the project.
- Tailor the organization's set of standard processes and other organizational process assets according to the tailoring guidelines to produce the project's defined process.
- Incorporate lessons learned from other current and completed projects in the organization by examining the process asset library as appropriate.
- Document the project's defined process.
- Conduct peer reviews of the project's defined process.
- Revise the project's defined process as necessary.

Going back to my example, from this point the project managers should have continued to state how the organizational process assets were used for helping to create and manage the project planning activities. For example, this scenario would fit the IPM description.

We would use the organization's measurement repository in this way to estimate the project's planning parameters:

- Use appropriate historical data from this project or similar projects.
- Record similarities and differences between the current project and those projects whose historical data will be used.
- Independently validate the historical data.
- Record reasoning, assumptions, and rationale used to select the historical data.

The project's defined process should cover all of the activities for the project and its interfaces to relevant stakeholders including:

- Project planning;
- Project monitoring;
- Requirements development;
- Requirements management;
- Supplier management;
- Configuration management;
- Quality assurance;
- Risk management;
- Decision analysis and resolution;
- Product development and support;
- Solicitation.

When performing the estimation that was described for the Project Planning and Project Monitoring and Control PAs in earlier chapters, integrated project management guides the project manager and team members to use the organization's measurement repository. Appropriate data from similar projects can be used to support the estimates the project has conducted on its own. Each project is strongly encouraged to independently validate the historical data and record the assumptions and rationale used to select the historical data as stated earlier.

It is my experience that many true Maturity Level 3 organizations have built a human interface between the project managers and the organization's process assets to support this ML 3 thinking. One such human interface allowed the project manager to only input his project's descriptive characteristics, and the human interface searched the organization's process asset database for the project manager. This resulted in identifying three or four other projects' profiles that most closely matched this project. References to any measurements that were used on those projects were also provided. The project manager was only required to study the samples provided to him, make the appropriate choice, document the rationale, and ensure that the project conducted proper estimation activities in order to validate the historical data from the other projects.

The Project's Work Environment

Work environment standards allow the organization and projects to benefit from common tools, training, and maintenance, as well as cost savings from volume purchases (economies of scale). Work environment standards address the needs of all stakeholders and enable consideration of items such as productivity, cost, availability, security, safety, ergonomic factors, and workplace health.

The work environment for a project should be derived from the organization's work environment standards. It should comprise an infrastructure of facilities, tools, and equipment that people need to perform their jobs effectively and efficiently in support of business and project objectives. The work environment should be selected with the same rigor as is applied to any other product development project. Trade-offs normally have to be made among performance, costs, and risks. The following are examples of each:

- Performance considerations such as interoperable communications, safety, security, and maintainability;
- Costs may include capital outlays, training, support structure, disassembly and disposal of existing environments, and operation and maintenance of the environment;
- Personal choice may have to be overridden in favor of standard organizational tool sets for large organizations and those belonging to international enterprises.

Examples of equipment and tools include the following:

- Office software;
- Decision support software;
- Project management tools;
- Requirements management tools and design tools;
- Configuration management tools;
- Evaluation tools;
- Prototyping or production equipment;
- Test and/or evaluation equipment;
- Communication-related tools and resources such as meeting rooms, e-mail, Web sites, and video and Web teleconferencing capabilities;
- Software productivity tools (e.g., workflow);
- (Proprietary) engineering or simulation tools;
- Computing resources;
- Operational support staff (e.g., help desk).

Work environment standards and instantiations of work environments on projects must also include maintaining the qualification of the various components, software, and hardware that compose the work environment. Examples of these components include:

- Databases;
- Computers;
- Printers;
- Networks;
- Tools;
- Test equipment;
- Appropriate documentation.

Qualification of software includes appropriate certifications. Hardware and test equipment qualification includes calibration and adjustment records and traceability to calibration standards.

Plans That Affect the Project

Integrated project management also has a strong implication that is not present in either the Project Planning process area or the Project Monitoring and Control process area. In project planning and project monitoring and control, representatives of different groups, departments, or disciplines develop or help the project to develop supporting plans that affect the project, but they are developed as if by independent suppliers.

In applying the principles of integrated project management, these “plans that affect the project” are developed by a more focused and integrated set of individuals who not only focus on the content of a supporting plan, but also on how well it is integrated with the other supporting plans. Each supporting plan complements the other supporting plans by tone and level of content. If the configuration management plan indicates that a QA Responsible will conduct the functional and physical configuration audits, then the quality assurance plan will clearly spell out how the QA Responsibles will carry out those configuration management audits. If the project plan is 2,000 pages long, then the project’s quality plan will not be merely 3 pages indicating that quality is a good thing.

There is another aspect of integrating all of the plans that affect the project that comes from the purpose statement of IPM. All of the plans are based on a tailored version of the organization’s set of standard processes. It is no longer acceptable for other engineering disciplines and support groups to develop their corresponding plans based on standards that are not acceptable at the organizational level. The process descriptions for each engineering discipline and support group are available for all to study and understand to enhance the communication between the relevant stakeholders in the development, review, and implementation of the integrated project plan.

Integrated Project Plan

The integrated project plan should incorporate the definitions of measures that will be used to manage the project, identify and analyze overall project risks, and schedule all tasks in a sequence that takes into consideration critical development factors. The integrated project plan should also identify how conflicts will be resolved that will inevitably arise between stakeholders involved in the project.

Managing the project using the integrated plans implies tracking risks, commitments, and the adequacy of the support environment on a periodic basis. This also implies that critical dependencies from all relevant stakeholders, including external suppliers, will be identified, tracked, and negotiated. Corrective action is again taken as necessary.

To keep the cycle of continuous process improvement going, it is necessary to capture lessons learned and contribute work products, process and product measures, and documented experiences to the organizational process assets. These experiences are placed back into the organizational measurement repository.

It is expected that organizations that operate within the requirements and constraints of ML 3 and state they satisfy the practices in integrated project management will routinely:

- Propose improvements to the organization's process assets;
- Store validated process and product measures in the organization's measurement repository;
- Submit candidate good practices for possible inclusion as best practices in the organization's library of process assets;
- Document lessons learned from the project for inclusion in the organization's library of process assets.

Coordination of Relevant Stakeholders Based on the Project's Defined Process

In Chapter 6, the concepts of stakeholders, relevant stakeholders, and a stakeholder plan were presented. Obviously, those concepts based on project management at Maturity Level 2 are necessary for an organization that wants to operate at Maturity Level 3. "So what are the differences?" the reader may be asking. Most importantly, as elaborated on earlier, all of the internal relevant stakeholders are basing their planning, communication, and implementation efforts on the project's defined processes, tailored from the organization's set of standard processes. This should greatly enhance the communication among the relevant stakeholders that is necessary for the success of the project. Relevant stakeholders also have a stronger spotlight on them. Either the project has conducted a process audit of its supplier to ensure that process performance is documented along with any maturity or capability level the supplier is advertising, or it is insisting that the supplier is following the project's defined process.

Critical dependencies that are identified between or among any of the relevant stakeholders are closely managed especially if the critical dependency is coming from an external supplier. Looking back at Chapter 9, we can see that a critical dependency on an external supplier may necessitate the monitoring of pertinent supplier processes or subprocesses and related work products to ensure the delivered product or product component has the required functionality and quality established by the contractual agreement between the project and the supplier.

All critical dependencies and the commitments for them must be tracked. This includes:

- Evaluating the potential effects of either late or early completion of critical tasks on planned future activities and milestones,
- Ensuring that necessary reconciliation and resolution processes as initially described in Chapter 6 are understood by all relevant stakeholders and used appropriately when needed.

Summary

Integrated project management represents the evolution of the basic project management functions. Based on the organization's set of standard processes, integrated project management starts by developing the project's defined process, incorporates other plans that support and contribute to the project's success, and guides the development of the integrated project plan that will be used to manage and control the project to successful completion. It establishes the project's work environment based on the organization's work environment standards.

Project managers manage their project based on these process standards, work environment standards, and project plans that result from them. All disciplines and support groups involved with the project work together to satisfy the technical requirements and the customer's requirements to ensure the product or product component works in the operational environment for the end users that must use it.

Summary of Chapters 6 Through 10

Project managers today, perhaps more than ever, are under great stress to understand the requirements of the customer and sponsor, develop or lead the development of a work breakdown structure, provide estimates by which to guide the project development, and manage the project to successful completion within the constraints of cost, schedule, performance, quality, functionality, risk, and customer satisfaction. They must be able to identify risks, analyze them, develop risk mitigation plans with a risk strategy, and monitor the risks in an attempt to prevent them from becoming problems that could hurt the success of the project.

Project managers must be able to coordinate the efforts of different support groups such as quality assurance and configuration management, different disciplines such as software engineering or hardware engineering, or other departments such as marketing or finance. They must be able to manage their own project members and remotely control suppliers, which should be considered project members. They must be able to direct the technical efforts of their project team members but do so with proper interpersonal skills that must be learned on top of all of the other tasks they are being asked to do.

In addition, project managers must identify, understand, and closely manage critical dependencies and the commitments to satisfying them both for internal and external relevant stakeholders.

Today, more than ever, project managers need the processes, procedures, and guidance of the project management and project management support functions that have been described in Chapter 6 through this chapter and are emphasized in the CMMI®.

Reference

[1] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI® Second Edition, Guidelines for Process Integration and Product Improvement,* Reading, MA: Addison-Wesley, 2007.

PART III

Engineering

CHAPTER 11

The Recursive Nature of Requirements Engineering

This chapter presents the "recursive" nature of the total requirements gathering and analysis process. It includes concepts from the initial identification of stakeholders, to deriving requirements, to validating requirements at all stages.

Requirements Development

Requirements are the basis for design. Requirements, gathered according to the CMMI®, address the needs of all relevant stakeholders including testing, quality assurance, database, software, hardware, and systems. Requirements may be focused on specific life-cycle phases. They may be focused on quality attributes such as maintainability, expandability, or reliability. Requirements may also address constraints caused by the selection of certain design solutions such as the use of an off-the-shelf database.

What Are Requirements?

The following list may provide some insight into the question "So what are requirements?" Requirements contain or provide:

- Customer's needs, expectations, and measures of effectiveness;
- Items that are necessary, needed, or demanded;
- Implicit or explicit criteria that must, should, or might be met;
- System, hardware, and software information;
- Requirements derived from other requirements during analysis and operational concept and operational scenarios development;
- Services the system is to provide;
- System behavior;
- Circumstances under which the system is to operate;
- Application domain information;
- Constraints on the systems operations;
- Systems properties or attributes;
- Constraints on the development process of the system.

Requirements might describe the following:

- *A user-level facility*. The word processor must include a spell checker and correction command.
- *A very general system property*. The system must ensure that personal information is never made available without authorization.
- *A specific constraint on the system*. The sensor must be polled 10 times per second.
- *How to carry out some computation*. The overall mark is computed by adding the student's examination, project, and coursework marks based on the following formula:

$$\text{Total mark} = \text{Exam mark} + 2 \times \text{Project mark} + 2/3 \times \text{Coursework mark} \qquad (11.1)$$

- *A constraint on the development of the system*. The system must be developed using the C++ programming language.
- Requirements invariably contain a mixture of problem information, statements of system behavior, system properties, design constraints, and manufacturing constraints. This can and normally does result in conflicts that must be negotiated and resolved, as discussed later in more detail.

Requirements come from many sources and must be identified and evaluated to determine the true impact they may have on the project or the impact the project could have on them. Stakeholders, as defined in Chapters 4 and 6, provide the wants, needs, constraints, and interface requirements. It is important for any project to know what the sources of requirements are and to control those sources through requirements gathering and management of requirement's change requests.

These requirements sources include:

- End user;
- Marketing;
- Surveys;
- Systems engineering;
- Hardware engineering;
- Software engineering;
- Existing systems and specifications;
- Standards;
- Industry studies;
- Academic research;
- Prototyping;
- Simulation;
- Modeling;
- Quality assurance;

- Configuration management;
- Testing;
- Database administrators;
- Regulatory agencies;
- Competitors' products and services;
- Operational environment,
- Application domain,
- Managers;
- Manufacturers;
- Logistics personnel.

Examples of sources of requirements that might not be identified by the customer include the following:

- Business policies;
- Business environmental requirements (e.g., laboratories, testing facilities, and information technology infrastructure);
- Technology;
- Legacy products or product components;
- Reuse product components.

Requirements are commonly placed into categories. Three examples are:

- Product requirements that define the technical criteria that must, should, or might be met by the delivered product;
- Project requirements that stipulate resources that will be made available, and how different aspects of the project will be carried out;
- Process requirements that indicate standards, procedures, methods, languages, and engineering and management processes that must be followed.

The needs of stakeholders are the basis for determining customer requirements. Frequently stakeholders' needs, expectations, and constraints are poorly identified or conflicting. In part, this is because developing organizations expect the stakeholders to "know" exactly what they want the product or product component to look like and act like, how it is supposed to interact with its intended environment, and how long it is expected to exist as a product. In today's fast-paced, technology-driven world, it is very unlikely that the stakeholders will have such insight.

The responsibility to help these stakeholders understand and "discover" what their requirements really are is the first challenge that must be confronted if the product development is to be successful and the delivered product declared useful in its intended environment. Environmental, legal, and other constraints that may be external to the customer must also be applied when creating and resolving the stakeholders' needs, wants, constraints, and interface requirements into a set of customer requirements.

Elicitation of Requirements

To go beyond the basic activity of collecting stakeholders' needs, expectations, constraints, and interfaces for all phases of the product life cycle, the requirements development process area suggests that requirements should be *elicited*. Eliciting goes beyond collecting requirements. Eliciting implies proactively identifying additional requirements not explicitly provided by stakeholders or customers. Techniques that can be used to identify and elicit stakeholders' needs include:

- Dialogue;
- Brainstorming;
- Scenario reviews;
- Questionnaires and interviews;
- Technology demonstrations;
- Operational walkthroughs;
- End-user task analysis;
- Models;
- Quality function deployment;
- Simulations;
- Prototypes;
- Use cases;
- Market surveys;
- Business case analysis;
- Observations of existing systems;
- Extractions from sources such as documents, standards, and specifications;
- Customer satisfaction surveys;
- Reverse engineering;
- Beta testing;
- Interface and technical control working groups.

Figure 11.1 illustrates a more detailed process of eliciting requirements.

Objective setting. The overall organizational objectives should be established at this stage. These include general goals of the business, an outline description of the problems to be solved, why the system may be necessary, and the constraints on the system such as budget, schedule, and interoperability constraints.

Background knowledge acquisition. This is a very important stage in which the requirements engineers gather and understand background information about the system. This includes information about the organization where the system is to be installed, information about the application domain of the system, and information about any existing systems that are in use and that may be replaced by the systems being specified.

Knowledge organization. The large amount of knowledge that has been collected in the previous stage must be organized and collated. This involves identifying system stakeholders and their roles in the organization, prioritizing

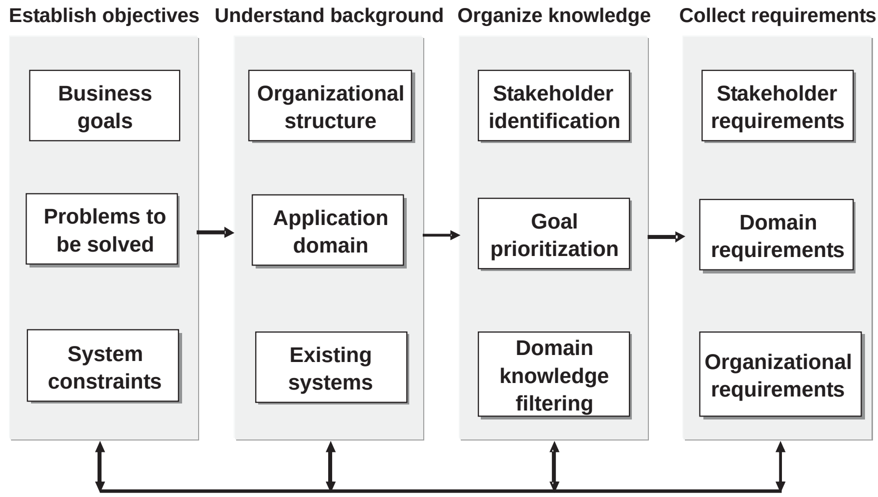


Figure 11.1 Generic requirements elicitation process. (*From:* [1]. © 1998 John Wiley & Sons, Inc. Reprinted with permission.)

the goals of the organization, and discarding domain knowledge that does not contribute directly to the system requirements.

Stakeholder requirements collection. This stage is what many people think of as the elicitation stage. It involves consulting system stakeholders to discover their requirements and deriving requirements that come from the application domain and the organization that is acquiring the system.

The definition of "customer requirements" was first provided in Figure 4.2 in Chapter 4 and repeated here as Figure 11.2. The list of stakeholders found on the left within the oval is not meant to be exhaustive, but merely indicates that stakeholders are found inside and outside of the organization and must be seriously considered based on their potential influence on the project.

It is helpful to understand the definition of a customer from the CMMI® point of view. A *customer* is the party (individual, project, or organization) responsible for accepting the product or for authorizing payment. This has been updated for CMMI® v1.2. The customer is external to the project, but not necessarily to the organization. Customers are a subset of stakeholders as Figure 11.2 illustrates. Customer requirements then are the result of eliciting, consolidating, and resolving conflicts among the needs, expectations, constraints, and interfaces of the product's relevant stakeholders in a way that is acceptable to the customer.

The customer requirements as defined in Chapter 4 and reillustrated in Figure 11.2 represent the common understanding of what will satisfy the stakeholders. This input must be consolidated and conflicts must be removed before being captured as customer requirements. Notice that I said "conflicts must be removed." There will be conflicts and it is important to thoroughly understand stakeholder relationships before attempting to resolve them. Stakeholders, by their very nature, want what they want in a product or product component and really do not care what some other stakeholder wants. Often these conflicts show up as nonfunctional requirements.

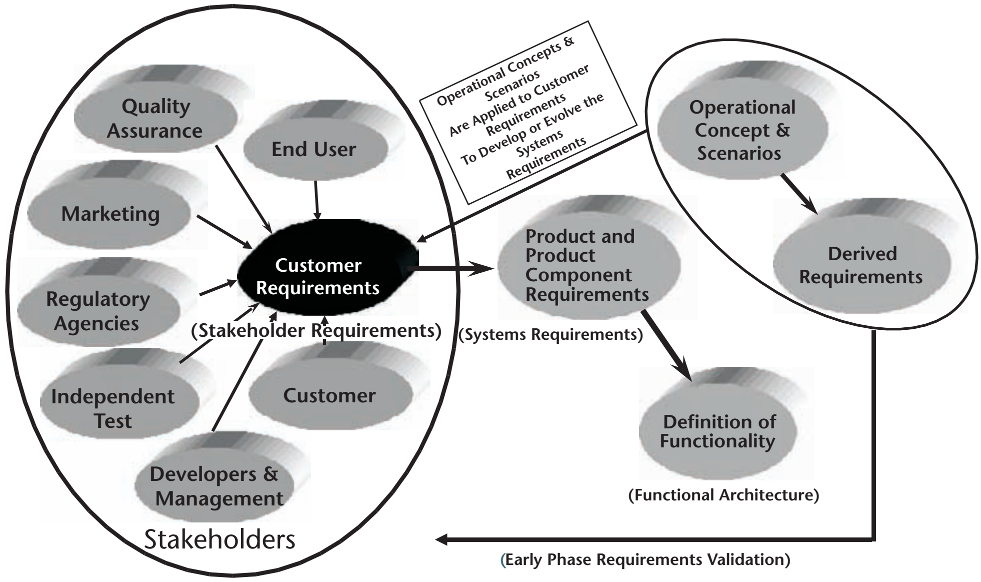


Figure 11.2 Customer, product, and product component requirements and customer requirements.

Nonfunctional Requirements

Nonfunctional requirements are requirements that are not specifically concerned with the functionality of a system. They place restrictions on the product being developed, the development process, and the external constraints that the product must meet. Nonfunctional requirements define the overall qualities or attributes of the resulting system and place restrictions on how the user requirements are to be met. These nonfunctional requirements are often of critical importance and functional requirements may need to be sacrificed to meet these nonfunctional constraints. Nonfunctional requirements are defined because of the need of the user to achieve certain goals such as:

- Budget;
- Organizational policies;
- Need for interoperability with other software or hardware;
- Need for certain development processes to be followed;
- Need to care for external factors such as safety and security regulations.

The following examples may help to illustrate the possible conflicts:

- A requirement for a certain level of performance may be contradicted by reliability and security requirements that rely on processor capacity to carry out dynamic system checking;
- A requirement on space utilization of the system may be contradicted by another requirement that specifies that a standard compiler that does not generate optimized code be used.

Product quality can be described through a number of quality factors or attributes such as maintainability, expandability, reliability, portability, reusability, efficiency, and interoperability. Each factor has several attributes that describe it called criteria. Each criterion has associated with it several metrics that, taken together, quantify the criterion. It is important to specify quality requirements into the project early in the development process.

The basis for product quality is the user's need for certain product characteristics after the product is put into use—that is, its fitness for the intended use. During the elicitation of requirements it is important to listen closely to the user's concerns and translate those concerns into quality factors or quality requirements or nonfunctional requirements. When the user is concerned about how secure the system is, the person or person's eliciting the requirements should list integrity as one of the important quality factors. If the concern is over how easy is it to repair the system, then maintainability should be listed as one of the important quality factors. If system expansion is the concern, then expandability should be listed as one of the important quality factors.

The Requirements Development process area does not explicitly focus on the quality factors in its descriptions, but the Technical Solution process area mentions them when it comes to designing the system. It is clearly pointed out that one cannot "test in" portability at systems test time if the portability quality factor is not designed in and verified and validated before the system enters systems test.

Once the conflicts are removed, the stakeholders' wants, needs, constraints, and interfaces are captured as "customer requirements" (see Figure 11.2), the process of refining those requirements through analysis and validation starts and continues in a recursive manner all the way through to the definition of alternative solutions that will eventually lead to the "best" solution being implemented.

This chapter focuses primarily on the requirements side but will show the overlap with the evolution of the alternative solution that will be discussed in more detail in the next chapter on alternative solutions.

Product or Product Component Requirements

The next step in the evolution of the requirements into a product or product component is the translation of the customer requirements into the product or product component requirements. Product and product component requirements define what the system is required to do and the circumstances under which it is required to operate. They define the services that the product or product component should provide and establish constraints on how they will operate. Product and product component requirements *should include technical requirements* and the criteria that will be used to verify that the products satisfy the requirements.

The process of determining what the system is intended to do is also referred to as the definition of functionality or more commonly referred to as *functional analysis*. The definition of functionality should include actions, sequences, input, outputs, and other information that clearly describes how the product will be used. When we define functions, place them into logical groupings, and associate them with the requirements, we refer to this as the functional architecture.

Operational Concept and Operational Scenarios

The recursive nature of requirements focuses on deriving a detailed and precise set of technical requirements that, if implemented, will satisfy the customer requirements. To accomplish that, the requirements team should analyze the customer requirements concurrently with the development or refinement of the operational concept. Analysis at one level of requirements makes sure that they are necessary and sufficient to meet the objectives of higher levels of the product hierarchy. This recursive analysis of requirements may produce derived requirements that result from design decisions and should also address the cost and performance of other life-cycle phases to the extent possible along with the organization's business objectives.

The *operational concept* is a general description of how the system operates, where in the operating environment the system will be distributed, how long the system must operate, and how effective the system's performance must be. An initial understanding of the operational concept may be obtained during the elicitation of the requirements, but is expected to continue as the development team strives to refine its understanding of the requirements to eventually choose between alternative solutions to implement.

Operational scenarios are classified as a requirements elicitation technique, but are in reality most often used together with the operational concept to refine and discover new requirements, needs, and constraints. An operational scenario is a sequence of events that might occur that includes the interaction of the product with its environment and users, as well as the interaction among its product components. The operational scenario should be consistent with the level of detail required by the stakeholders' needs, expectations, and constraints in which the proposed product is expected to operate.

Operational concepts and scenarios should not only focus on the system's functionality but should also include:

- Performance;
- Quality;
- Maintenance;
- Sustainment;
- Reuse;
- Training;
- Support;
- Disposal as appropriate.

The operational concept should also describe the environment in which the product is to operate, including boundaries and constraints. Operational concept and scenario development is an interactive process that continues into technical solution development.

Product and product requirements that come out of the refinement of the customer requirements should address the satisfaction of customer, business, and project objectives and any associated attributes such as effectiveness and affordability. The relationships among requirements should be established and maintained for consideration during change management and requirements allocation.

Architectural and Interface Requirements

The gathering and analysis of customer requirements and their translation to product and product component requirements include not only functional requirements but also architectural and interface requirements.

Architectural requirements that are commonly included are critical product quality factors such as maintainability and expandability. Maintainability refers to the characteristic of the system that makes it easy to repair if a problem is discovered in one of its product components or subsystems. Expandability refers to how easy it is to expand or add another function or product component to the existing integrated system. We have much to learn about architecture from building houses. Deciding to knock out a wall to expand the house when it is nearly 95% complete may not be possible if it was not architected to allow that expansion to happen. Similarly, if a system is built that is not highly cohesive and loosely coupled, trying to find and fix an error may result in many other parts of the system being negatively affected. Companies who have chosen to not consider these quality factors relating to architecture often find themselves in a position that they can no longer fix a problem or enhance the existing system without serious negative side effects. These same companies frequently elect to outsource their system to another company who has less ability than they do and soon they are totally out of business.

Interface requirements between product and product components should be identified in the product architecture and should be defined. The interfaces between functions must be defined and controlled as part of the product and product component integration. This includes both internal and external interfaces as is illustrated by Figure 11.3. As the interface designs are defined, the design becomes a requirement for products and product components that are affected by the interface. As the design progresses, the product architecture may be altered by the technical solution processes, creating new interfaces between product components and those components external to the product.

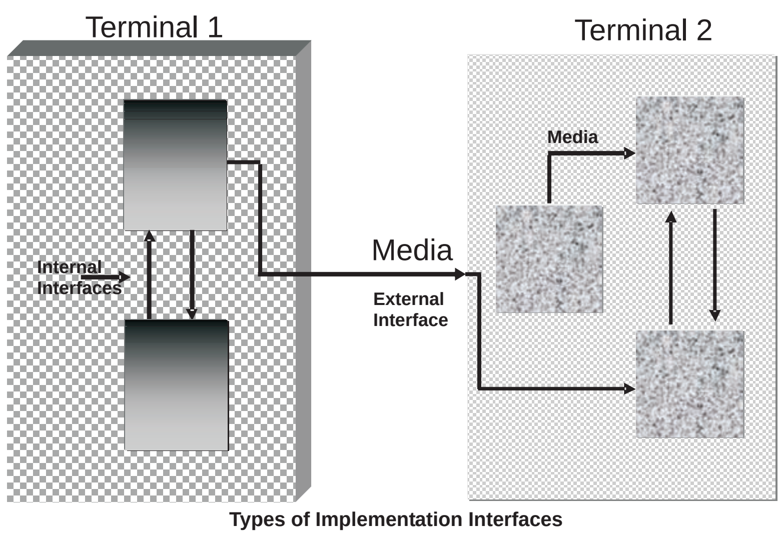


Figure 11.3 The importance of interfaces.

Allocation of Requirements

As indicated in Figure 11.4, the process of evolving the requirements from stakeholders' wants, needs, constraints, and interface requirements to product and product component requirements and to alternative solutions is also affected by how we allocate those requirements to the various engineering functions, services, processes, and people. When we say "allocate requirements to hardware," we are really saying that we intend to satisfy those requirements through a hardware implementation solution. Likewise, allocating requirements to software implies we intend to implement those requirements through software. We will see in the next chapter that the decision to allocate requirements to hardware, software, or other engineering or manufacturing functions may be altered as we strive to find the optimal alternative solution.

Validation of Requirements

There is one more important concept that is brought out in the requirements development process area and that is the concept of validation of requirements. Perhaps when you hear the phrase "validation of requirements," you think of conducting testing near the end of the project life cycle. The CMMI® clearly shows us that validation needs to start almost immediately after we have gone through the first round of gathering customer requirements and needs to continue until the "best" alternative solution has been chosen and implementation has begun. This is illustrated in Figure 11.5 as a spiral model taken from Kotonya and Sommerville's book on requirements engineering [1].

In the early rounds of requirements gathering and analysis, validation of requirements may mean meeting with the customer and feeding back the under-

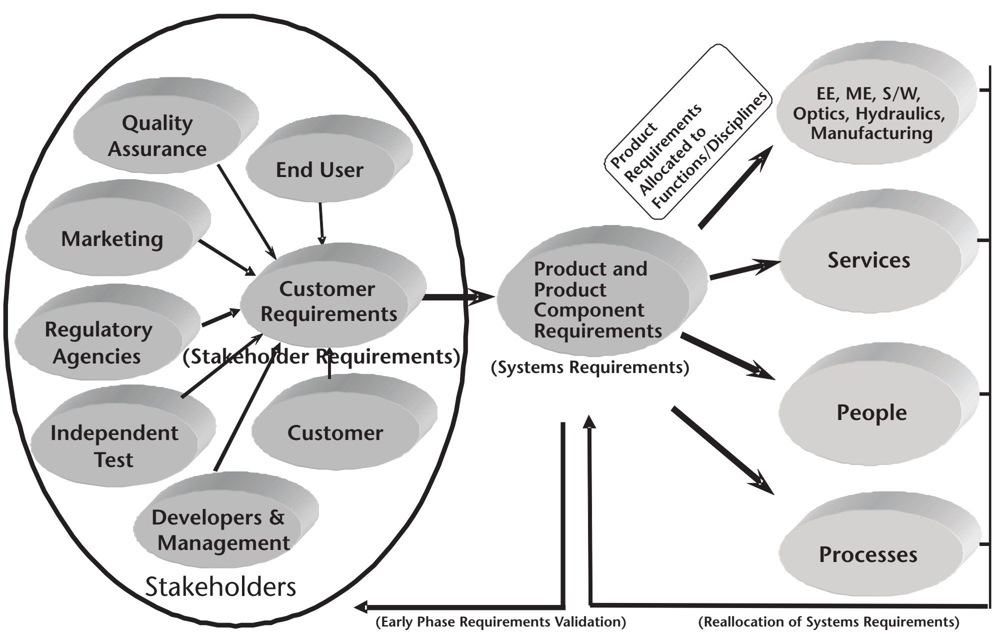


Figure 11.4 Product requirements allocated to functions.

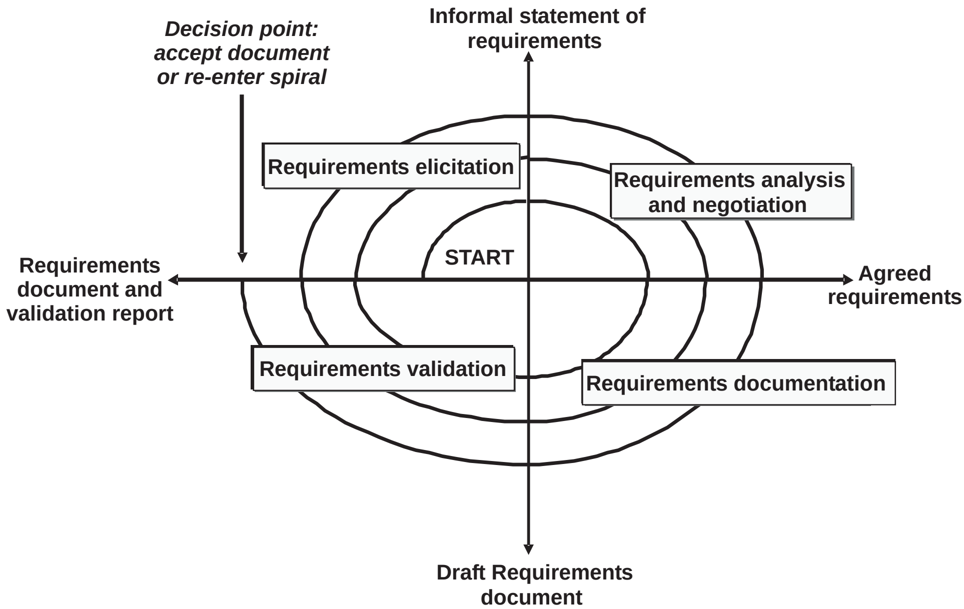


Figure 11.5 Spiral model of the product requirements engineering process. (*From:* [1]. © 1998 John Wiley & Sons, Inc. Reprinted with permission.)

standing of how the operational needs can be met. As the requirements are recursively refined and lower level requirements derived, techniques such as those used during the requirements elicitation stage can be used to *gain the customer's confidence* that the project is capable of using the customer requirements to develop the product that will indeed meet their operational needs.

The CMMI® reads as if it only applies to large projects that start from scratch. It is written that way so that it is clearly understood what is required in requirements development if a project did, indeed, start from scratch. However, on closer examination, the CMMI® covers all situations regarding the gathering or elicitation of requirements. Every project must "go around" the validation of requirements loop presented in Figure 11.5 at least once. The customer may provide a set of requirements to the project, or provide the design, or the requirements may exist as an output of a previous project's activities. Let us briefly examine the following types of projects in contrast to the "start from scratch" project:

- Product line or ongoing program;
- Build to specification (BTS);
- Build to print (BTP).

Product Line or Ongoing Program For product lines that have been ongoing for years, a new functional requirement may be required. The project should certainly conduct an impact analysis to the existing system, the development schedule, existing components, and even the integration and test environment. Investigation might even be conducted into the impact the new functionality might have on the operational environment or an existing and accepted operational scenario. At the very least, the project must share the impact analysis and validate the requirements with the "customer" and/or relevant stakeholders and make one validation loop.

Build to Specification A BTS system is one in which the customer provides the requirements specification. Project members may be tempted into thinking that they must satisfy the customer's requirements found in the provided requirements specification. At the very least, project members should review the customer's requirements specification against a list of requirements attributes and determine if a technical solution that will satisfy the customer's requirements in the operational environment is possible. At least one validation loop must be made.

Build to Print It may be even more tempting to think that a BTP type of project does not require any requirements validation. After all, the design is being provided, and all that is required is that the manufacturing department produce the product or product component from the design specification. But, even in this case, it is the responsibility of the project members to determine if their product environment sufficiently matches the assumptions made by the customer in the design specification. Single-source suppliers may be a reason to have extended discussions with the customer. Therefore, even in the BTP case, at least one validation loop must be made.

Requirements Management

While the requirements elicitation, analysis, documentation, and validation stages are evolving, it is important for the organization to maintain strong management control of those requirements. Requirements control must continue throughout the remainder of the project and possibly even the product life-cycle phases including when:

- The optimal alternative solution is chosen;
- The product is being implemented;
- Change requests to the requirements are received.

The Relationship Among RD, TS, and REQM

Figure 11.6 illustrates the Requirements Management (REQM) process area's ongoing and continuous relationship with the Requirements Development (RD) and Technical Solution (TS) process areas. As RD and TS are running around the track eliciting, analyzing, refining, and deriving more requirements, REQM is running around in parallel to ensure that their efforts are managed and controlled from the moment the first requirement is elicited. As RD captures its initial customer requirements, REQM reaches in periodically and captures those requirements, identifies them, and places them in a baseline with version control. When RD validates the requirements with the customer, REQM is again there to capture the improved understanding of the requirements and control them until the next improvement version is offered. This process continues even through the definition and selection of alternative solutions.

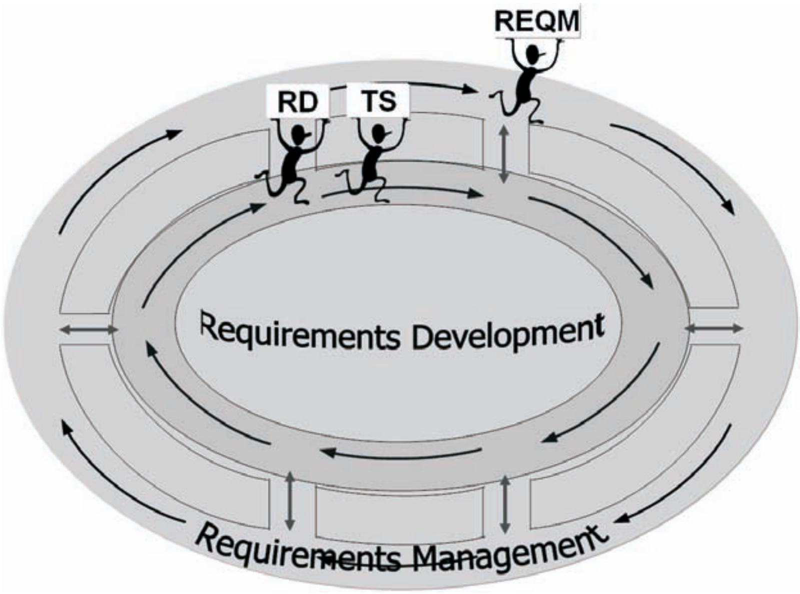


Figure 11.6 The requirements management and requirements development partnership.

For example, if a design decision is made to use a commercial off-the-shelf product, the interfaces to that COTS product become the requirements for the other product components that will interface with it. For this example, REQM would be there again to support the capture and control of the new requirements into the evolving baseline.

Configuration Management of Requirements Change Requests

Requirements management is essentially configuration management of requirements change requests. This was how it was originally stated in the CMM for software. However, Requirements Management became a process area on its own to indicate the great importance of not only gathering and analyzing requirements but to ensure that all relevant stakeholders inside and outside of the project's organization are kept aware of the current state of those requirements. Remember also that a requirements change request can come from anyone including the developers themselves, as well as the customer and end users. Requirements management deserves to be studied and implemented appropriately throughout all organizations.

One way to ensure that the requirements are reviewed by a multidisciplined team is to ensure that each discipline and support function that is necessary for the project's success has a shared understanding of the requirements. The review would take into consideration the viewpoints of, for example, the software, hardware, systems, quality assurance, independent test, and marketing functions.

Each requirement should be documented as a triple:

- *Requirement statement;*
- *Verification criteria* (How will the requirement be verified, through what mechanisms, and at what risk?);

- *Rationale* (Why is the requirement needed? Industry experience indicates that by insisting on the documenting of the rationale, up to half of the requirements can be eliminated).

Requirements Criteria for Valid Requirements

The objective of the review is to make sure the requirements meet certain criteria to ensure they are valid requirements [2]. These criteria include the following:

- *Necessary*. If the system can meet prioritized needs without the requirement, it is not necessary. A lack of relevance or necessity can be the result of reluctance to throw away information or a tendency to gather too much information.
- *Prioritized*. Requirements should show their relative priority based on a predefined scheme. Prioritization permits trade-offs to be made during analysis of possible solutions and negotiation of schedules, costs, and functionality.
- *Feasible*. The requirements are doable. Each item or logical set of items in the requirements specification can be implemented with the techniques, tools, resources, and personnel that are available within the specified cost and schedule constraints.
- *Concise*. The requirements are simply stated.
- *Complete*. All conditions under which the requirement applies are stated and it expresses a whole idea or statement.
- *Unambiguous*. The requirement can be interpreted in only one way.
- *Understandable*. The requirements must be understandable and usable by different sets of readers (i.e., customers, users, analysts, architects, designers, engineers, testers) with varying levels of technical knowledge.
- *Design independent*. It does not pose a specific implementation solution.
- *Not redundant*. It is not a duplicate requirement.
- *Consistent with each other*. The requirement is not in conflict with other requirements.
- *Testable*. Able to be verified and validated through reviews and testing.
- *Traceable*. The source of the requirement can be traced and it can be tracked throughout the system (e.g., to the design, code, test, and documentation).

Commitment Process for Changed Requirements

Commitment must be obtained from all relevant stakeholders on the current requirements. Changes to commitments must be renegotiated with all of the relevant stakeholders that were involved in the review and the baseline of the initial set of requirements. Changes to commitments made external to the organization should be reviewed by senior management, as one of the relevant stakeholders, to ensure that commitments can be accomplished along with the other previously approved commitments. In the past many people thought that this meant that the senior management team was supposed to conduct a technical review of the requirements. This is not the case. This is a business case review to ensure that the organization can indeed make this commitment or change in commitment along with all other commitments that have been made or are in the process of being made.

As the requirements evolve according to the practices described in the Requirements Development and Technical Solution PAs, the Requirements Management PA ensures that all project participants commit to the current, approved requirements and commit to the resulting changes in project plans, activities, and work products. Requirements Management also emphasizes that all changes to existing commitments should be negotiated before project participants commit to the requirements or requirements change.

Changes to the requirements must be controlled as they evolve over the project and product life cycle due to changing needs and derived requirements. All relevant stakeholders must review and agree on the change requests to the requirements before they are applied. Approved changes to the requirements are tracked and a change history is maintained for each requirement along with the rationale for the change.

Impact Analysis

To effectively analyze the impact of the changes, it is necessary for the source of each requirement to be known and the rationale for any change documented. To make sense of this, it is helpful to remember the different elicitation techniques that were used to obtain the requirements from the customer in the first place. An impact analysis procedure should take many things into consideration before the requirements change request is approved. Following is a list of items that should be examined at a minimum for the impact on them:

- Development schedule;
- Release schedule;
- Changes required to this system;
- Staffing;
- Components;
- Development and target equipment;
- Risks;
- Scope;
- Costs;
- Changes required to other systems or interfaces within the project;
- Other existing products or product lines.

Requirements Traceability and Keeping Life-Cycle Work Products Consistent

The CMMI® book [3] provides us with a new definition regarding requirements traceability. Requirements traceability is defined as a discernible association between requirements and related requirements, implementations, and verifications. The significance of this definition comes due to the change in the definition of bidirectional traceability. Bidirectional traceability used to be defined to be bidirectional from the requirements to systems test and to the project plans and back to the requirements. Project plans were removed from this definition. This left a gap on how to handle project plans whenever there was an approved requirements change request. The important benefit of requirements traceability is to ensure that all of the life-cycle work products, plans, and related activities are updated as necessary to keep them consistent with the approved changes made to the requirements. This guarantees that the integrity of the system is kept.

Bidirectional Traceability

The practices found in requirements management also point out another critical function that must be put in place and exercised for successful control of requirements; that function is bidirectional traceability. Bidirectional traceability was only mentioned at Maturity Level 3 in the CMM® for Software but is more properly stated as a Maturity Level 2 activity in the CMMI®. Bidirectional traceability, according to the CMMI® [3], is defined to be an association among two or more logical entities that is discernible in either direction (i.e., to and from an entity). Although this definition is found in the description of the Requirements Management process area, it may be more important to think of it as being part of Configuration Management. Bidirectional traceability is necessary if impact analysis is to be properly conducted whenever any life-cycle work product is being changed. For example, if we wish to make a change to a design document, bidirectional traceability is necessary to determine what life-cycle work products must be changed from the design document down to systems test and from the design document back up to the systems requirements specification.

Traceability information is needed to find other requirements that might be affected by proposed changes. Bidirectional traceability helps determine that all source requirements have been completely addressed and that all lower level requirements can be traced to a valid source.

Summary

During the iterative process of requirements analysis and definition, the following guidelines should be continuously applied:

- Analyze stakeholders' needs, expectations, constraints, and external interfaces to remove conflicts and to organize into related subjects.
- Determine what impact the intended operational environment will have on the ability to satisfy the stakeholders' needs, expectations, constraints, and interfaces.
- Analyze requirements to ensure that they are complete, feasible, realizable, and verifiable.
- Analyze derived requirements to determine whether they satisfy the objectives of higher level requirements.
- Identify key requirements that have a strong influence on cost, schedule, functionality, risk, or performance.

- Identify technical performance measures that will be tracked during the development effort.
- Analyze operational concepts and scenarios to refine the customer needs, constraints, and interfaces and discover new requirements.

Requirements management processes manage all requirements received or generated by the project including both technical and nontechnical requirements and those that are imposed on the project by the organization. When the requirements management, requirements development, and technical solutions processes are all implemented, their associated processes are normally tied closely together and performed concurrently.

References

[1] Kotonya, G., and I. Sommerville, *Requirements Engineering: Processes and Techniques,* New York: John Wiley & Sons, 1998.

[2] Young, R. R., *The Requirements Engineering Handbook*, Norwood, MA: Artech House, 2004.

[3] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI® Second Edition, Guidelines for Process Integration and Product Improvement,* Reading, MA: Addison-Wesley, 2007.

CHAPTER 12

Solution Definition

Solution definition is the process of transforming system-level and lower level requirements into a description of product components and interfaces in order to implement and integrate the product components into subsystems and systems while satisfying the requirements. It includes the identification of key design constraints and the identification of alternative solutions and the selection of a life-cycle balanced solution. This chapter presents the concepts and guidelines used to establish selection criteria and then select product or product component solutions from alternative solutions. It includes the concepts of decision analysis and resolution for more formal decision making. This chapter also focuses on the design (including top-level design or product architecture) and implementation of the product or product component and on the verification of the product components through peer reviews and unit testing.

Selecting the "Best" Alternative Solution

Alternative solutions need to be identified and analyzed to enable the selection of a balanced solution across the life of the product in terms of cost, schedule, performance, and quality. This concept was first brought out in Chapter 6 to meet the project and product constraints that needed to be optimized.

Figure 12.1 shows the concurrency of events taking place that eventually leads toward the selection of the "best" alternative solution. These concurrent activities are described in the Requirements Development (RD), Technical Solution (TS), and Product Integration (PI) process areas. These concurrent events elicit the requirements; analyze and validate them; develop the operational concepts and operational scenarios; allocate the requirements to the appropriate engineering disciplines; make the build, buy, or reuse decision; and decide on the appropriate architecture.

To illustrate why identifying and selecting the "best" alternative solution is such an important consideration, let us build a scenario.

The Problem

Provide a product or product component solution across the life of the product that optimizes the constraints of:

- Cost:
    - a…b;

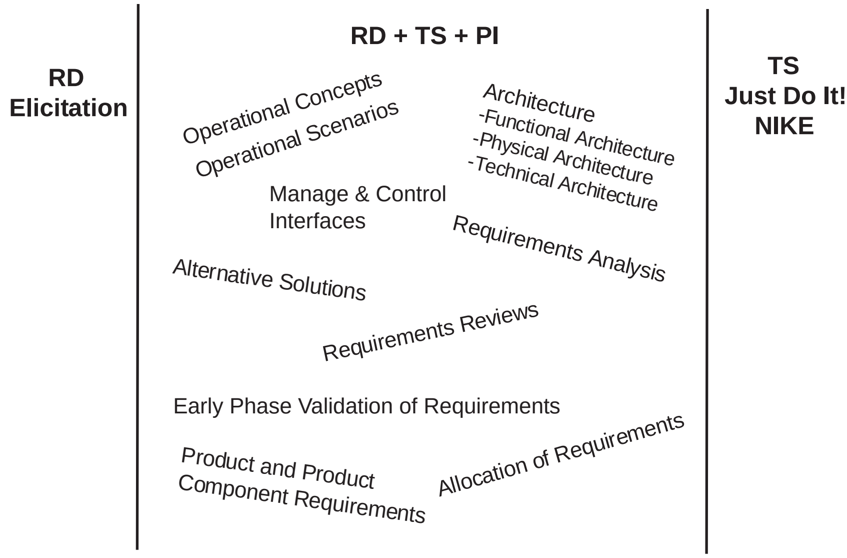


Figure 12.1 Concurrency of RD, TS, and PI.

- Schedule:
    - c…d;
- Performance;
- Quality;
- Functionality;
- Risk;
- Customer satisfaction.

The qualifiers "a…b" and "c….d" represent ranges of acceptable cost or schedule constraints respectively.

The Solution

This may be accomplished through the allocation of the requirements to:

- Software;
- Hardware;
- Electronics;
- Mechanics;
- Hydraulics;
- Manufacturing processes;
- Services;
- People.

The solution may be accomplished through:

- In-house development;

- Purchase of commercial-off-the-shelf products;
- Use of suppliers;
- Use of reuse components.

It will be influenced by the proposed product architecture that addresses the critical quality factors of the product such as maintainability and expandability.

Satisfying the Selection Criteria

Given this scenario, for reasonably large and/or complex systems, the combinations and permutations possible will require us to identify alternative solutions. Then, based on selected criteria, we select the "best" solution to solve our originally stated problem. "Best" implies that the alternative solutions must fit within the range of given values for cost, schedule, performance, quality, functionality, risk, and customer satisfaction. It is possible that a project will find that it cannot fit within those given ranges for one of more of the constraints. This may necessitate a change to the product architecture. It may cause a reallocation of the requirements to the engineering functions or processes or services. It may cause a renegotiation with the stakeholders to determine if any of the constraints can be eased. In any event, "best" is not meant to be "the guaranteed best solution," but the alternative solution that "best optimizes" the given constraints or selection criteria.

Allocation of Requirements as a Solution Set

Solutions are typically defined as a "set." Alternative solutions are not only different ways of addressing the same requirements, but they also reflect a different allocation of requirements among the product components as a solution set—not as single product components. The objective is to optimize the set as a whole. Detailed alternative solutions and selection criteria consistent with business objectives typically include:

- Cost (development, procurement, support);
- Technical performance;
- Complexity of the product component;
- Product expansion and growth;
- Technology limitations;
- Sensitivity to construction methods and materials;
- Risk;
- Disposal;
- Capabilities and limitations of end users and operators.

Choosing an alternative solution under the conditions described above normally requires a formal evaluation process that focuses on identified alternatives against established criteria. The CMMI® Decision Analysis and Resolution process area may be called on to assist in the selection of the "best" alternative solution that will satisfy the wants and needs of the project. Decision analysis and resolution involves making good business decisions by:

- Selecting a decision-making technique and level of structure;
- Identifying criteria that will be the basis of the decision:
    - This criteria should address design issues for the life of the product, such as provisions for easy insertion of new technologies or the ability to better utilize available commercial products.
    - The criteria needed for decision-making techniques range from consensus-based decisions to the use of probabilistic models and decision theory.
- Identifying alternatives;
- Evaluating the alternatives against the criteria.

The final selection of an alternative should be accompanied by:

- The selected technique, criteria, and alternatives;
- The rationale for the selection of the final solution;
- The rationale for not selecting one of the other alternative solutions.

Although the use of formal decision analysis techniques should always be considered, some guidelines for requiring formal decision making include:

- When a decision directly relates to topics assessed as being of medium or high risk;
- When a decision relates to changing critical requirements;
- When a decision relates to changing the architecture;
- When a decision causes schedule delays over a certain percent or specific amount of time;
- When a decision causes rethinking of the operational concepts;
- When a decision affects the operational scenarios that have been developed;
- When a decision's costs are reasonable compared to the decision's impact.

Selection of the best solution set establishes the requirements, provisionally allocated to that solution, as the set of allocated requirements that will be designed and implemented.

Commercial Off-the-Shelf Products

If the decision is to purchase an off-the-shelf product, the requirements developed so far should be used to establish the COTS supplier agreement. Factors that can affect the make-or-buy decision include:

- Functions the products or services will provide;
- Costs of acquiring versus developing internally;
- Critical delivery and integration dates;

- Functionality and quality of available products;
- Skills and capabilities of potential suppliers;
- Stability of potential suppliers, that is, length of time in business, reliability of existing products, ongoing technical support services;
- Licenses, warrantees, responsibilities, and limitations associated with products being acquired;
- Product availability.

Designing and Implementing the Product or Product Component

Product or product component designs must provide the appropriate life-cycle content for implementation, modification, maintenance, sustainment, installation, support, and training. Product design typically consists of at least two design phases that may overlap during actual execution: preliminary design and detailed design. Preliminary or top-level design establishes product capabilities and the product or systems architecture. Detailed design fully defines the structure and capabilities of the product components.

Preliminary or Top-Level Design

Defining the architecture requires the following items to be described:

- Processes exist that need to take place in order for the system to accomplish its intended functions.
- The individual processes transform either data or materials that flow between them.
- The processes or activities or operations follow rules that establish the conditions under which they occur.
- The components that will implement the design (hardware, software, personnel, and facilities) must be described.

Systems or product architecting has been defined as the process of creating complex, unprecedented systems. Building systems in today's fast-moving world is difficult at best. Requirements of the marketplace are ill defined, and rapidly evolving technology provides new services at a global level instantly. Uncertainty is increasing about the way the system will be used, the components that will be incorporated, and the interconnections that will be made.

Traditional Approach to Systems Architecting

Many methodologies have been developed to support a traditional systems development model. The steps normally consist of defining the requirements, considering several options, and emerging with a well-defined design through a process of elimination. This is illustrated in Figure 12.2.

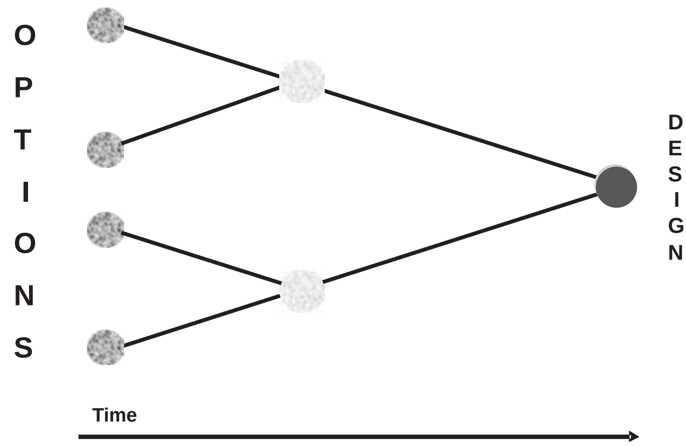


Figure 12.2 The traditional approach. (*From:* [1]. © 1999 John Wiley & Sons, Inc. Reprinted with permission.)

The traditional approach to architecting is effective when the requirements are well defined and remain essentially constant during the system development period. If the implementation of the system is long, on the order of years, the requirements may change due to changing needs and new technology that offers different alternatives and opportunities that the traditional approach to systems architecture cannot handle.

Evolutionary Approach to Systems Architecting

A new systems architecture approach has been emerging that has its roots in software systems. It is called the evolutionary approach and is oriented to deal with the uncertainty in requirements and technology, especially for systems with long development times and expected long life cycles. This approach is illustrated in Figure 12.3. Requirements are allowed to be more abstract and, therefore, subject to interpretation. Alternative solutions are explored and pursued further as new technology options become available. Intermediate designs are saved and some of them are implemented as prototypes but not operationally implemented. Others are implemented in traditional ways. At any time in the development process, when there is a need to build a system, the available solution that best meets the current requirements and product constraints is selected and implemented using one of the systems engineering approaches.

Architectures may include standards and design rules that govern development of product components and their interfaces, as well as guidance to assist product developers. In the context of the architectural requirements, multiple architectures supporting alternative solutions may be developed and analyzed to determine the advantages and disadvantages. Operational concepts and scenarios are used to generate the use cases and quality scenarios that become the guidelines for refining the architecture and evaluating the suitability of the architecture.

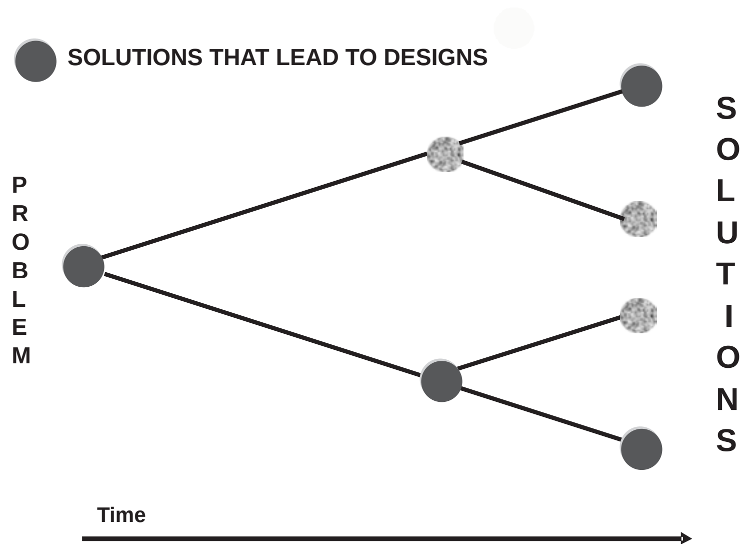


Figure 12.3 Evolutionary approach. (*From:* [1]. © 1999 John Wiley & Sons, Inc. Reprinted with permission.)

Detailed Design

During detailed design, the product architecture details are finalized, product components are completely defined, and interfaces are fully characterized. Design criteria are often established to ensure that the product or product component exhibits one or more of the following quality attributes:

- Modularity;
- Clarity;
- Maintainability;
- Expandability;
- Portability;
- Efficiency;
- Reliability;
- Security;
- Usability;
- Scalability.

These quality factors can be defined following the goal–question–paradigm shown in Figure 12.4. The criteria that define the quality factors and the metrics that help measure the extent to which the product or product component exhibits that characteristic or quality factor must be used for design purposes. It can be used to verify that the quality factors are being built into the product or product component. It can be further used to validate that the product operates in the operational environment and will indeed exhibit the required quality characteristic.

There is significant debate about effective design methods. Whether or not a design method is effective depends on how much assistance it provides the designer and the cost effectiveness of that assistance. Criteria must be established and main-

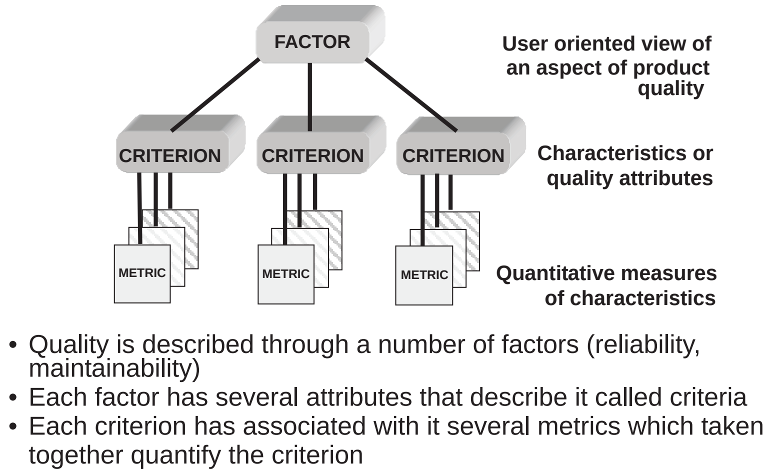


Figure 12.4 Product quality metrics.

tained against which the effectiveness of the design methods can be determined. Highly sophisticated methods are not necessarily effective in the hands of designers who have not been trained in the use of the methods. Examples of techniques and methods that facilitate effective design include:

- Prototypes;
- Structural models;
- Object-oriented design;
- Entity relationship models;
- Design reuse;
- Design patterns.

Technical Data Package

The design should be recorded in a technical data package that is created during the development of the preliminary design to document the architecture definition. The package should be maintained throughout the life of the product to record essential details of the product design. The technical data package is similar to the "unit record folder" or the "software development file" descriptions from the 1970s and 1980s. A technical data package provides the developer with a comprehensive evolutionary description of the product or product component as it is developed.

The technical data package typically includes:

- Work breakdown structure;
- Product architecture description;
- Derived requirements;
- Allocated requirements;
- Acceptance criteria;
- Product component descriptions;

- Product-related life-cycle process descriptions;
- Key product characteristics;
- Required physical characteristics and constraints;
- Design constraints;
- Interface requirements;
- Materials requirements;
- Fabrication and manufacturing requirements;
- Drawing package;
- Verification criteria used to ensure that the requirements have been achieved;
- Conditions of use (environments) and operating/usage scenarios;
- Rationale for decisions (design choice).

Interface Descriptions

As mentioned in Chapter 11, interfaces are one of the most important configuration items to be identified and controlled throughout a project life cycle. Projects must develop detailed interface descriptions during the refinement of the product and product requirements and during the selection of alternative solutions. Interfaces must be defined with other product components and external items. When that external item is a COTS product, the interface description for the COTS product becomes a requirement to which all of the other product components that will interface with that COTS product must adhere.

The criteria for interfaces frequently reflect a comprehensive list of critical parameters that must be defined—or at least investigated—to determine their applicability. Interface designs include the following:

- Origination;
- Destination;
- Stimulus and data characteristics for software;
- Electrical, mechanical, and functional characteristics for hardware.

Internal and external interfaces were discussed in Chapter 11 and are further explained in Chapter 14.

Implementation

The product design process to be implemented is described in the Technical Solution process area as follows:

- Software is coded.
- Data is documented.
- Services are documented.
- Electrical and mechanical parts are fabricated.
- Facilities are constructed.

- Materials are produced.
- Unique manufacturing processes are put into operation.

Peer Reviews and Unit Testing

The requirement to conduct peer reviews and unit testing is also in the description of the Technical Solution process area. Detailed descriptions of peer reviews and unit testing are found in Chapter 13. Peer reviews are further discussed in Chapter 14, which covers product integration, verification, and validation.

Examples of peer reviews include structured walkthroughs and inspections. Examples of unit testing methods that give an indication of product component quality include:

- Statement coverage testing;
- Decision coverage testing;
- Condition coverage testing;
- Multiple condition coverage testing.

See Chapter 13 for a full description of these test coverage techniques.

Product Support Documentation

Although the focus within the description of technical solution is on architecture, design, and implementation, it is important to note that this is not complete until the documentation that will be used to install, operate, and maintain the product is also developed, reviewed, and revised as necessary. It is recommended that the project develop preliminary versions of the installation and that operator's and user's manuals be developed in the early phases of the project life cycle and reviewed by all relevant stakeholders inside and outside of the organization. This documentation will be part of the delivery package that will be described in Chapter 14.

Summary

Choosing a technical solution that will implement the product or product requirements in the operational environment requires alternative solutions to be identified and analyzed. Alternatives and their cost–benefit analysis should be considered in advance of selecting a solution. For companies that have indicated they "only develop software," developing alternative solutions based on the combinations and permutations of building, buying, and using COTS products and/or making use of reuse components is still important.

Architectural features must be considered that will provide the foundation for the implementation and evolution of the product and product components. Reuse, COTS, and nondevelopmental items such as customer-furnished equipment must be considered relative to cost, performance, and product maintenance and support. Design constraints for each engineering discipline must be taken into consideration and compared for each allocated set of systems requirements. Product components must be verified using techniques such as peer reviews and unit testing before they are brought together for product integration and testing.

The selection of the best technical solution enables the selection of a life cycle that is a balanced solution in terms of cost, schedule, technical performance, quality, functionality, risk, and customer satisfaction. Identifying and selecting the criteria to guide the selection of the best alternative solution may require a formal decision analysis and resolution procedure. The rationale for the chosen alternative solution and the rationale for alternative solutions that were not chosen must be documented for future reference as the system evolves.

Reference

[1] Sage, A. P., and W. B. Rouse, *Handbook of Systems Engineering and Management*, New York: John Wiley & Sons, 1999.

CHAPTER 13

Reviews and Testing

A review is an evaluation of a life-cycle work product(s) or project status to determine if there are any deviations from planned results and to recommend improvement. Peer reviews may be thought of as *human-based* testing as opposed to *computer-based* testing. An anomaly is any condition that deviates from expectations based on requirements specifications, design documents, standards, plans, and so on, or from someone's experiences. These anomalies are most often called *defects*.

Why Should We Conduct Reviews?

Reviews are conducted for these reasons:

- Detect defects.
- Remove defects as close to the point of insertion as possible.
- Determine product progress/status.
- Identify potential improvements.
- Produce technical work of a more uniform and predictable quality.
- Gain ownership by the project team.
- Assist employees with cross-training.
- Reduce costs to build and maintain better products.
- Reduce development time.
- Reduce testing cost and time.
- Reduce total system maintenance cost dramatically (as much as 10 to 1 according to recent statistics).

The sooner a defect is found, the cheaper it is to fix. Studies over the years have indicated that if a defect can be detected and fixed in the requirements phase, it might cost \$1. If that defect moves through subsequent life-cycle phases until systems test, the cost to find and fix the defect might be as much as \$500. If the defect continues further and reaches the customer before it is detected, the cost to fix may be nearer \$1,000 or higher. The reasoning is simple if you look at the rework costs. If the defect makes it to the customer and comes back to the supplier organization to be resolved, the cost of rework includes the time required to analyze the problem, the time to fix the problem, the time to conduct a review, the time to conduct unit testing, the time to conduct regression testing, and the time to get the fixed system back to the customer. This does not include the time supplied by the support groups such as quality assurance and configuration management that is necessary to support the defect finding and fixing process.

The ultimate objective of a peer review is to minimize the number of defects being passed along from one life-cycle work product to another and from one life-cycle phase to another by finding and fixing the defects at the point in the life cycle in which they were created. When one looks to peer reviews as a technique to support quality management [1], "…the inevitability of defect is not the true source of concern for quality management, but rather at what point in the product life cycle these defects are detected and corrected. If we recognize that development is an error-prone process, and gear our quality program to finding and correcting defects early in the process, we reduce the cost of defect correction and increase productivity—while we improve quality."

Reviews as a Management Control Tool

During any project, management requires a means of assessing and measuring progress. Questions such as these might be asked:

- Are we ahead or behind where we expected to be?
- Will we complete the work as planned?
- Do we require more computer or human resources to meet the planned schedule?
- Is the required functionality being implemented?
- What risks are we taking based on the project information that we have today?

We are accustomed to writing status reports and making project progress presentations. But how do successive levels of management and ultimately the customer know just what progress has been made? "True" progress cannot be measured by counting the completion of tasks unless there is a reliable way of measuring the quality of the work performed and knowing that it would not have to be redone or changed later (rework!). In other words, unless techniques such as reviews and testing are being applied to the life-cycle work products and the results analyzed, management progress reports could represent little more than wishful thinking. Figure 13.1 shows a representative sample of the different types of reviews and audits that can be applied for achieving quality purposes to support project progress reporting.

Management or Progress Reviews

A management review is a formal evaluation of a project-level plan or project *progress* relative to that plan by a designated review team. Management reviews are normally conducted for these reasons:

- Track progress according to plan, based on an evaluation of the product development status.

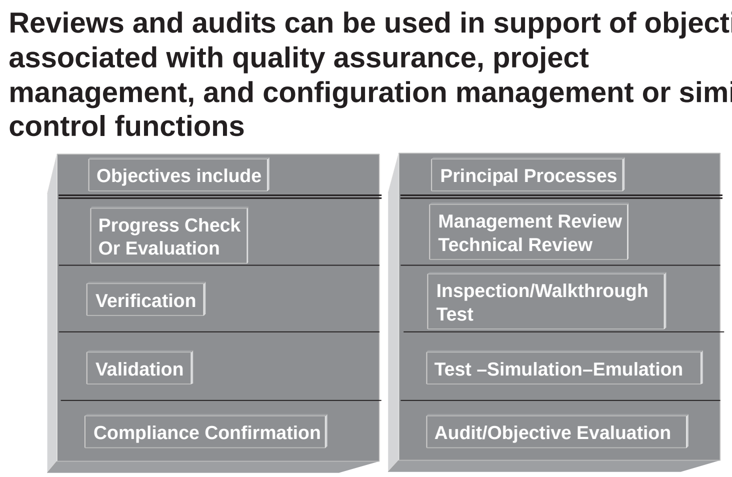


Figure 13.1 Principal processes for achieving quality objectives.

- Identify inconsistencies with and deviations from plans.
- Confirm requirements and their system allocation.
- Monitor risks.
- Evaluate the effectiveness of management approaches used to achieve fitness for purpose.
- Take corrective action and change project direction or identify the need for alternative planning.
- Maintain control of the project through adequate allocation and reallocation of resources.
- Change the scope of the project.

Three basic types of management reviews or progress reviews can be found in the CMMI®: project manager reviews, milestone reviews, and senior management oversight reviews. Let us take a brief look these types of management reviews.

Project Manager Review

The project manager review normally translates into the weekly project meeting that is called by the project manager. At a minimum, the project manager and development staff are in attendance. Representatives from systems engineering, QA, CM, and testing should also be in attendance. Items discussed at the weekly project meeting include tracking of the standard issues of technical activities, cost, and schedule performance against the plan. Staffing concerns may also be brought up. Resources planned for and consumed are reviewed. Risks are identified, prioritized, and contingency plans discussed. Necessary commitment changes are acknowledged, documented, and approved. Stakeholder involvement is discussed to ensure that the relevant stakeholders are involved at the time it was deemed necessary and the project is receiving the necessary results from their involvement. Corrective actions may be necessary and all planning documents are updated as necessary.

Milestone Reviews

Formal milestone dates are documented in the project management plan, tracked, and reviewed. Milestones normally represent meaningful points in the project's schedule. This might signify the end of a phase or a major activity. It might be linked to customer involvement and even contract payments such as in the defense world of conducting preliminary design reviews (PDRs) and critical design reviews (CDRs).

Typical milestone reviews include examination of technical progress, planned engineering activities, commitments, and plans. They normally address project and other engineering discipline risks. Action items from previous meetings are brought up to determine progress and closure and new action items are recorded, assigned, and reviewed as necessary. All decisions are documented with corresponding actions and resolutions.

Senior Management Oversight Reviews

Watts Humphrey's original intent for senior management oversight reviews was to provide higher level management with the appropriate visibility into the processes being used on the projects.

When discussions around senior management oversight reviews come up, many individuals and organizations believe that this strictly refers to a meeting with the senior management team to discuss the project progress including technical, cost, staffing, schedule, performance, risks, and other necessary functions. Clearly these senior management team progress reviews are necessary. But Mr. Humphrey wanted the senior management team to receive more than the normal project status. He wanted the senior managers to understand what processes were being followed on the projects, whether they efficient and effective, and whether they helping the project team members produce the required product and service quality.

It is the senior management team that owns the organization's process, and the senior management team is the only group that can reallocate resources to support increased process improvement and quality needs. Only the senior management team can authorize mass training and award the necessary authority to carry out these process improvement and quality management activities.

Management oversight did not and does not indicate that the senior management team missed something important. It was an indication that they could see over the projects, understand the processes and work environment provided to them, and decide on the priorities to support the improvement of the processes in order to assist the projects in accomplishing their business goals.

Peer Reviews

Although the SEI's Capability Maturity Model® defines peer reviews and refers to inspections and structured walkthroughs as examples of peer reviews at CMMI® ML 3, a number of peer reviews types can be used for different objectives and expected results including:

- Buddy checks;

- Circulation reviews;
- Technical reviews;
- Inspections;
- Walkthroughs;
- Structured walkthroughs.

These peer review types will be described in an overview fashion in this chapter but first it might be helpful to provide a historical look at how use of the term *peer reviews* came about in the CMM® and now the CMMI®.

Walkthroughs and structured walkthroughs had been introduced to the industry by leading figures such as Gerald Weinberg[1] and Ed Yourdon[2] in the late 1960s and early 1970s. In 1972, Michael Fagan[3] of IBM developed a more formal approach to defect detection for early-phase life-cycle work products called inspections. Inspections were a more formal approach to conducting reviews with an absolute focus on detecting major defects and eliminating them if possible.

As Mr. Watts Humphrey worked for IBM for more than 30 years, he naturally brought with him that notion of conducting a review on selected life-cycle work

1. For more than 45 years, Jerry Weinberg has worked on transforming software organizations, particularly emphasizing the interaction of technical and human issues. After spending between 1956 and 1969 as software developer, researcher, teacher, and designer of software curricula at IBM, he formed the consulting firm of Weinberg & Weinberg to help software engineering organizations manage the change process in a more fully human way. In June 1997, he was inducted into the Computer Hall of Fame, along with such notables as Charles Babbage, Semour Cray, James Martin, Grace Hopper, Gerald Weinberg, and Bill Gates. Jerry is author or coauthor of several hundred articles and more than 30 books. His earliest published work was on operating systems and programming languages, but the 1971 publication of *The Psychology of Computer Programming* is considered by many the beginning of the study of software engineering as human behavior. His subsequent works have been an elaboration of many of the software engineering topics raised in that book, through all phases of the software life cycle, including defining problems and requirements, analysis and design, testing and measurement, as well as management. He cowrote the *Handbook of Walkthroughs, Inspections, and Technical Reviews* with Daniel P. Freedman in 1982. (*From:* http://www.geraldmweinberg.com/BIOStuff/EachBIO/bio.Jerry.html.)
2. Ed Yourdon is an internationally recognized computer consultant who specializes in project management, software engineering methodologies, and Web 2.0 development. According to the December 1999 issue of *Crosstalk: The Journal of Defense Software Engineering,* Ed Yourdon is one of the 10 most influential men and women in the software field. In June 1997, he was inducted into the Computer Hall of Fame, along with such notables as Charles Babbage, Semour Cray, James Martin, Grace Hopper, Gerald Weinberg, and Bill Gates. Yourdon is widely known as the lead developer of the structured analysis/design methods of the 1970s. He was a codeveloper of the Yourdon/Whitehead method of object-oriented (OO) analysis/design and the popular Coad/Yourdon OO methodology. He wrote his book on *Structured Walkthroughs,* 4th ed., in 1989. Mr. Yourdon began his career in the computer industry at Digital Equipment Company more than 35 years ago. Mr. Yourdon is the author/coauthor of more than 25 computer books, as well as more than 525 technical articles since 1967. (*From:* http://www.cutter.com/meet-our-experts/ eybio.html.)
3. Michael Fagan is the founder and CEO of Michael Fagan Associates and the creator of Fagan Inspections and the Fagan Defect-Free Process™. As a product development manager at IBM, Michael created the Fagan Inspection Process for use on his own projects. Over the years, this has been enhanced and expanded into the Fagan Defect-Free Process™, incorporating formal process definition, and reinforcing the continuous process improvement aspect of the inspection process. The methodology developed by Michael Fagan is credited with dramatically reducing the number of defects in software and hardware products, increasing the feature content per release, shortening cycle time, increasing customer satisfaction, improving development processes, accelerating SEI/CMM® maturity in organizations, and significantly reducing costs!

   Michael Fagan has 20 years of experience in IBM as a line manager of software development, engineering development, and manufacturing. In addition, he was manager of programming methodology for IBM's DP Product Group (Worldwide); the first software senior technical staff member in IBM's T. J. Watson Research Laboratory; a member of the Corporate Technology Staff; and one of the founding members of the IBM Quality Institute. (*From:* http://www.mfagan.com/bio_frame.htm.)

products, especially in the early phases of the life cycle. As fate would have it, I wrote 50% of IEEE Standard 1028, *Reviews and Audits,* which was approved before I joined the SEI in 1988. I wrote the section on walkthroughs and Bob Ebenau wrote the section on inspections. As the CMM® started evolving, Mr. Humphrey insisted that inspections be required at Maturity Level 3. But in the intervening time before the CMM® was made public, many debates were held over what these required reviews would be called. Not all people appreciated the IBM way as Watts did, so in the end, the term *peer reviews* was used. However, even in the initial days of the CMM®, the intent was to apply the formal approach to life-cycle work products in the manner defined by Michael Fagan.

Let us now examine each of the peer reviews types listed earlier in an overview fashion.

Buddy Check

A buddy check is normally thought of as an informal verification technique in which the life-cycle work product is examined by the author and one other person. The buddy check operates on basically the same principles as a walkthrough. The "buddy" may or may not prereview the life-cycle work product prior to the peer review. The author walks the "buddy" through the life-cycle work product and describes what is intended in each section. The "buddy" offers comments for improvement on technical correctness, style, order of presentation, clarity, and understandability. Most importantly the buddy decides on the "fit" to the described intent by the author. The author may or may not accept the buddy's suggestions for improvement.

The objectives of buddy checks include:

- Improve the life-cycle work product;
- Consider alternative implementations;
- Exchange techniques and style variations;
- Point out problems with clarity and understandability;
- Allow the author to look at the life-cycle work product from a different "angle" or point of view;
- Mentoring of others in the concepts embedded in the life-cycle work product.

The second-to-last objective is often the greatest benefit of using the buddy check. It is not uncommon for the author to discover defects while describing the intent of the work product to the buddy.

There is another reason—a very powerful and very human reason—to incorporate buddy checks into a person's, project's, or organization's set of peer review definitions. When we ask our colleagues to give some of their very valuable time to review a work product of ours to help make it better, it is simply common courtesy for the author to make his/her life-cycle product as good as possible before giving it to a colleague or peer. No one wants to be asked to review a life-cycle work product that is incomplete or laced with small but annoying defects such as spelling errors or incorrect formatting, which distracts the reviewer from the real task of reviewing to detect defects.

Circulation Review

Circulation reviews take on attributes of both buddy checks and walkthroughs. Circulation reviews can be informal or follow strict rules. The life-cycle work product is circulated to each reviewer who reviews it and either attaches comments, questions, and recommendations directly on the life-cycle work product or places them into a separate document. Circulation reviews are dependent on the goodwill of the reviewer. Normally, the author has the authority to accept the reviewers' comments and suggestions or ignore them.

Circulation reviews are especially useful in these situations:

- Reviewers are geographically separated and face-to-face or teleconferencing is not possible.
- Individuals who are asked to be reviewers are willing to "work in" the review but not take the time out of their schedule to participate in a more formal review.
- The author is looking for a large cross section of opinions regardless of background or experience.
- A large population of stakeholders must be satisfied.
- The time during which the review must be completed is not a constraint, but the author needs as detailed a look at the life-cycle work product as possible.

The objectives of circulation reviews include:

- Improve the life-cycle work product;
- Consider alternative implementations;
- Point out problems with clarity and understandability;
- Point out areas of concern and offer comments and suggestions;
- Gain consensus from a large population of reviewers;
- Gain input from valuable contributors who cannot be present for a face-to-face review.

Technical Review

A technical review is a formal team evaluation of a life-cycle work product to:

- Identify any discrepancies from specifications and standards;
- Determine its suitability for use;
- Provide recommendations after the examination of various alternatives.

Technical reviews may be held at the request of functional management, project management, quality management, systems engineering, software engineering, or hardware engineering. Technical reviews may be required to evaluate the impacts of hardware anomalies or deficiencies of the software.

The objectives of technical reviews are to ensure that:

- The life-cycle work product conforms to its specifications;

- The development or maintenance of the life-cycle work product is being done according to plans, standards, and guidelines applicable to the project;
- Changes to the life-cycle work product are properly implemented and affect only those areas of the system identified by the change specification.

Technical reviews are typically used for the classic design reviews such as PDRs, CDRs, and test readiness reviews (TRRs). Technical reviews are also appropriate for life-cycle work products such as architectural specifications.

Next we describe inspections, then walkthroughs, and finish up with a discussion of structured walkthroughs and a comparison to inspections.

Inspections

An inspection is a formal verification technique in which life-cycle work products are examined in detail by a group of peers for the explicit purpose of detecting and identifying defects. The process is led by a moderator or facilitator or inspection leader who is not the author and is impartial to the life-cycle work product under review. The author is not allowed to act as the moderator. Written action on all major defects is mandatory. Rework due to corrections of major defects is formally verified. Defect data is systematically collected and stored in an inspection database. This defect data is analyzed to improve the product, the process, and the effectiveness of the inspection process. Individuals holding management positions over any member of the inspection team are normally not permitted to participate in the inspection.

The objective of an inspection is to detect and identify life-cycle work product defects in a rigorous, formal, peer examination that does the following:

- Verifies that the life-cycle work product satisfies both its specification and preceding intermediate work products.
- Verifies that the life-cycle work product conforms to applicable standards.
- Identifies real or potential deviations from standards and specifications.
- Collects engineering data (i.e., defect and effort data).
- Does not examine alternatives or stylistic issues.

Inspections are most effective if they are an integral part of the development process.

Inspector's Responsibilities

Inspectors/reviewers should be chosen to:

- Identify and describe the major and minor defects found in the product or product component (checking);
- Be chosen to represent different points of view (requirements, design, development, test, independent test, project management, quality management, and so on);

- Be assigned specific review topics to ensure effective coverage:
  - Conformance with a specific standard;
  - Overall coherence.

Inspection Minimum Entry Criteria

An inspection shall not be conducted until all of the following events have occurred:

- The product or product component is complete and conforms to project standards for content and format.
- Any automated error-detecting tools required for the inspection are available.
- Required supporting documentation is available.
- For a reinspection, all items noted on the defect list are resolved.
- The author is available.
- A trained moderator and sufficient number of skilled inspectors are available.

Inspection Preparedness

The inspection leader or moderator is required verify that the inspectors are prepared for the inspection. The inspection leader may reschedule the inspection (logging) if the inspectors are not adequately prepared.

Inspection Defect List

The recorder is responsible for entering each major and minor defect, location, description, and classification on the defect list. The author should be ready to answer specific questions and contribute personally to the defect detection process. If there is disagreement about a defect, the potential defect shall be logged and marked for resolution at the end of the meeting. The moderator is responsible for having the defect list reviewed with the team to ensure it completeness and accuracy.

Walkthroughs

Walkthroughs were designed to be a less formal verification technique in which life-cycle work products are examined by a group of peers for the purpose of finding defects, omissions, and contradictions. The walkthrough is normally led by the author or the producer of the material being reviewed. As the walkthrough progresses, errors, suggested changes, and improvement suggestions are noted and documented. The consolidated notes are taken by the author for review and revision as the author sees fit.

The objectives of walkthroughs in addition to detecting defects are to:

- Improve the life-cycle work product;
- Consider alternative implementations;

- Point out efficiency and readability problems, or modularity problems if the life-cycle work product is code;
- Exchange techniques and style variations;
- Educate the participants.

The walkthrough leader/author normally distributes the product or product component and conducts the walkthrough meeting. Each team member prepares by examining the life-cycle product in advance and prepares a list of items for discussion during the meeting itself. The author presents an overview of the life-cycle product under review, facilitates the general discussion, and serially presents the life-cycle work product in detail, accepting team members' comments as the author proceeds.

Walkthroughs should provide data for the analysis of:

- The quality of the product;
- The effectiveness of the acquisition, supply, development, operation, and maintenance processes;
- Efficiency of the walkthrough itself.

Structured Walkthroughs

Structured walkthroughs are more closely aligned with inspections than the informal walkthrough. One can think of doing a walkthrough while using many of the inspection requirements such as:

- Ensuring that each participant has individually prepared for the structured walkthrough and is prepared with a listing of major and minor defects before coming to the face-to-face part of the structured walkthrough;
- Having roles and responsibilities preplanned such as recorder, reviewer, and any special roles;
- Controlling the checking and logging rate;
- Concentrating on major defects;
- Restricting the structured walkthrough to presenting and consolidating a list of major and minor defects and not allowing solution discussions;
- Collecting the defect data and analyzing it for trends and process improvement.

In fact, structured walkthroughs can be made to be very close in nature to the demands of inspections. The one significant difference is that the structured walkthrough is led by the author. There are possible positive and negative reasons for the author to lead the structured walkthrough. The positive reason is that the author understands the intent of his/her work and can provide that guidance to the reviewers to help them gain greater insight and possibly find more major defects. One negative aspect of the author leading the structured walkthrough is that he/she can lead the reviewers around or away from areas that might contain significant major defects.

I have found reasons to use both inspections and structured walkthroughs for artifacts that are critical, complex, or that have a history of being found with major defects. Although many arguments have been put forth over the years, historical data collected by IBM, TRW, and MITRE Corporation indicates formal inspections are the most efficient form of defect removal.

Steps in the Inspections Process

The inspection process is shown in Figure 13.2. An abbreviated description of each inspection block is offered here.

Entry

The inspection process is only entered when a specified set of entry criteria have been met:

- Is the author of the life-cycle work product ready for inspection?
- Is the life-cycle work product ready for inspection?
- Are all of the source and kin documents identified and available against which the life-cycle work product will be evaluated?
- Are the “exit” criteria for this inspection documented and available?

Planning

The moderator is responsible for leading the inspection planning effort, which includes these tasks:

- Obtain and distribute supporting documents:
    - Life-cycle work product;

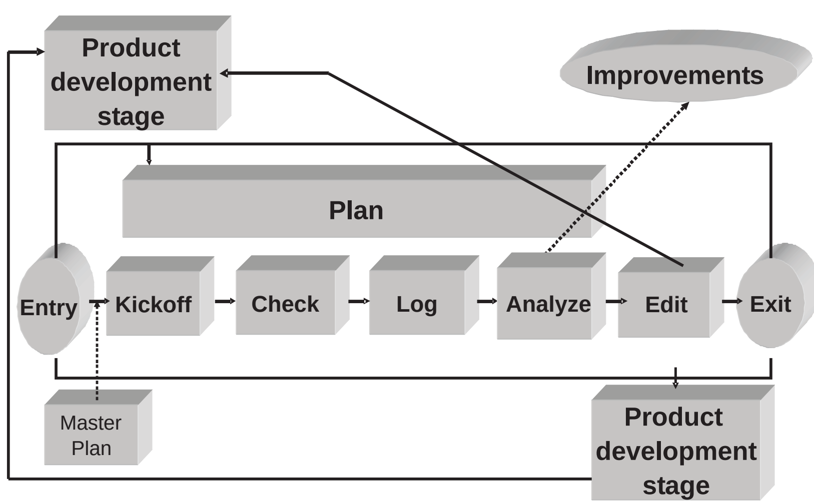


Figure 13.2 Steps in the inspection process.

- Source/rule/kin documentation;
- Checklists.
- Establish optimum checking rate.
- Decide format for life-cycle work product to be inspected (full, sample, chunk).
- Select potential review team members.
- Determine inspection roles, including any special roles.
- Set logging meeting time and date.
- Schedule kickoff meeting, including meeting room, and notifies applicable parties.
- Ensure that all necessary forms are available.

Kickoff

The moderator conducts a "kickoff" meeting prior to checking by completing these tasks:

- Set ground rules or expectations for the inspection.
- Review the inspection process as a refresher to those who have previously participated in inspections or to those who are participating for the first time.
- Get agreement with the team goal from all of the reviewers:
    - Finding the most major defects in the time given (efficiency);
    - Finding the most major defects (effectiveness);
    - Finding the most defects (major and minor).
- Get agreement on the strategy to meet the team goal:
    - Full life-cycle work product inspection;
    - Systematic sampling of life-cycle work product;
    - Chunking.
- Assign standard roles and special roles.
- Review past examples of defects for that type of life-cycle work product from historical data.
- Agree to approach to logging.
- Provide educational overview as needed.
- Make sure that all participants are focused on the correct life-cycle work product and the appropriate rules and checklists.
- Ensure that the reviewers have all necessary forms to use during the checking.

Checking

Each inspection team member inspects the life-cycle work product and logs possible major and minor defects. Note that checking is completed prior to the logging meeting. Reviewers must be given the necessary source/rule/kin documents to review the life-cycle work product against.

Logging

This is the part of the inspection where the reviewers meet face to face to log and consolidate the defects found during their individual checking period. The inspection team concentrates on reporting orally and logging, in writing, major issues, at a rate of at least one per minute. The logging meeting should last a maximum of 2 hours; 1½ hours is an excellent target duration.

Analyzing

The analysis subprocess of the inspection process is a phase I created by separating it from the logging phase. It was done more for psychological reasons, but in many cultures, it works and makes reviewers more comfortable. The author reviews the recorded major and minor defects and answers these questions:

- Are all recorded defects understandable?
- Are there any disagreements with the defect as a defect that can be resolved with a short statement?
- Can the inspection team come to consensus with the author within an agreed-on short time period?

Process Improvement Brainstorming

Immediately, if possible, or shortly after the logging and analysis meeting, time may be allotted to brainstorm the defect causes and process improvements to prevent these causes. Improvements may be made to the inspection process, the source and kin documents, and/or the checklists.

Edit

Some written action, including corrective action, must be carried out by the author or an editor.

- Some written action must be taken on all logged issues—if necessary by sending change requests to other authors.
- The author may request a solution brainstorming session with the other reviewers or anyone else that he/she believes can help to determine the best solution for the documented defects.
- The moderator may review that the defects have been corrected and spot-check the solution on a sampling basis.

Exit

The moderator determines whether the formal exit criteria have been met before signing off on completion of the inspection process. The moderator must determine the following:

- Is the life-cycle work product ready for promotion to the next phase or activity in the life cycle?
- Are additional inspections on this life-cycle work product necessary?

Testing

Testing is a quality control function in that it is used to verify the functionality and performance of life-cycle work products or product components as they move through the product life cycle. Why test? The expected answer would be to find and eliminate defects, but there are other objectives for testing that are significant and worthy of consideration.

Purpose of Testing

The purpose of testing is to:

- Establish confidence that a program or system does what it is supposed to do.
- Make lack of quality visible.
- Execute a program with the intent of finding errors.
- Exercise a component to verify that it satisfied a specific requirement.
- Provide continual assessment of whether the software being produced will meet the needs of the user.

Thought Process Involved in Designing Tests

David Gelprin of SQE fame together with his partner, William Hetzel, came up with the slogan "Test then code." The first time I saw that slogan, I had to think about the message being offered. If an engineer, especially a software engineer, thinks about what it will take to test his/her program, stating the expected testing results helps spot system errors, often without ever running the test.

Key Testing Principles

A number of testing principles should be taken into account before testing activities are assigned:

- *Complete testing is not possible*. It has been stated and agreed to for decades that complete testing of a system is not always possible. If we accept this premise, then what are the guidelines to help determine what should be tested? It is not adequate for any individual involved in the various stages of testing to simply decide to test or not test a component, an interface, or a scenario based on their experience and feelings. What should be tested and to what extent should it be documented according to project/organizational guidelines? What account factors, such as criticality, complexity, interfaces, and past problems with the component if it is a reuse component for this project, should be considered?

- *Testing is creative and difficult.* Testing is not an activity that is given to an engineer simply because he/she is new to the organization, is expendable, or management cannot decide what else to do with the person. In my past management years, I had the privilege of managing a gentleman who had more than 40 years of experience building microprocessors, developing the software to be run on them, and testing the software and hardware together as a system. He was briefly put into a management position, but it was quickly realized that his systems testing skills were so great that it was best to make him a technical fellow, have him continue with his testing activities, and be responsible for mentoring and coaching the next generations of testers.
- *Testing must be planned.* Testing is an intricate set of activities that must be planned just like any other project activity. Planning for testing, including the testing environment, is not a trivial task and should not be minimized. The following aspects should be considered for every test:
    - Test philosophy and criticality;
    - Objectives and completion criteria;
    - Methods;
    - Responsibilities and people involved;
    - Resources and test tools;
    - Budget;
    - Schedule;
    - Documentation;
    - Problem recording;
    - Problem fixing.
- *Testing requires independence.* Although unit testing or "white box" testing is normally carried out by the developers, integration and systems testing requires individuals who are independent from the project developers. It is not human nature for a developer to conduct unit testing and show his boss the defects that were found during that exercise, and then put on a systems test hat and show his boss all of the defects that he found on his components during systems testing. Objective evaluations are required for effective systems testing.
- *Expected results.* A necessary part of a test case is a definition of the expected output or result.
- *Invalid and unexpected input conditions.* Test cases must be written for invalid and unexpected input conditions as well as for valid and expected input conditions.
- *What is the program expected to do?* Examining a program to see if it does what it is supposed to do is only half of the battle. The other half is seeing whether the program does what it is not supposed to do. In other words, we do not want programs to have functionality for which there were no agreed-on requirements.
- *Probability of more errors.* The probability of the existence of more errors in a section of a program is proportional to the number of errors already found in that section. This tautology is often perplexing for developers. If their testing

efforts result in a large number of defects found, they want to be rewarded—and they should be rewarded. But then, they must think about their design and perhaps design additional tests because there are at least as many defects left in the program as they found.

- *Good test case versus successful test case.* A good test case is one that has a high probability of detecting an error that has not been discovered yet. A successful test case is one that detects an error that has not been discovered yet.

Different Levels of Testing

Many different levels or types of testing can be performed throughout the development life cycle. The more standard types are briefly described here.

Unit Testing

Unit testing is a process of testing the individual components, subsystems, hardware components such as programmable logic arrays, and software components such as subprograms, subroutines, or procedures. Unit testing for software focuses on white box or glass box testing. It focuses on test statements, branches, and paths through discrete pieces of code.

The objectives of unit testing may be stated as basic questions:

- Does the logic work properly?
    - Does the component do what was intended?
    - Can the program or component/subsystem fail?
- Is all of the necessary logic present?
    - Are any functions missing?
    - Does the module do everything specified?
- Has any unplanned functionality been added?

Integration Testing

The objective of integration testing is to test component interfaces and confirm that the components meet the interface requirements and that the components can indeed be assembled. The product components need to be checked for quantity, obvious damage to hardware components, and consistency between the product component and the interface descriptions. Integrations and systems test engineers want to obtain a working skeleton as rapidly as possible and establish the confidence that the skeleton parts interface correctly. Integration testing also normally includes demonstrating that simple test cases and transactions are being handled properly by the system.

Systems Testing

Systems testing is the first time at which the entire system can be tested against the systems requirements specification. Systems testing measures and determines what the system capabilities are and ends when the system capabilities have been measured and enough of the problems have been corrected to have confidence that the acceptance is ready to be executed. Systems test planning covers types of testing to be performed, test strategies, test coverage approaches, methods for tracing requirements to test cases, and reliability metrics.

Acceptance Testing

The purpose of acceptance testing is to confirm that a system is ready for operational use and that the confidence built up in systems testing is justified. The primary issue in acceptance testing is usability and reliability—will the system support operational use? Acceptance criteria should be discussed and agreed on in advance of the actual acceptance testing. The focus should be on functional requirements, performance requirements, operational use profile, and recovery requirements.

Acceptance test objectives must be clearly stated together with "success" criteria. Quality criteria such as reliability and maintainability must be acceptance tested. Installation documentation and user documentation must be included in the acceptance testing.

Regression Testing

Regression testing is the execution of a series of tests to check that modifications to parts of an existing system will not negatively affect other working components or subsystems. Regression testing should be applied when a baselined system is being enhanced or repaired. Regression testing criteria should be developed to establish what part of the regression test suite to run and the adequacy of the regression testing.

Test Readiness Criteria

For any level of testing, a component or components must have satisfied its/their readiness criteria. For example:

- Test plans, test procedures, and test cases are reviewed and documented by peers of the developers of the plans and procedures before they are considered ready for use.
- Components have successfully passed a peer review and unit testing before they enter integration testing.
- Components have successfully passed integration testing before they enter system testing.
- A test readiness review is held before the product or system enters acceptance testing.

Configuration Management of Test Items

Key engineering work products should be maintained under configuration management such as:

- Test plans;
- Test cases;
- Test procedures;
- Test scenarios;
- Test scripts;
- Test data.

Test Coverage for Software

Test coverage analysis is the process of finding areas of a program not exercised by a set of test cases, creating additional test cases to increase coverage, and determining a quantitative measure of code coverage that serves an indirect measure of quality.

Statement Coverage

Statement coverage measures whether each executable statement is encountered. Block coverage is the same as statement coverage except that the unit of code measured is each sequence of nonbranching statements. Do–while loops are considered the same as nonbranching statements.

Although statement coverage is a good start on test coverage, statement coverage is completely insensitive to the logical operators (|| and &&) and it cannot distinguish consecutive "switch" labels.

Decision Coverage

Decision coverage measures whether Boolean expressions tested in control structures (such as if-statements or while-statements) are evaluated to both true and false. The entire Boolean expression is considered one true-or-false predicate regardless of whether it contains logical "and" or logical "or" operators.

A disadvantage of decision coverage is that this measure branches within Boolean expressions that occur due to short-circuit operators:

```
If (condition1 && (condition2 | | function1()))
  statement1;
Else
  statement2;
```

This measure could consider the control structure completely exercised without a call to function1. The test expression is true when condition1 is true and condition2 is true. The test expression is false when condition1 is false. The short-circuit operators preclude a call to function1.

Condition Coverage

Condition coverage measures the true or false outcome of each Boolean subexpression. Condition coverage is similar to decision coverage but has better sensitivity to the control flow. Full condition coverage, however, does not guarantee full decision coverage. An example follows:

```
Bool f(bool e) {return false;}
Bool a[2] = {false, false};
If (f(a && b)) ….
If (a [int (a && b) ] ) …
If ((a && b) ? False : False) …
```

All three of the if-statements shown here branch false regardless of the values of *a* and *b*; however, if you exercise this code with *a* and *b* having all possible combinations of values, condition coverage reports full coverage.

Multiple Condition Coverage

Multiple condition coverage measures whether every possible combination of Boolean subexpression occurs. For languages with short-circuit operators (such as C, C++, and Java), an advantage of multiple condition coverage is that it requires very thorough testing. Multiple condition testing is similar to condition testing.

A disadvantage of this measure is that is can be difficult to determine the minimum set of test cases required. This measure could also require a varied number of test cases among conditions that have similar complexity. For example:

```
a && b && (c | | (d && e))
((a | | b) && (c | | d) ) && e
```

To achieve full multiple condition coverage, the first condition requires 6 test cases, whereas the second condition requires 11 test cases.

Path Coverage

Path coverage measures whether each of the possible paths in each function has been followed. Path coverage has the advantage of requiring very thorough testing but it also has two disadvantages:

- The number of paths is exponential to the number of branches. A function containing 10 if-statements has 1,024 paths to test—adding one more doubles the count to 2,048.
- Many paths are impossible to exercise due to relationships of data.

General Observations on Testing

I have had the privilege of conducting appraisals in more than 15 countries and working in 25 countries. Here are some general observations on testing from those experiences:

- Testing still gets squeezed at the end.
- Test results, particularly negative ones, are not respected and are oftentimes disregarded by developers.
- Test data and test case sets are frequently not planned or established.
- The integration test strategy is often left up to personal choice with minimal evidence of any documented standards, guidelines, or procedures.

- System tests are often no more than a collection of developers' unit and integration tests.
- Independent test groups are becoming more prevalent, but they are not staffed by senior software developers with in-depth application and technical knowledge. They are not staffed by engineers with a "testing mentality." Systems testers are not included in up-front planning with other engineering disciplines (e.g., systems engineering, software development, QA, and CM).
- Training for the testing function is frequently minimal or nonexistent.
- Most orgranizations do not have testing methodologies, resulting in unit testing being performed inadequately.
- There is often a lack of test cases, expected results of tests are not established, and integration and systems testing is left solely up to developers.
- Regression testing is seldom performed and does not typically take the complexity and criticality of the system into consideration. There is inadequate understanding of adequacy of regression tests being applied.

Unit Testing and the CMMI®

The reference to unit testing can be found in the Technical solution Process area. Specifically, it is Specific Practice 3.1, Subpractice 4 [2]. Unit testing involves the testing of individual hardware or software units or groups of related items prior to integration of those items. Frequently unit testing for software is left up to the developer. This may or may not be a good idea depending on the complexity or criticality of the module or unit that is being tested. An example unit test procedure is offered here as a "starter kit" on which the reader can base his/her unit tests.

Software Test Planning

The master software test plan contains general, high-level information regarding unit testing, integration testing, and hardware/software system testing. The topics addressed in the master software test plan are listed in Table 13.1. Specific test procedures, descriptions, and so on, are contained in supporting documentation (e.g., integration test specification).

Unit Test Specification/Procedure

The unit test specification/procedure is intended to guide the testing and verification of all functions within each coded program module and subsystem before it is integrated and system tested. Table 13.2 lists the minimum amount of information contained in the unit test specification/procedure. Specific unit test specification procedures and results of unit tests are contained in the software development file and/or technical data package.

The objectives of module testing may be stated as basic questions:

- Does the logic work properly?

Table 13.1 Master Software Test Plan

Unit Test Planning	Describe the requirements, responsibilities, and schedules for their testing. Provide general project standards for unit test thoroughness. Describe how the developer will conduct and monitor unit testing to ensure compliance with this plan. Identify unit test input data that must be supplied by external sources and the plan for obtaining these data.
Integration Test Planning	Describe the integration and test requirements. Identify the organizations responsible for the preparation, execution, reporting, control, review, and audit of the integration and testing. Describe the various types or classes of tests that will be executed during the integration and testing period (e.g., timing tests, erroneous input tests, maximum capacity tests). Describe an overall integration and test schedule that is consistent with overall development plans.
System Testing Planning	Describe the general requirements that apply to all formal tests and unique requirements that apply to selected formal tests. Identify the organizations responsible for the preparation, execution, reporting, review, audits, and control of formal tests. Describe the various classes of formal tests to be executed (e.g., timing tests, maximum capacity tests). For each class of test, describe: Test purpose; Software requirements to be demonstrated; Special software, hardware, and facility configurations to be used; Generic test input environment and output conditions. Identify each systems test and state its objectives. Identify the support software to be used in testing and the plan for obtaining and validating this software prior to the start of formal testing. Identify the computer hardware and interfacing equipment that will be required to conduct the formal testing, including any simulations needed before actual hardware is available. Define the criteria to be used for determining acceptable system performance. Define the protocol(s) to be followed by the developer in reviewing, reporting, and accepting test executions and results. Define the criteria and procedures for incorporating software modifications and performing retest. Describe any assumptions that were made in test planning and test conditions that must be observed during testing to ensure valid results.

    - Does the code do what was intended?
    - Can the program fail?
- Is all the necessary logic present?
    - Are any functions missing?
    - Does the module do everything specified?
- Has any unplanned functionality been added?

Evolutionary Path

The evolutionary path for implementing inspections, testing, and improving on defect prevention—not just defect detection—is described in the steps listed next:

Table 13.2 Minimum Amount of Information Contained in the Unit Test Specification/Procedure

Name of Person	Persons conducting/observing unit test.
Overview	A general description of the approach that will be used to actually perform the unit testing.
Unit Testing Schedule and Budget	Dates unit testing took place or retesting took place.
Roles and Responsibilities	Developer Unit tester (may be the same as the developer). Quality assurance (spot-checks actual results or unit tests against the expected results). Configuration management (project-level CM for updating modules, design, and unit tests, based on unit testing results). Project manager. Hardware or simulation support.
Assumptions	Any assumptions that will be used in performing the actual unit tests (e.g., the assumed state the system will be in when this module or unit is invoked).
Test Techniques/ Approach	A description of the testing techniques that will be used during unit testing. Recommended testing techniques include: Execution of all paths through the unit (if all possible paths cannot be executed, execution coverage criteria should be listed, i.e., statement coverage, branch coverage, or multiple condition coverage); Verification of the correctness of the function; Verification of the computational accuracy of the function; Testing for the function's response to illegal input; Testing for the function's response to saturation conditions; Testing to detect boundary condition errors; Verification of all data output options and formats, including error and information messages.
Unit Test Environment	The hardware and software configuration under which this unit will be tested.
Test Data Required	Type, format, and quantity of test data necessary.
Test Tools	Test tools and test harnesses that are required for successful execution of the unit tests.
Success Criteria	The *expected results* that indicate the module successfully passed the module or unit test.
Support Requirements	Any necessary software or hardware support necessary to conduct this unit test (e.g., test data generators, test drivers, prototype hardware).
Traceability to Requirements	Identifies which requirement or requirements this unit test is supposed to satisfy.
Test Procedure	The actual scenario or test script to be followed in executing the test.
Unit Interfaces	What interfaces must this unit or module be compatible with?
Performance Test	What are the performance requirements that can or must be shown with this unit?
Test Coverage	What test coverage is being achieved by the unit test? Example: Statement coverage; Decision coverage; Condition coverage; Multiple condition coverage; Path coverage; Function coverage; Loop coverage.
Test Reports	What is to be reported and in what format for each unit test? What is the scope of the corrective action required based on the unit test results? Was retesting required following corrective action?
Unit Test Defects	Categorization of unit test defects that can be compared with peer review results as well as the results of subsequent test efforts.

1. Train the development staff on the process of conducting inspections.
2. Provide special training for a group of moderators whose responsibility will be to lead the inspections.
3. Develop the high-level categories of major and minor defects.
4. Ensure that all developers and any other inspection participants follow the inspection process strictly to ingrain the process as it was intended to be followed. Allow no deviations until most developers have exhibited a willingness and a competence in conducting the inspection.
5. Develop finer grained categories of major defects for tracking, analyzing for trends, and process improvement.
6. Provide feedback to the participants of the inspections on the effectiveness of inspections in which they have participated.
7. Control the formality or informality of the inspections through the checking and logging rates.
8. Develop organizational optimal rates for checking and logging depending on life-cycle work product and life-cycle phase and capabilities of the reviewers.
9. Ensure that the testing methodology is documented and audited by quality assurance for compliance.
10. Develop readiness criteria for movement between component development (code for software) and unit testing, unit testing and integration testing, and integration testing and systems testing, and ensure their enforcement.
11. Develop equivalent categories of major defects for the testing phases.
12. Start comparing the types of defects found in the inspection of life-cycle work products in the early phases of the product life cycle with those found in the respective testing phases.
13. Perform causal analysis activities to determine targets of improvement on suspected weak processes.
14. Collect and analyze the defects found after the product has been placed into production for 1 to 2 years.
15. Start to compare the after-production defects found by the customer with those categories of major defects found in the testing phases and through inspections of the life-cycle work products in the early phases of product development.
16. Perform formal causal analysis on the defect types that make it through all of the inspections and testing and appear during production use.
17. Improve processes that contribute the most toward finding and eliminating the defects that affect cost, schedule, performance, quality, risk, and customer satisfaction when the product is placed into production.
18. Track the improved processes to determine their impact and cost effectiveness.
19. Improve the processes at the organizational level.
20. Train, mentor, and coach the implementation of those improved processes on all projects throughout the organization.

Summary

A review is an evaluation of a life-cycle work product(s) or project status to determine if there are any deviations from planned results and to recommend improvement. We conduct reviews to detect and remove defects as close to the point of insertion as possible, to identify potential improvement in the work product and the process used to develop it, and to reduce costs to build and maintain better products. We can also conduct management or progress reviews that help us answer these questions:

- Are we ahead or behind where we expected to be?
- Will we complete the work as planned?
- Is the required functionality being implemented?
- What risks are we taking based on the project information that we have today?

Peer review types that can be used for different objectives and expected results include buddy checks, circulation reviews, technical reviews, walkthroughs, and inspections.

An inspection is a formal verification technique in which life-cycle work products are examined in detail by a group of peers for the explicit purpose of detecting and identifying defects and is the most effective peer review technique.

Testing is a quality control function as it is verifying the functionality and performance of life-cycle work products or product components as they move through the product life cycle. The purpose of testing is to:

- Establish confidence that a program or system does what it is supposed to do;
- Make lack of quality visible;
- Execute a program with the intent of finding errors;
- Exercise a component to verify that it satisfied a specific requirement;
- Provide continual assessment of whether the software being produced will meet the needs of the user.

Test coverage analysis is the process of finding areas of a program not exercised by a set of test cases, creating additional test cases to increase coverage, and determining a quantitative measure of code coverage that serves an indirect measure of quality.

Peer reviews (inspections) combined with effective testing can find and detect most of the major defects before a product is put into production without being required to do 100% software inspections or 100% path coverage in testing.

Peer reviews and unit testing are the product component verification techniques suggested in the Technical Solution process area that prepare the product component for integration and systems testing. Without applying these verification techniques judiciously to the product components, a project loses control of its integration process with the result being late deliveries, poor quality, nonworking functionality, and rework!

References

[1] Ebenau, R. G., and S. H. Strauss, *Software Inspection Process,* New York: McGraw-Hill, 1993.

[2] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI® Second Edition, Guidelines for Process Integration and Product Improvement,* Reading, MA: Addison-Wesley, 2007.

CHAPTER 14

From Components to Products: Gluing the Pieces Together

This chapter presents the CMMI® process areas of Product Integration, Verification, and Validation as a "triple." Here we show how their use guides projects from the building blocks developed during the Technical Solution phase to an integrated, verified, and validated set of product components that is ready for packaging and delivery. We also emphasize and reemphasize the importance of interfaces based on their own merit as well as from the CMMI® point of view.

The Product Integration process area addresses the integration of product components into either more complex components or subsystems, or into complete products. Although it is not wrong to think of product integration as a one-time assembly of the product components, it is normally an iterative process of assembling product components, evaluating them, and then assembling more product components until the complete product is assembled and the system is fully tested.

Product Integration can be a very deceptive process area to those who read its goals and practices as if all of the events were to be executed in sequential order. Its real value comes to the surface if one thinks about the Product Integration process area as a main software program that must be executed to assemble the product components, and Verification and Validation as its software subroutines that are called to perform peer reviews, testing, simulation, and other verification and validation activities.

The Integration Strategy

The performance of effective product integration involves the establishment and the maintenance of an integration sequence, the environment for performing the integration, and the development and use of integration procedures.

An integration and test strategy should be developed early in the project, concurrently with product development plans and specifications. Some life cycles require that the integration and test strategy be one of the very first documents developed for the project following the successful baselining of the allocated requirements.

During the establishment of the product integration strategy, the following 11 questions should be answered:

1. When will the product components be available?
2. Which ones are on the critical path?

3. What alternative integration sequences have been considered?
4. What work needs to be done to prepare and conduct the integration activities?
5. Who is responsible for each integration activity?
6. What resources will be required?
7. What schedule is to be met and what are the expectations?
8. Are the necessary procedures to be followed documented and in place?
9. Are any special tools required during the integration?
10. What must be included in the product integration environment?
11. What personnel skills are required for the individuals conducting and supporting the integration?

Other considerations include:

- What modules should be integrated first?
- How many modules should be integrated before integration testing starts?
- What order should be used to integrate the modules?
- Should there be more than one skeleton?
  - How is each skeleton defined?
  - Are there distinct build levels?
- How much testing should be done on each skeleton?

Alternatives for the order of product component integration include top-down product component integration, bottom-up product component integration, critical product components first, related functional product components first, as-available product components, and complete product component integration.

The Integration Environment

Of great importance is the establishment of the integration environment needed to support the integration of the product components. The requirements of an integration environment can vary widely depending on the product components that are integrated into the deliverable product. For contrast, imagine the integration environment needs for a software accounting system. Now compare these to the integration environment required for the development of a Boeing 747 aircraft. The space, safety concerns, assembly equipment, required personnel, tools, recording equipment, and so on, take a long time to plan and set up to support the integration for a Boeing 747. This integration environment is considerably different both internally to the systems themselves and externally to the users.

The environment may include test equipment, simulators, pieces of real equipment, and recording devices. It may be purchased or developed. The product integration environment may also include the reuse of existing organizational resources as long as reuse is planned for early enough in the project life cycle.

It is also quite possible for the integration, verification, and validation activities to share the same environment. One IT company in Asia and one in Central Europe set up “mirror” environments for their development, integration and systems test, and production functions. This company believed that it was cheaper in the long run to duplicate these environments with the same software and hardware rather than to worry about what effects partial or different environments might have on the development, implementation, and test results.

Product Integration Procedures

To carry out the integration sequence, product integration procedures must be developed or, if they already exist, reviewed for usefulness. These procedures may include guidance on the number of planned iterations and expected test results for each iteration. Criteria should also be established indicating the readiness of the product component for integration. Criteria may include the degree of simulation permitted for a product component to pass a test.

Readiness for Integration: Interface Requirements

One of the most important readiness criteria needed for product integration is the assurance that all of the product components are confirmed to be compliant with their interface requirements (Figure 14.1). Interface requirements drive the development of the interfaces necessary to integrate the product components.

The interface requirements may have been collected during the requirements elicitation phase and expanded during product architecture development, and they must be managed throughout the product life cycle. Interfaces are usually classified into three main classes: environmental, physical, and functional.

Typical categories for these classes include the following:

- Mechanical;
- Fluid;
- Sound;
- Electrical;

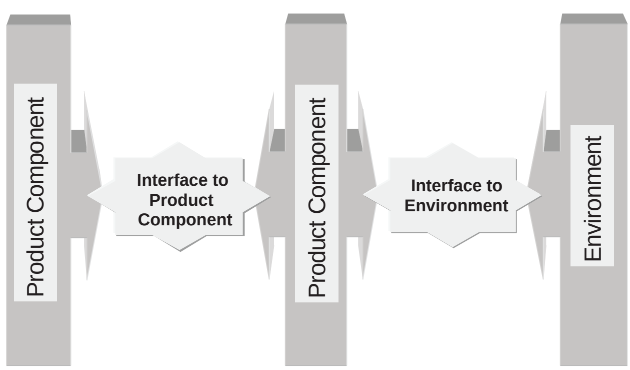


Figure 14.1 Ensuring interface compatibility.

- Climatic;
- Electromagnetic;
- Thermal;
- Message interfaces;
- Human–machine or human interface.

The interface descriptions defined during requirements evolution and the definition of alternative solutions must be placed under configuration management as one of the project's *most important* configuration items. Change requests to these interface descriptions must strictly follow the change control process. Many product integration problems arise from unknown or uncontrolled aspects of both internal and external interfaces. Effective management of product component interfaces helps ensure that implemented interfaces will be complete and compatible.

To ensure product component readiness for integration:

- Ensure that the product components are delivered to the product integration environment in accordance with the planned product integration strategy.
- Verify the receipt of each product component.
- Ensure that each product component meets its description.
- Check the configuration status of the product component against the expected configuration.
- Confirm that each product component is compliant with its interface requirements.

Ensuring interface compatibility and getting prepared for the assembly of the product components also includes *interface testing*. When organizations speak on the integration and systems testing phase, they rarely also speak of interface testing. The truth is that interfaces often do not get tested for one reason or the other and the cost and schedule pressure are typically listed as the culprits for this oversight. While teaching product integration during a CMMI® workshop, I often joke that systems testers are doing a bad job! This has the desired shock effect, and participants are curious as to how I am going to dig my way out of this one, especially those who have the job of systems testing. My response goes like this:

- Projects within organizations do not often carry out rigorous peer reviews on critical life-cycle components during the development stages.
- Unit testing is often left to the whims of the developers especially in software.
- Interfaces are not tested.
- The entire mess is passed along to systems test.
- The systems testers normally have half of the time requested for systems testing taken away from them.
- They try like crazy anyway and slog away at finding the defects that have been injected into the system.
- When they find defects and notify the developers, they are yelled at.

It therefore seems logical to come to the conclusion that systems testers do not do a good job. Of course this is a joke, but the point of making sure the interfaces are tested before systems testing of the product components is conducted is not a joke.

A reliability colleague of mine told me about a year ago that the systems testers for a major satellite project admitted they did not do a good job of testing the interfaces of components because it was too hard! The satellites only cost about $250 million.

Assembly of Product Components

Once the readiness criteria of the product components have been established, the actual process of integrating these product components can commence. This is normally expected to be done in an iterative fashion from the initial product components, through the interim assembly of product components, to the product as a whole. The readiness of the product integration environment should be ensured and the product components assembled according to the product integration sequence and integration procedures.

Evaluation of Assembled Product Components

Throughout the evolution of the product or product components, each interim state or final product state must be evaluated to show that they satisfy the functional, performance, and quality requirements. Actual product evaluation results should be compared against expected results and spot-checked by an independent party, such as quality assurance. The assembled product components must, therefore, be verified and validated according to the integration sequence and the verification and validation strategies.

Verification

Verification focuses not only on the final assembled product, but also includes verification of the intermediate work products against all selected requirements, including customer, product, and product component requirements. Verification methods address the technical approach to work product verification and the specific actions, resources, and environments that will be used to verify specific work products. Verification typically begins with involvement in the definition of product and product component requirements to ensure that these requirements are verifiable. The verification environment must be appropriate to support the verification method and may be shared with product integration and validation.

When verification activities are performed, the results must be documented. This includes the "as-run" verification method and the deviations from the strategies and procedures that were necessary during its execution that may be used later for process improvement purposes. During any verification activity, it is important to remember that the end goal is to ensure that each product component of a system satisfies its requirements.

System verification includes functional, physical, interface verification as well as word product verification. Functional verification includes *system performance testing,* which verifies performance with respect to the requirements, and *qualification testing,* which verifies system performance within its specified operational environment (e.g., temperature, vibration and shock, electromagnetic interference, electrical, flow system, and telecommunications).

Verification Techniques and Methods

Verification techniques ensure that the integrated product meets the specified requirements. In other words, it ensures that the project built the product right. Methods of verification include, but are not limited to:

- Inspections;
- Peer reviews;
- Audits;
- Walkthroughs;
- Analysis;
- Demonstrations;
- Comparisons;
- Path coverage testing;
- Simulations;
- Functional testing;
- Qualification testing;
- Factory testing;
- First article qualification;
- Bench testing;
- Load, stress, and performance testing;
- Operational scenario testing;
- Observations and demonstrations.

Verification methods commonly considered during system testing applied prior to packaging and delivery include:

- Load, stress, and performance testing;
- Functional decomposition-based testing;
- Operational scenario testing.

Systems Testing Versus Bench Testing

System testing can be defined as measuring a product component's performance while installed in a real or reasonable approximation of a functional system. Bench testing, in contrast, is defined as the insertion of a product component into a test loop where all of the variables can be independently controlled, measured, and recorded. Bench testing is also known as the process of characterizing the failure mode of a sample using the various bench equipment to stimulate the sample and measuring its response.

Validation

System validation is an end-to-end process that is needed to ensure that the completed and integrated system will operate as needed in the environment for which it was intended. It can be defined as a measure of customer satisfaction, given the customer's operational need and profile. System validation must always take into account the customer/end user during testing.

The development of validation procedures should include the test and evaluation of maintenance, training, and support services. As pointed out in Chapter 11, validation activities are performed as early in the product life cycle as possible, starting with the customer and product requirements. As the validation activities are performed, the "as-run" validation procedures should be documented and the deviations that occurred noted.

During validation it is important to remember to:

- Demonstrate that the maintenance tools are operating in the actual product,
- Verify in the field that support of the product is effective as specified by the customer (e.g., mean time to repair),
- Demonstrate adequate training of the products and services.

The final task during validation is to answer the question "Did the product perform as expected in its intended operational environment when used by the end users required to work in that environment?"

Validation procedures and criteria are established to ensure that the product or product component will indeed fulfill its intended use when placed in its intended environment. To validate a product means to demonstrate that you have built the right product. Validation methods should be selected based on their ability to demonstrate that user needs are indeed satisfied.

Validation Techniques and Methods

The CMMI® and other systems engineering sources offer examples of validation methods that include:

- Discussions with the users during the early phases of the life cycle;
- Prototype demonstrations;
- Simulation demonstrations especially with the end users present;
- Structured scenario testing;
- "Break it" testing to uncover unintended behaviors;
- Emulation of hardware components;

- Modeling (of hardware to validate form, fit, and function of mechanical designs);
- Reliability analysis;
- Functional demonstrations such as for the system, hardware, software, service documentation, and user interfaces;
- Pilots of training materials.

Acceptance Testing

Acceptance testing may satisfy the final verification and validation criteria. The purpose of acceptance testing is to confirm that a product or product component is ready for operational use. The acceptance test is performed for, or in conjunction with, someone else to demonstrate that the confidence is justified. The primary quality factors addressed are usability and reliability in order to answer the question "Will the product or product component support operational use?" Acceptance testing should also include the review of user documentation to verify its technical correctness compared to the system functionality, its completeness, and its ease of use.

Packaging and Delivery

Prior to packaging and delivering the product, the requirements, design, product, test results, and documentation are reviewed again to ensure that all issues affecting the packaging and delivery of the product are identified and resolved. Configuration audits are conducted prior to packaging and delivery to ensure the following:

- The product or product component that is included satisfies the customer and product requirements and all approved change requests and nothing more.
- The documentation that is to be delivered to the customer/end user matches the delivered product or product component.

Verification and validation results that have been conducted throughout the development life cycle are used as input to this final configuration audit.

The packaging requirements for some products can be addressed in their specifications and verification criteria, which are especially important when items are stored and transported by the customer.

Other important factors include environmental and stress conditions:

- Economy and ease of transportation (e.g., containerization);
- Accountability (e.g., shrink wrapping);
- Ease and safety of unpacking (e.g., sharp edges, strength of binding methods, childproofing, environmental friendliness of packing material, and weight).

Final delivery includes satisfying the applicable packaging requirements, preparing the operational site for the installation of the product, delivering the product and related documentation, and ensuring that it can be installed at the operational site.

Summary

The steps leading up to product delivery include establishing the integration environment, developing necessary integration procedures and criteria, ensuring that all product components meet their interface descriptions, integrating the product components according to the integration sequence, assembling the product components, and evaluating the product components, subsystems, and eventually the full system utilizing verification and validation techniques throughout the project life cycle as appropriate. Sequences of triples may be required to be carried out before the final product is assembled, verified and validated:

- (PI_1, VER_1, VAL_1),
- (PI_2, VER_2, VAL_2),
- (PI_3, VER_3, VAL_3),
- …,
- (PI_N, VER_N, VAL_N).

A delivered product is not considered to possess quality unless it can be shown that it satisfies all of the specified requirements and performs as anticipated in the operational environment by the customers or end users who have to use it.

PART IV

An Organizational Focus

CHAPTER 15

Improving Processes at the Organizational Level

Since Watts Humphrey started the process program at the Software Engineering Institute in 1986, he emphasized having an organizational focus on developing processes. He also stressed the need to have process improvement champions at all levels while making the statement, "Process improvement is not going to happen by itself." Humphrey would follow up an assessment with a recommendation to all organizations that they should establish a process group that would facilitate the process improvement program for that organization if they had not already done so. His thinking was that the SEPG or engineering process group (EPG) today would not only facilitate the development of the necessary processes but also serve as an example by "acting" or "behaving" at a maturity level at least one above where their organization was assessed. Thus, an organization would not only have champions who kept a constant vigil on the process improvement activities but champions who would serve as an example for other projects on "how it should be done."

This chapter describes the organizational components necessary to establish and keep the organization on a path of continuous process improvement. It includes the CMMI® process areas of Organizational Process Focus and Organizational Process Development. It will provide a sample process improvement infrastructure and description of the roles and responsibilities that are necessary to make a process improvement initiative successful. The description of the organizational process development process area will emphasize the various components that must be in place before an organization can claim compliance with the requirements and guidance provided by this process area.

Focusing Your Organization's Process Improvement Efforts

One of the most important activities the process group performs is determining the strengths, weaknesses, and improvement opportunities for the organization's processes. This is normally carried out relative to a process standard such as the CMMI®. It is important to remember that the organization's vision and business objectives must be supported by the process improvement effort. In all of my experience, it is rare that a process improvement initiative will be kept going and be successful if it is not directly linked to the vision and business objectives set by the senior management team. It is wise to consider the issues related to finance, quality, human resources, and marketing for process improvement and not just technology.

It is also important to identify the process performance objectives such as time to market and product and service quality when putting together a process improvement initiative for your organization. Other commonly used process performance objectives include:

- Earned value;
- Cycle time;
- Percentage of defects removed by type of verification activity;
- Peer review coverage;
- Test coverage;
- Peer review effectiveness;
- Test effectiveness;
- Defect escape rates;
- Defect removal efficiency;
- Number and density of defects (by severity) found during the first year following product delivery;
- Rework time as a percentage of total project life-cycle time.

What Processes Currently Exist?

To focus the organization's process improvement efforts, it is necessary to look at what currently exists and what constraints the organization must work within and also to examine other inputs that can provide valuable process improvement hints and suggestions. To start the process, process group members may choose to identify the policies and standards applicable to the organization's process. They may also examine relevant process standards and models for good and best practices.

To further understand and define or refine the characteristics of the organization's processes, the process group can evaluate processes that are currently being used in the organization, process and product standards imposed by the organization, and process standards and product standards imposed by customers/end users.

Assessment or Appraisal

A classic approach to determining the organization's process capabilities is to conduct an assessment or appraisal. Process appraisals normally result in:

- Determining the strengths and weaknesses of the organization's processes;
- Creating improvement recommendations for the organization's processes;
- Understanding of the business consequences of following or not following organizational processes.

Other Process Improvement Sources

Although an assessment is a very effective mechanism for determining an organization's process capabilities, other candidate process improvement sources can offer insight into the organization's and project's processes. Some of these sources include:

- Quality assurance evaluation of processes;
- Measurement and analysis of the processes;
- Effectiveness and suitability of deployed processes;
- Lessons learned from tailoring the organization's set of standard processes;
- Lessons learned from process implementation;
- Process improvement proposals,
- Senior management inputs,
- Results of process appraisals,
- Peer review results and defect trends;
- Testing results;
- Trouble reports;
- Results of other organizational initiatives.

Realizing a successful process improvement program requires the cooperation and coordination of all levels of management and practitioners—process improvement for any organization can never be the sole responsibility of the process group. Figure 15.1 illustrates a sample process improvement infrastructure. Descriptions of the components of that infrastructure follow.

Sample Improvement Infrastructure

The sample improvement infrastructure shown in Figure 15.1 was based on the original software engineering process organization developed in 1988 by Westing-

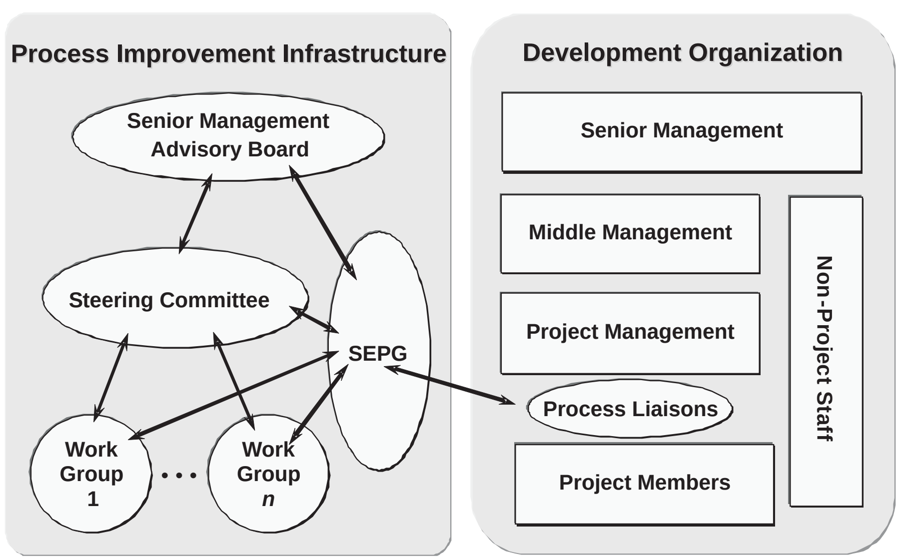


Figure 15.1 Sample process improvement infrastructure.

house Electronic Systems Group in Baltimore, Maryland. Westinghouse ESG established the steering committee, the working groups, and the EPG. This model was subsequently adopted by the Software Engineering Institute and published in an SEI technical report [1].

The sample process improvement infrastructure is shown in Figure 15.1 on the left side of the diagram and is composed of members from the organization, as represented by the right side of the diagram. This in no way indicates that an organization needs to reorganize itself in order to engage in process improvement. It merely illustrates, as stated earlier, that process improvement requires the support of all levels of management and the practitioners. Senior managers, middle managers, project managers, and practitioners must be committed and visibly participating to make process improvement work.

Senior Management Advisory Board

A senior management advisory board is necessary to initially and then continuously share the organization/business unit's vision and business objectives. This ensures that the process improvement effort supports that vision and provides measurable improvements to support the business objectives. It is vital to the success of the process improvement effort for each person in the organization to be able to relate the changes they see in their immediate environment and daily work to the broader goals of the organization. People at all levels like to know that what they are doing is supporting the overall health of the organization. They want to be a part of success. They do not want to start projects and then stop them for no apparent reason.

Understanding the vision and business objectives helps all individuals to realize the value of their individual efforts and the value of the process improvement effort. Only visible senior management support can provide that guidance. Senior management must demonstrate regular and consistent visible support for specific change actions and for continuous process improvement in general. Senior management needs to proactively let the organization's members know why the process improvement initiative is indeed one of the organization's most important projects and to communicate to the organization why the proposed changes are needed.

Last but not least, senior management needs to ensure that the necessary resources for the process improvement initiative are available in a timely manner. This does not, however, mean that the senior management team must approve and support everything required for improvement at one time. Improvement efforts like any other project have to be prioritized and have to support the organization to stay in business so they can improve and become more competitive in the future.

Process improvement efforts require:

- Dedicated people for the steering committee, working groups, and the EPG;
- Training courses at many levels;
- Computers;
- Databases;
- Internal and external consultants;

- "Hand-holding" support for the project teams as they strive to accomplish their project objectives while incorporating new processes and procedures to accomplish those tasks.

Virtually all of the process improvement success stories have had strong senior management involvement. Dr. Edwards Deming stated that the process belonged to the senior management. It is the senior management that must be willing to support the process improvement initiative or it will flounder and possibly die altogether. That was true 50 years ago and it remains true today.

Proposed membership on the senior management advisory board is as follows:

- Organization senior manager and/or product line manager;
- Key line managers;
- Key product managers;
- Quality director;
- Chief financial officer;
- Marketing director;
- Process manager (also on steering committee).

Steering Committee

Without senior management involvement, the process improvement initiative may flounder and not be oriented toward supporting the organization's business objectives. Without middle management involvement, the critical process improvement resources and individuals needed to work on specific focus area improvements may not be made available. Middle management must be on board to ensure the "right" people are provided where they are needed for as long as they are needed.

Westinghouse Electronics Group recognized the critical role that middle managers play in process improvement as early as 1988 and created the concept of a steering committee. The steering committee became the management implementation mechanism that represented senior management on a daily basis.

Middle managers constantly receive pressure from above to guide projects under their authority to completion on time, within budget, and with the promised functionality or better. Middle managers also receive constant pressure from below in that product developers are continually asking for new state-of-the-art tools including computers, databases, languages, file servers, workstations, web-enabled applications, and so forth. Each request comes with a high price tag but with the promise that the productivity may be increased by 10%, 15%, or even 25%.

The process improvement initiative can be invaluable because it can give control back to middle managers so that they are in a position to make informed risk management decisions based on data and not emotion.

Too many organizations allocate to few and often unqualified resources to EPGs and working groups. Individuals with software/hardware/systems experience and product line knowledge are required for a truly successful process improvement initiative.

If individuals are to devote 20% of their time to a process improvement working group, that 20% must be made as important to them as the 80% they are being asked to devote to project work. They must be evaluated for their contributions to the process improvement effort in the same manner they are evaluated for their development efforts. In addition, if an individual is asked to devote 20% of his/her time to process improvement, this must not be 20% above the 120% they are already working. It is middle management who owns these resources and who must balance their work to accomplish business goals set by the senior management team. It is middle management who must protect the time allocated and spent on the working groups, or the entire process improvement effort will be in danger from day 1.

The steering committee is normally involved in the following activities:

- Ensuring that the process improvement activities are in line with the vision and business objectives that have been established by the senior management advisory board by:
  - Reviewing the proposed budget for the improvement effort;
  - Making recommendations to the senior management advisory board regarding program direction, budget, and program risks.
- Ensuring that the necessary resources for the working groups and EPG are available in a timely fashion by:
  - Establishing the working groups to concentrate on prioritized focus areas (e.g., commitment process, estimation procedures, testing methods);
  - Supporting, where needed, negotiations for people's dedicated time to the process improvement effort.
- Conducting process improvement program oversight reviews on a periodic basis (recommended once per month) by:
  - Ensuring that process improvement activities progress are in line with documented budgets and plans;
  - Performing reviews and approval of working group deliverables.
- Providing *visible* support for the EPG and working groups.

Proposed membership on the steering committee includes:

- Functional managers;
- Project managers;
- Chief systems engineers;
- SQA manager;
- HQA manager;
- Software process manager.

It is appropriate to make the case at this time for the middle manager to be considered the "process owner." Process owners are often associated with those individuals who have the most technical expertise and interest in leading a working group or process action team. However, as argued earlier, it is the middle managers who truly own the resources that must make the project successful and the process improvement efforts happen. By making the middle managers the "process owners," the senior manager is ensured that his/her vision and business objectives are being considered for each piece of the process improvement effort. Furthermore, it becomes the middle managers' responsibility to ensure that the working groups have the proper resources, tools, and guidance to develop new processes or revise existing ones and get them deployed on the projects.

The middle managers are not responsible for actually doing the process development work; instead, they are working with the working group that is taking care of improving the process areas for which the middle managers are responsible. This means that the middle managers must communicate with the EPG and the working group and understand the issues being faced along with the progress and future activities. They can then offer guidance to the working groups based on their interactions with senior managers and can report back to senior managers with accurate process improvement progress updates. From my experience, in every instance when middle managers have been made process owners, the organizational process improvement initiative has had measurable success.

Engineering Process Group

The software/systems engineering process group, process action group, process group, process improvement group, or whatever name is given to it is an organizational focal group for action planning, process improvement, technology insertion, training, and awareness and expectation setting. EPGs are frequently viewed as a channel for institutionalizing the organization's knowledge of process methods, practices, and technology. EPGs are the organization's "champion of change" and its members "change agents." An EPG must facilitate the process improvement efforts at the organizational, project, and individual level.

What does the EPG have to know? Collectively, the members of the EPG need to be able to demonstrate their ability to manage, develop, coach, and guide process improvement initiatives and their accompanying cultural changes. First and foremost they need to understand senior management's vision and the organization's business objectives to be able to efficiently and effectively guide the process improvement effort. Without this explicit knowledge, the organization's process improvement effort may demonstrate compliance with a model such as the CMMI® but not be supportive of the organization's business needs at all.

EPG members must have a solid engineering background. They must have a general knowledge of the organization's application domains and knowledge of modern software/systems engineering techniques and methods. They must be up to date on the accepted software/systems engineering standards (DoD, MoD, IEEE, ISO, ESA, NASA, and so forth). They must also have a good understanding of the quality management functions such as quality assurance and configuration management. They must be respected by the managers and product engineers alike. They must have a strong knowledge and good experience in project management and a working knowledge of metrics to help the projects manage and control their projects better.

EPG members must be people oriented with superior communication skills and willing to perform most of their work in the labs of the product developers who need their support to understand just how the process improvement ideas fit into their daily lives. They should always be ready to provide "hand-holding" support for the managers and practitioners on the various projects where the process ideas are being introduced.

Although EPG members must have the technical background to maintain credibility with the product developers, they must also be knowledgeable about organizational development skills (i.e., managing technological change, team building, collaborative consulting) to effect successful technology transition.

The EPG should strive to show by example how process improvement should be accomplished; that is, EPG members should "walk the talk."

Although the many tasks attributed to the EPG are important ones for its members, it must be stressed that the job of the EPG is to be the champion of the process improvement effort. The EPG is expected to facilitate the process of change—*not be responsible for the process change*. EPG responsibilities include, but are not limited to:

- Coordinating the process improvement initiative up, down, and across the organization by:
    - Participating in senior management advisory board reviews;
    - Participating in steering committee reviews;
    - Facilitating the activities of the working groups, which means staying on top of what is going on, what difficulties are being encountered, and what successes are being realized;
    - Promoting technical awareness and education about process improvement (this is a continuous job for the EPG); the process liaisons help the success of this function.
- Managing and facilitating the process improvement initiative by:
    - Facilitating the definition/improvement of the technical and managerial processes, methods, techniques, and tools for developing and maintaining products and product components;
    - Assisting in the evaluation of new tools and techniques based on their understanding of the existing processes;
    - Facilitating the definition and maintenance of organization policies and standards for processes and products;
    - Discovering "good practices" and getting them adapted for general use on the projects throughout the organization and baselining them as "best practices";
    - Overseeing and facilitating pilot projects and implementation of improvements into the projects and across the organization;
    - Directing the definition of process metrics, initiating the collection of data, and assisting the working groups and projects in the analysis and use of the resulting information.

- Ensuring that the processes are "living" by:
  - Maintaining a dialogue with project personnel regarding the application and performance of developing processes (sharing good ideas from other parts of the organization and listening to issues/ideas from the practitioners);
  - Initiating periodic process improvement progress checks and reassessments;
  - Initiating practitioner-driven review of specific processes.
- Maintaining a process asset library of product and product component process assets by:
  - Overseeing the process asset library for product and process assets used across the organization;
  - Facilitating the development and retention of tailoring guidelines for specific use of the assets in the process asset library.

An EPG must have some full-time members. Watts Humphrey suggested that 1% to 3% of the product development and maintenance budget should be devoted to the EPG. It is certain that one EPG member is not enough. The EPG may have part-time members as well, but it should be recognized that less than 50% of a person's time will not be very effective. Proposed membership in the EPG includes these personnel:

- Process improvement manager (EPG leader);
- Process champions (people who are motivated, respected, and knowledgeable and can be team players);
- Associate members who may be drawn from pilot projects and/or product segments;
- Associate members who may represent functions, product line, or other affected organizational entities.

Corporate Engineering Process Group

Before we leave the topic of the EPG, it is worth examining corporate-level EPGs or enterprise EPGs versus the organizational EPG that is referenced in the CMMI®. Large organizations may be tempted to simply establish a corporate-level EPG and give it the charter to develop generic processes that will be used by all organizations/business units under the control of the corporation. It would be erroneous to believe that generic EPG activities in a corporate headquarters location would automatically bestow maturity-level benefits on division locations elsewhere. The corporate-level EPG in large companies can and should develop generic processes for use at the organizational/business unit. But in doing so, the division must accept the responsibility and be given the authority by the corporate-level EPG to review and adapt or tailor the corporate-level process description. The division must also ensure that the projects within the division are able to effectively use the adapted processes and produce the necessary quality products and product components.

A corporate-level EPG can be a tremendous corporate asset and ensure that the corporation's divisions follow common processes where appropriate and apply corporation-authorized tailoring guidelines to enable the development and use of organizational/divisional process descriptions where they are needed. A corporate-level EPG should not be asked, directed, or expected to ensure a Maturity Level 2 or Maturity Level 3 compliance at a lower divisional level. A divisional-level EPG is needed to guide that achievement.

Process Improvement Manager

Each process improvement infrastructure should have an identified process improvement manager. This individual is a senior person with most of the attributes listed in the section on what EPG members should know. The process improvement manager is responsible for coordinating all of the process improvement activities throughout the organization. He/she serves as the link among the senior management advisory board, the steering committee, and the EPG. The process improvement manager also serves as the link between the process improvement initiative and the organization's line, function, and project management.

Working Groups

Working groups or process action teams/process improvement teams are the most common mechanism used for the development and/or revision of processes that will be piloted on selected projects and eventually implemented throughout the organization. Working group members may be involved in the development of the action plan for a particular focus area, may be involved in supporting the implementation of a process or procedure on a project or projects, or both. Some working group members are committed to projects long term to ensure continuity. Working group members may be involved with the action planning up to 100% of the time, but are normally involved with the subsequent process development, process revision, problem fixing, and so forth, and the associated implementation and support 20% of the time. EPG members often facilitate multiple working groups and coordinate the activities of the working groups to avoid duplication of effort.

Working group members are interested in contributing to the improvement of a particular focus area and usually have some background knowledge and experience in that focus area. Working group members are involved with supporting process improvement in a focus area for at least 9 months but normally rotate out after 1.5 years. Examples of activities that working groups take on are:

- Developing the action plan for a specific focus area;
- Developing new or improved processes, procedures, guidelines, templates, and so on, based on a formally documented and approved plan;
- Identifying, screening, and evaluating technologies based on the organization's and project's processes;
- Suggesting and/or developing training plans;
- Supporting the piloting of those processes;

- Evaluating the pilot performance and revising the processes as necessary;
- Reporting process improvement progress to the steering committee;
- Sharing lessons learned with other working groups and project members;
- Supporting the institutionalizing of the tested and approved processes.

Proposed membership of the working groups is as follows:

- Working group leader (high percentage effort: =50%),
- Core members (high effort: 40% to 50%),
- EPG representative (moderate to high effort: =30%),
- QA representative (low to moderate effort: 20%);
- Members (low to moderate effort: 20%);
- Internal consultants (low effort: 10%);
- Reviewers (low effort: 5%).

Process Liaisons

Process liaisons were added along with the senior management advisory board in 1993 to the sample improvement infrastructure. The concept of process liaisons is very simple. Usually, on a project there is at least one person who is quality minded or process improvement minded. Although this person may not be ready to change his/her career and join the EPG full time, he/she is probably interested in process improvement and willing to spend about 5% of his/her time keeping up with the process improvement progress and sharing process improvement news with the other project members. Process liaisons can help transition the process improvement ideas into their projects.

General activities for process liaisons should include the following:

- Act as a point of contact between the process improvement organization and the project practitioners:
  - Keeps project members informed of process improvement activities;
  - Provides input to the EPG/working groups regarding good practices being performed on the project and possible candidates for working group involvement;
  - Serves as the process improvement advocate for the project.
- Provide information and feedback to the EPG and working groups regarding:
  - Issues impacting development performance;
  - Use of new processes in the project.

Establishing, Maintaining, and Implementing Action Plans

Process action plans are detailed implementation plans. These plans differ from the overall organizational process improvement plan. They are normally focused on improving processes that belong to one or more related process areas. These process action plans normally address weaknesses revealed by appraisals or one of the other of candidate process improvements sources.

Figure 15.2, taken from the Kasse Initiatives Action Focused Assessment method, is a generic working group model that can be used to establish, maintain, and implement action plans. The steps discussed next provide the details behind the flowchart shown in Figure 15.2.

Provide Management Direction

Understand the vision, business objectives, quality goals, and priorities from the senior management team. Ensure the process improvement effort is in line with the vision and business objectives of the organization. Management direction and authority are necessary to ensure that allocated resources are protected and unforeseen obstacles are dealt with appropriately. Identify the members of the steering committee and from what organizational departments they should come.

Create the steering committee to:

- Review/approve process improvement priorities;
- Conduct monthly process improvement progress checks with the EPG and the working group chairpersons;
- Report progress to senior management advisory board;
- Ensure that resources assigned to process improvement tasks are protected;
- Approve the resources for participation on the working groups, making their activities part of their job description.

Establish the Working Group

Identify the members of the working group (WG) per the guidelines provided earlier in the working group description. This implies that all participants are personally committed to the tasks of the WG and that their management agrees to and schedules time to support the working group.

Plan Working Group Activities

Develop a plan and schedule for each WG focus area.

Provide Task-Specific Training

Provide any task-specific training to the WG. Task-specific training is training that is required to put all working group members on equal technical footing. It should help the working group fill in any knowledge gaps that may exist. The training should be an overview of the state of the practice for the particular focus area and what exists in the organization today. Training working group members in configuration management functions would be an example of task-specific training.

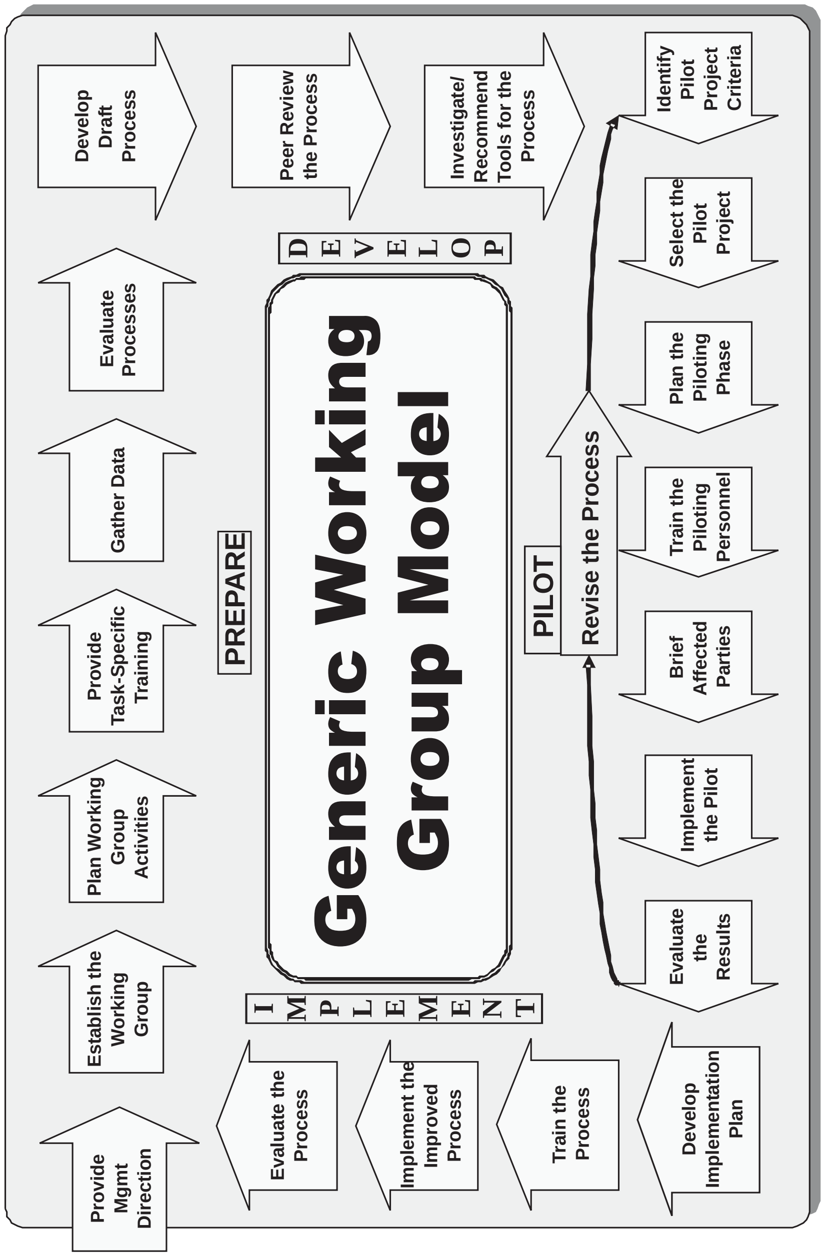


Figure 15.2 Generic working group model.

Gather Data

Gather available data related to the focus area. Gathering data means collecting the existing policies, procedures, processes, reports, and so forth. It includes collecting samples of processes that exist in other business units or organizations as well as state-of-the-practice examples and articles.

- Collect all task-related policy and process documentation.
- Document actual processes in use (internal to the business unit).
- Review any historical data available (internal and external).
- Search for industry data.

Evaluate Processes

Evaluate actual existing and documented processes. Analyzing the data includes finding out why existing processes are or are not used, analyzing any measurement data that may exist, and analyzing what needs to be created or improved.

Develop Draft Processes

Draft processes need to be developed, as follows:

- Determine interfaces among units, departments, phases, and activities.
- Document the process action plans including:
  - Process improvement infrastructure;
  - Process improvement objectives;
  - Improvements that will be covered;
  - Procedures for planning and tracking progress for each process action;
  - Strategies for implementing the process actions including the identification and selection of pilot projects;
  - Training, mentoring, and coaching needed for the pilot project members;
  - Responsibility and authority for implementing the process actions;
  - Resources, schedules, and assignments for implementing the process actions;
  - Evaluation criteria for successful implementation of the process actions;
  - Methods for determining the effectiveness of the process actions;
  - Risks associated with the process action plans;
- Develop guidelines for tailoring to project specific needs.
- Identify the means for measuring the effectiveness of the process actions.

Peer Review the Processes

Submit the process for peer review (all affected parties):

- Revise the process if necessary.
- Submit the revised process to the steering committee for approval.

Investigate and Recommend Tools for the Process

Investigate tools in use or available within the business unit/organization/world and match tool functionality to the process.

Implement Process Actions

Implementing process actions is project management for the process action plans including:

- Negotiating commitments among the process action team members;
- Coordinating with other process action teams;
- Tracking progress on action plan implementation;
- Monitoring progress and results of the process actions;
- Communicating status, activities, plans, and results of the process action plan implementation;
- Comparing the action plan results against the organization's process improvement objectives;
- Using pilot projects to test selected improvements.

Identify Pilot Project Criteria

Establish and maintain pilot project selection criteria. The criteria should include:

- Criticality to the organization (it is all right for the project to be important to the organization's business success);
- Life-cycle phase;
- Willingness of the project manager to support the new and/or revised processes;
- Attitude of the project team toward process improvement;
- Stress on the project;
- Ability of the EPG and quality management group to support the pilot project.

Select the Pilot Project

The use of pilots should be institutionalized within the organization. The use of pilots minimizes the widespread impact to people and projects throughout the organization while the process is exercised and real-world problems are identified and handled. Pilot projects were chosen because they were not in the mainstream of the organization and were thought to be safe. I recommend that a project be identified as a pilot if it is critical to the success of the organization, and intervention along with new or revised processes might make a measurable difference for that project.

Plan the Piloting Phase

Create implementation plans for each pilot project on which the process actions are tested. An implementation plan is necessary to gain the commitment of the pilot project leader and practitioners to the process, to establish the contract for EPG/WG support to the pilot project, and to provide a basis for measuring and verifying the performance of the process.

- Determine how to introduce the new methods/procedures.
- Determine success criteria.
- Establish the contract within the pilot implementation plan.

Train the Piloting Personnel

Train the pilot project members and others who interface with that project in the new process. This step prepares pilot practitioners to perform the activity. Training also involves teaching practitioners how to use any tools built or purchased to support the process.

Brief Affected Parties

Although many groups internal and external to the organization may have requested or even demanded change, these same groups may not react well to pilot projects and they may actually exhibit different behavior. It is often necessary to brief relevant stakeholders or affected parties in order to set their expectations during the time they interact with the pilot project.

Implement the Pilot

Implement the process actions on the designated projects and record measurement data.

Evaluate the Results

Evaluate the results and process performance data based on implementing the new or revised processes on the pilot project(s):

- Collect process performance data.
- Compare the performance data to the established evaluation criteria.
- Get feedback from the pilot project participants.
- Brief all participants on pilot results.

Revise the Processes

Revise processes based on pilot results and then:

- Modify the process action plan if necessary.
- Repilot, reevaluate, revise, and repilot until readiness for full implementation is determined.

Develop a Process Improvement Deployment Plan

Develop the implementation plan for expanding or deploying the process improvement throughout the entire organization. Even when the pilot projects have been successful, a plan still needs to be developed for deploying the improved processes throughout the entire organization. This plan describes when and how the improvement will be deployed across the organization. The deployment approach may have to be done project by project.

Deployment of the standard processes and other organizational process assets must be continually supported within the organization, particularly for new projects at start-up. It is important that new projects use proven and effective processes to perform critical early activities (e.g., project planning, receiving requirements, and obtaining resources). Projects should also periodically update their defined processes to incorporate the latest changes made to the organization's set of standard processes. This periodic updating helps to ensure that all project activities derive the full benefit of what other projects have learned.

Train the Processes

Provide training, mentoring, and appropriate coaching to all of the other projects in the organization in support of full-scale use of the piloted processes.

Implement the Improved Process

Implement into full-scale use in this way:

- Draft and publish a policy for the application of the process actions if none exists.
- Provide general training in each approved process action.
- Ensure all support mechanisms including tools are in place and functioning.
- Conduct quality assurance audits on process compliance.

Evaluate the Processes

Monitor, evaluate, and improve processes. Even if a pilot project achieved success and the organization appeared as if it were accepting of the new processes, progress could slow down, start to regress, or even stop entirely.

It is important to remember that continuous process improvement means looking at existing processes to see if they need further support to keep them living as well as working on getting new ones implemented and institutionalized. Processes will deteriorate without continuing emphasis/support, and processes must evolve as business environments change.

Incorporating Lessons Learned

Of course process improvement is continuous and the intent is not just to get things implemented into full-scale use but to continue to conduct periodic reviews of the effectiveness and suitability of the organization's set of standard processes and related process assets. Reviews provide feedback and let you derive lessons learned from defining, piloting, implementing, and deploying the process assets. An important step in the maturity of any organization is to set up a visible and easy-to-use mechanism for handling process improvement proposals similar to the setup for handling trouble reports. This signals that the process improvement effort is really one of the most important projects in the organization, and ineffective processes are going to be analyzed and improved as if they were a problem report against an operational product.

Communicate Status and Results of Process Improvement Activities

It is important to communicate process improvement success on a regular basis through all multiple media mechanisms available. Publish achievements of culture change and business objectives—let everyone know the successes that have been achieved no matter how small!

Continue to Improve

Process improvement is not always a steadily increasing function. It is important to remember that continuous process improvement means reviewing and adapting process improvement efforts to the changing organization's vision and business objectives, looking at existing processes to see what changes are needed to keep them living, and working on getting new ones implemented and institutionalized to stay aligned with the business objectives.

Process Assets

According to the CMMI®, the purpose of the Organizational Process Definition process area is to establish and maintain a usable set of organizational process assets and work environment standards. An asset is defined to be "an item of value" according to *Merriam-Webster's Collegiate Dictionary*. A process asset, then, is anything of value that helps to implement the practices of a given process area and achieve its goals. Organizational process assets are artifacts that help describe, implement, and improve processes such as:

- Policies;
- Process descriptions;
- Templates;
- Measurements;
- Process implementation support tools.

The term *process assets* also indicates that these artifacts are developed or acquired to meet the business objectives of the organization and are not just developed to satisfy an audit and then become shelfware. They should represent an investment by the organization that is expected to provide current and future business value.

The CMM® for Software referred to the "organization's standard software process" (OSSP). For many organizations, this was interpreted to mean that an organization had to have only one standard process that is applicable to all of the ongoing projects or ones to be implemented throughout the organization. This was never the intent of the authors of the CMM® for Software and certainly would not work in organizations with multiple product lines or multiple systems needs.

The CMMI® indicates that an organization needs to establish and maintain the organization's set of standard processes. The operative word in the CMMI® definition is *set*. The set of standard processes should be able to be tailored for each of the organization's business areas or product lines. The organization's set of standard processes refers to the standard processes established at the organizational level and typically includes technical, management, administrative, support, and organizational processes. Multiple standard processes may be needed to address the needs of different application domains, life-cycle models, methodologies, and tools.

Figure 15.3 provides a graphical view of the components that make up the description of the Organizational Process Definition process area. We examine the components in more detail next.

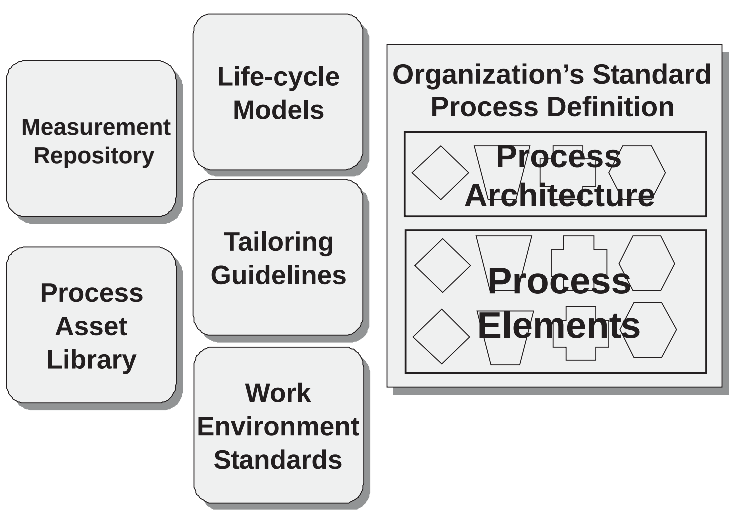


Figure 15.3 Organization's process assets.

Process Asset Library

The organization's process asset library is a collection of items maintained at the organizational level for use by the people and projects within the organization. This collection of items includes descriptions of processes and process elements, descriptions of product life-cycle models, process tailoring guidelines, and process-related documentation and data.

Examples of these process asset library items include:

- Organizational policies;
- Projects' defined standards and procedures;
- Projects' development plans;
- Quality assurance and configuration management plans;
- Projects' measurement plans;
- Projects' process training materials;
- Checklists and templates;
- Lessons learned reports.

Process Elements

A process element is the fundamental (primitive or atomic) unit of process definition. Process elements may be templates, fragments to be completed, abstractions to be refined, or complete descriptions to be tailored or used as is. The elements must be described in sufficient detail such that the process can be consistently performed by appropriately trained and skilled people.

Process elements can be described by critical attributes, which include:

- Process roles;
- Applicable process and product standards;
- Applicable procedures, methods, tools, and resources;
- Process performance objectives;
- Entry criteria;
- Inputs;
- Tasks;
- Product and process measures to be collected and used;
- Verification points (e.g., work product inspections);
- Outputs;
- Interfaces;
- Exit criteria.

Process Architecture

The relationships among the process elements can be described by the *process architecture* (Figure 15.4). Much like the product architecture or software architecture,

Family	Stage
Requirements & Planning	1. Requirements & Planning
Design Implementation	2. Product Level Design 3. Component Level Design 4. Module Level Design 5. Code
Testing	6. Unit Test 7. Functional Verification Test
Packaging & Validation	8. Product Verification Test 9. System Verification Test 10. Package and Release 11. Early Support Program
General Availability	12. General Availability

Figure 15.4 Process architecture stages.

the process architecture refers to the rules for describing those relationships including:

- Ordering of the process elements;
- Interfaces among the process elements;
- Interfaces with external processes;
- Interdependencies among the process elements.

One of the original process architectures was built and described by Ron Radice in the *IBM Systems Journal,* Volume 24, No. 2, November 1985. Its elegance was in its simplicity, which allowed multiple IBM divisions make use of the common architecture and still utilize specialized process elements.

Product Life-Cycle Models

Life-cycle models can be used for a variety of customers and applications because one life cycle may not be appropriate for all projects. Life cycles that are included in the process asset library must be approved for use; approved for use means that the descriptions of these life-cycle models are documented, trained, supported, and maintained. This does not mean that a project cannot choose another life-cycle model if it would be more appropriate to satisfy the project demands. What it does mean is that each project that chooses to use a nonapproved life-cycle model must be assessed for the risk involved due to the lack of organizational support and must be approved by the process improvement steering committee. The following subsections cover multiple examples of life-cycle models [2].

Research, Development, Test, and Evaluation Life-Cycle Model

Research by its nature is proactive. Whether it is independent or directed, research is based on some innovative idea that must be tested and evaluated by the planning and marketing organization. The research, development, test, and evaluation

(RDT&E) life-cycle model provides a framework within which to manage research and development (Figure 15.5). The classic three major phases of definition, development, and deployment have also been expanded to definition, design, development, deployment, and disposal. During the development phase, the constraints of the organizational goals can be measured by the capacity and necessity for corporate change. The prospect of realigning business processes to accommodate new production may provide insights needed to guide future research efforts.

Waterfall Life-Cycle Model

The more or less "pure" waterfall model (Figure 15.6) works well for products with clearly understood requirements that will undergo little change during the project. It works well for the development of products in a familiar domain. The technical tools, architectures, and infrastructures normally should remain stable. The waterfall model is well suited to projects with low risk in the areas of user interfaces and performance, but high risk in budget and schedule predictability. Its weakness surfaces when rapid or quick development is needed.

Overlapping Waterfall Life-Cycle Model

The overlapping waterfall model uses the same phases as the pure waterfall but enables the phases to overlap when needed (Figure 15.7). There is a return path for each step on the model to the previous step to account for any need to change the product of a previous step. It offers an orderly procedure for making changes as far back as necessary to meet the standards of verification and validation. Its weakness comes in various forms often driven by tight schedules and overzealous managers. Activities performed in parallel are subject to miscommunication and incorrect assumptions. Unforeseen interdependencies can create unanticipated problems.

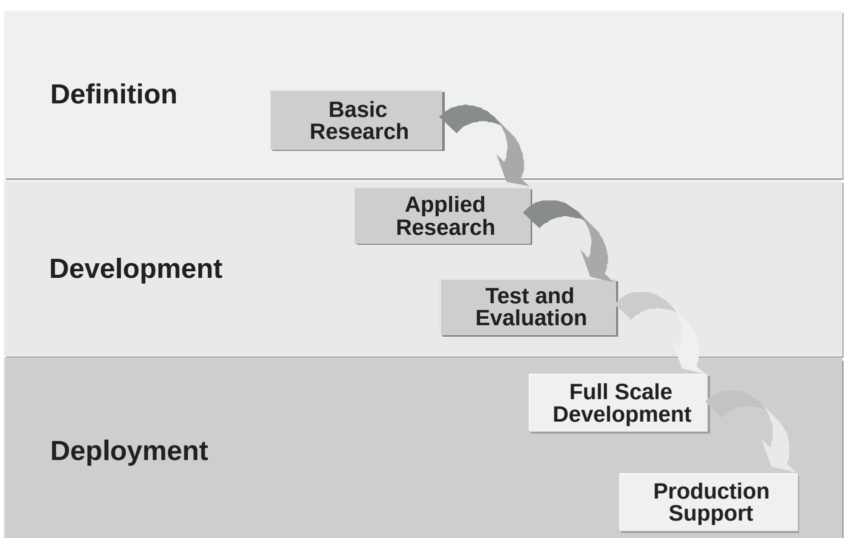


Figure 15.5 Research, development, test, and evaluation life-cycle model.

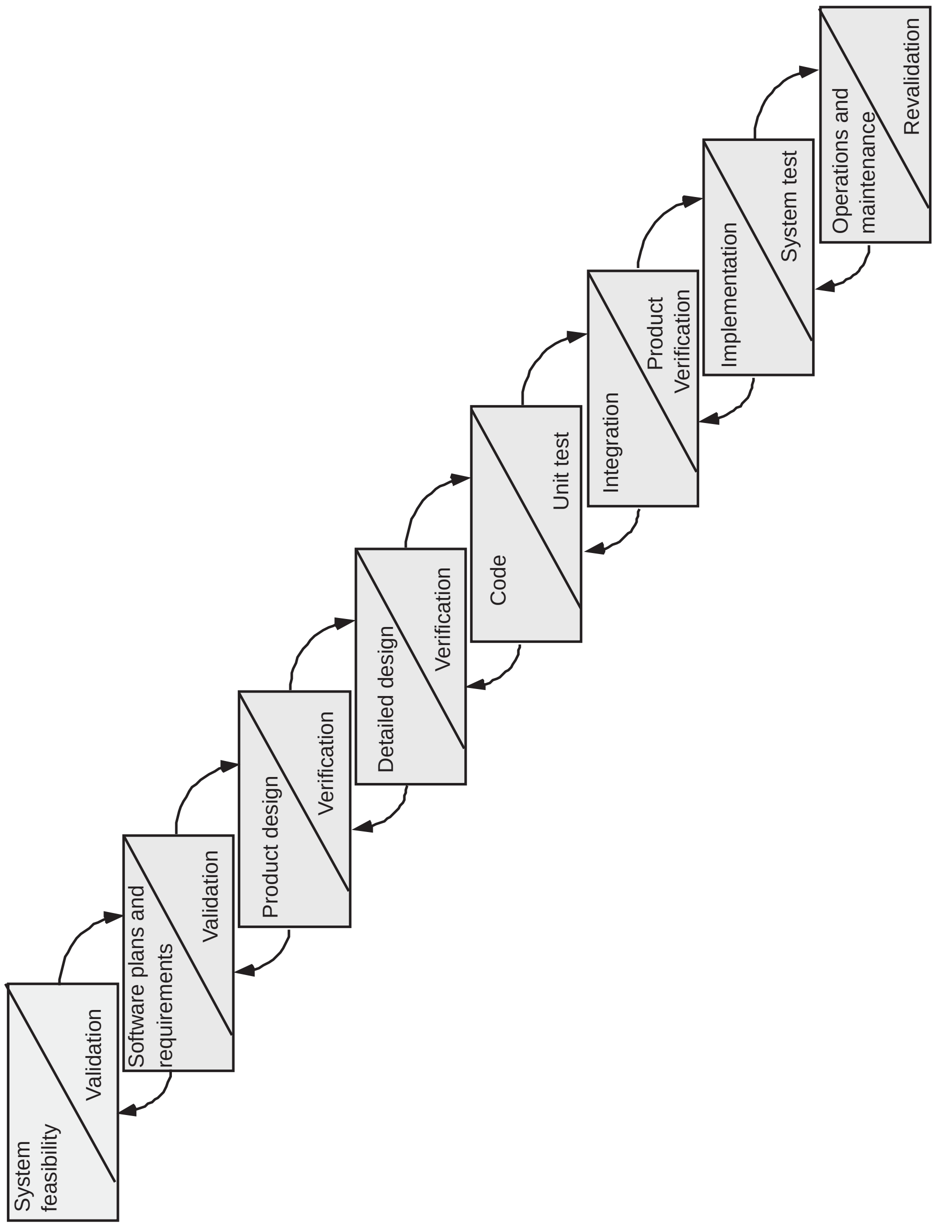


Figure 15.6 Waterfall life-cycle model.

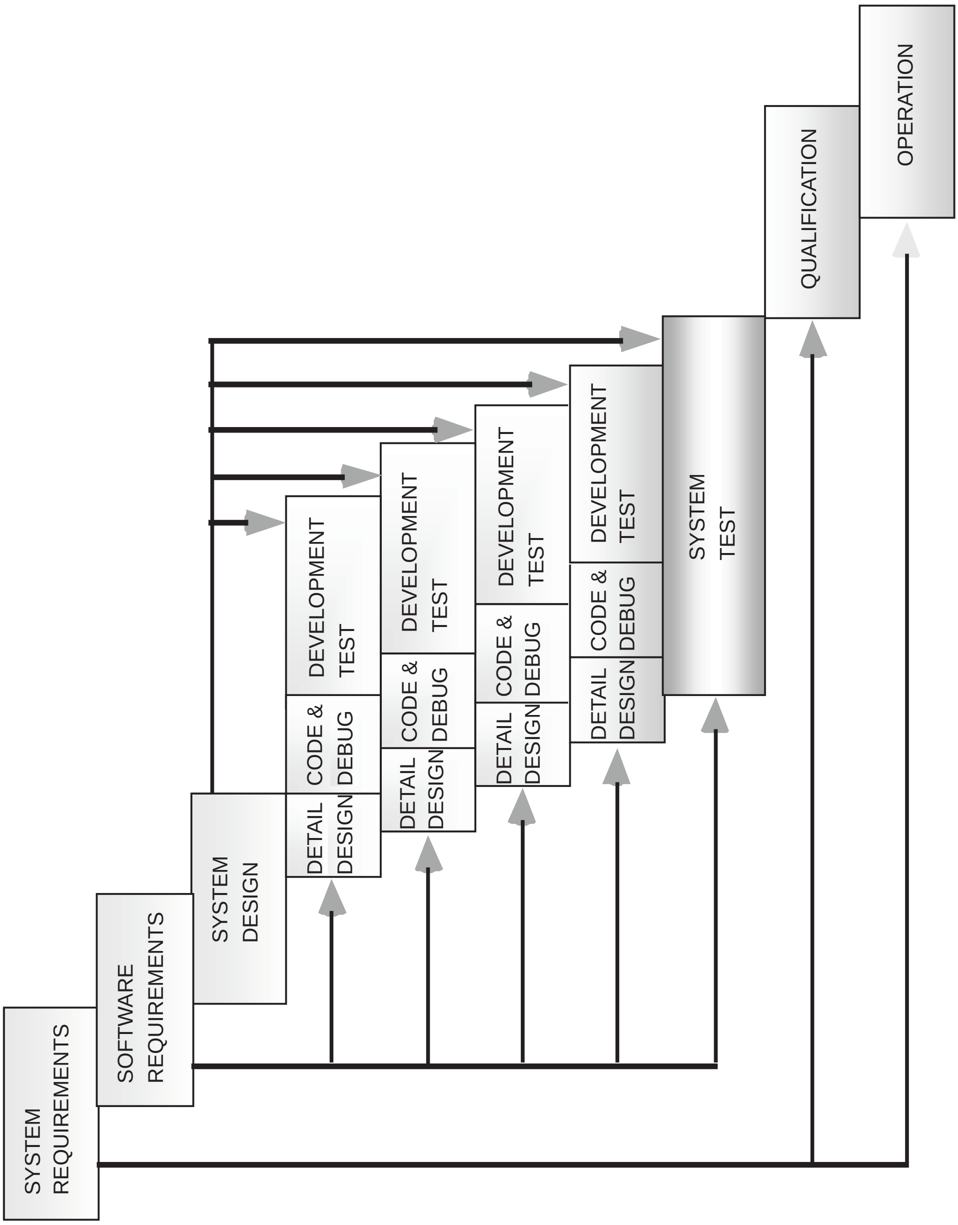


Figure 15.7 Overlapping waterfall life-cycle model.

V-Model Life-Cycle Model

The V-model is an extension of the waterfall model. However, instead of moving down in a linear way, the process steps are bent upward after the coding phase, to form the typical V shape (Figure 15.8). The V-model clearly demonstrates the relationships between each of the phases of development and their associated phases of testing. The V-model is helpful to companies because it can reduce the time required to develop a new product and can also be used to support some complex maintenance projects.

Incremental Life-Cycle Model

The incremental life-cycle model (Figure 15.9) is an evolution of the waterfall model. The product is designed, implemented, integrated, and tested as a series of incremental builds, with each one having incrementally more functionality. Incremental development model may be applicable to projects in which:

- Requirements are well defined, but realization may be delayed;
- The basic system functionality is required early.

Incremental development helps a project manage risk because risky product components are identified and handled during each increment. It is easier to test and debug for the same reason. It is also flexible and can easily handle project scope changes and changes to requirements.

Spiral Life-Cycle Model

The spiral life-cycle model (Figure 15.10) is the most generic of the life-cycle models. The spiral life-cycle model uses a risk management approach to product development. The spiral model can help a project break its effort into miniprojects, each addressing one of more major risks. After the major risks have been addressed, the spiral model flows into a waterfall model. Early iterations of the project are the cheapest, enabling the highest risks to be addressed at the lowest cost. This ensures that as costs increase, risks decrease. The great strength of the spiral model is its capability to develop increments or prototypes, with each full turn of the spiral. The prototype becomes a working core version of the final system.

Rapid Application Development

The rapid application development (RAD) methodology was developed to respond to the need to deliver systems very fast (Figure 15.11). The RAD method has a task list and a work breakdown structure that is designed for speed. However, the major difference in RAD from other life-cycle models is a set of management techniques:

- *Prototyping:* an approach based on creating a demonstrable result as early as possible and refining that result. The refinement is based on feedback from the business, the eventual users of the system.

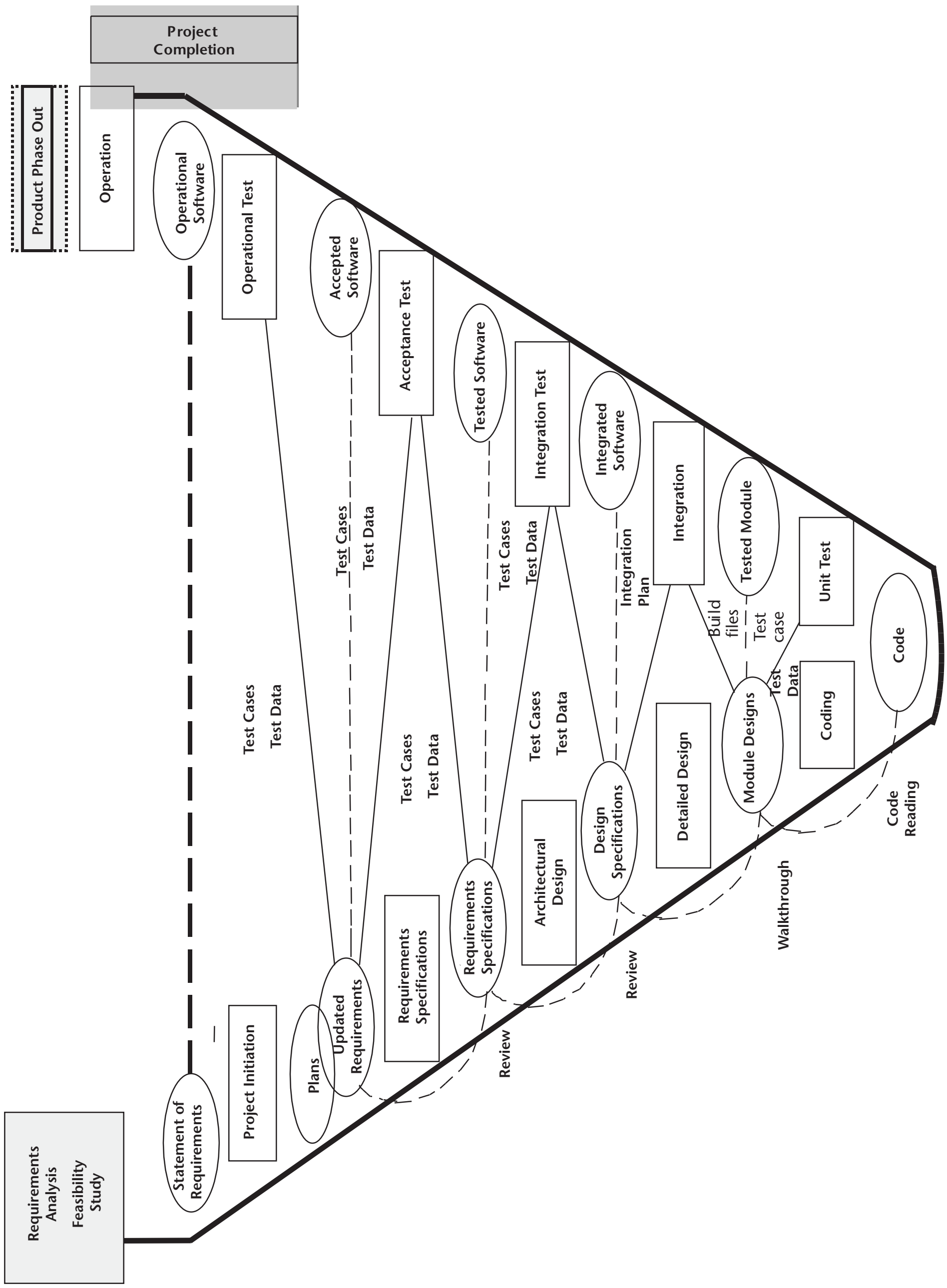


Figure 15.8 V-model life-cycle model.

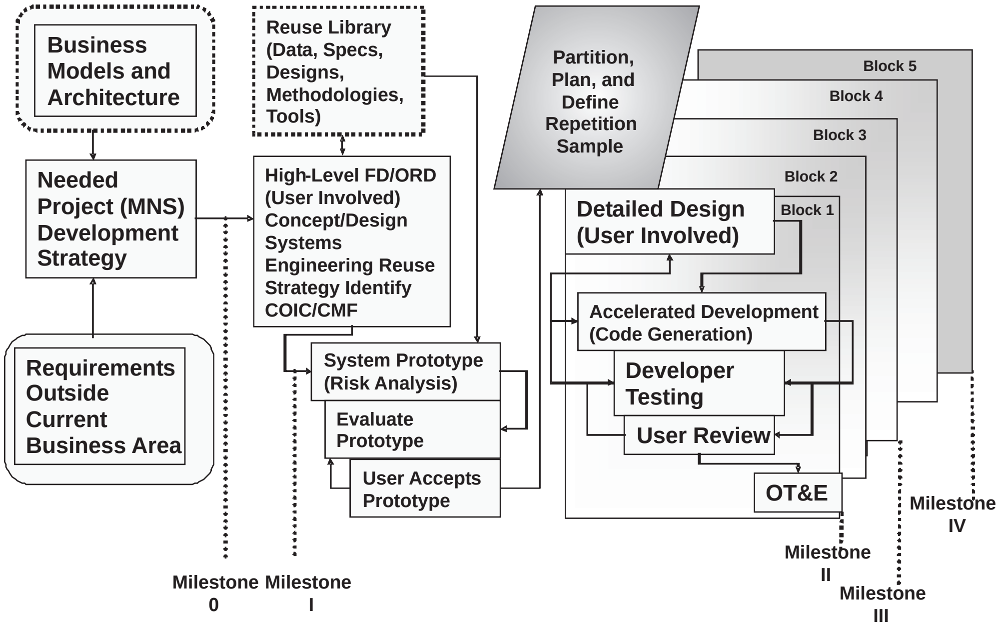


Figure 15.9 Incremental life-cycle model.

- *Iteration:* a commitment to incremental development based on refinement.
- *Timeboxing:* a management technique that focuses attention on delivery above all else. Under a timebox, the scope can change but delivery cannot.

Evolutionary Life-Cycle Model

The evolutionary life-cycle model is an attempt to achieve incremental development of products whose requirements are not known in advance (Figure 15.12). The prototype or increment may be refined and delivered as a product system or it may serve as a de facto specification for new development. In other words, each increment is subject to design, development, test, and delivery as well as elicitation of requirements for the next round. Each incremental delivery may not be acceptable to the customer, however the evolutionary model minimizes the risk of loss.

Work Environment Standards

Parallel to the preceding discussion on life-cycle models, the organization's process assets may include organizational standards for state-of-the-practice tools and methods that have been chosen to be applied across the organization's projects. The CMMI® refers to these tools and methods as *work environment standards*. Work environment standards allow the organization and projects to benefit from common tools, training, and maintenance, as well as cost savings from volume purchases (economy of scale).

Examples of methods in the work environment standards include:

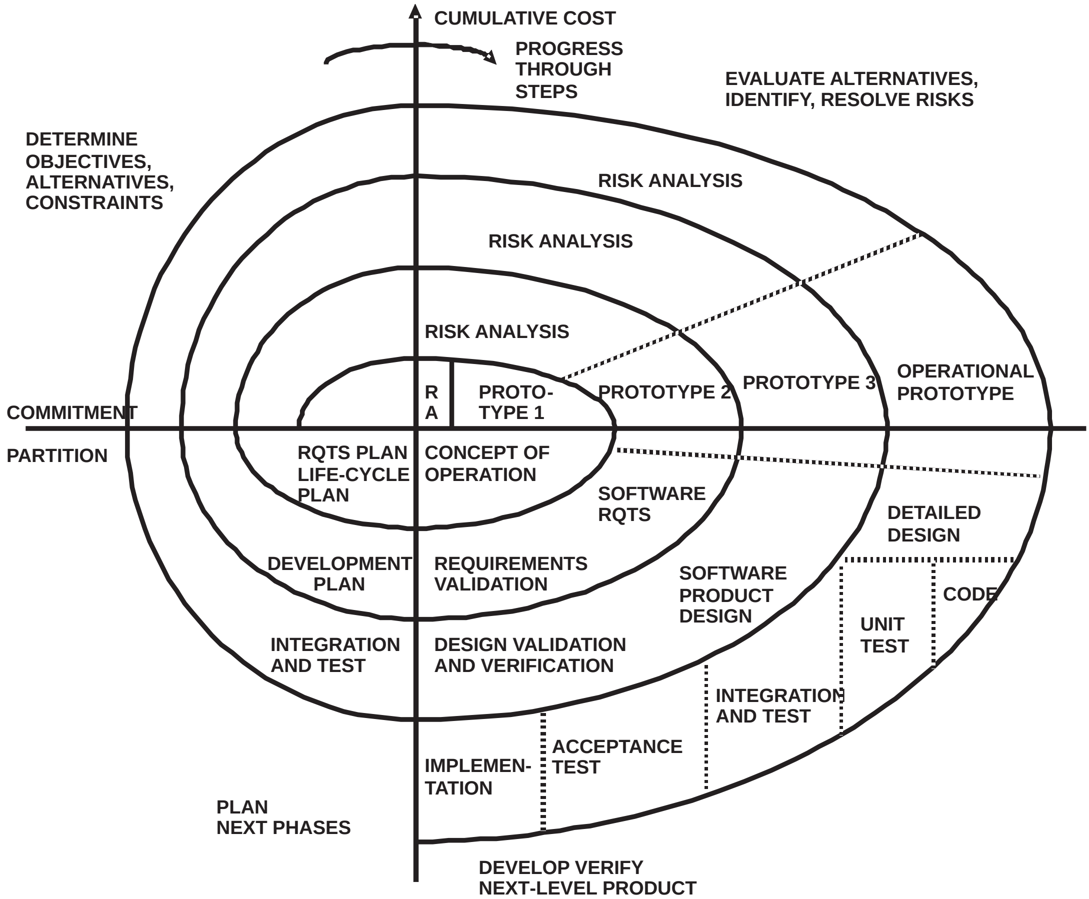


Figure 15.10 Spiral model.

- Object-oriented design;
- Web-enabled development;
- Joint application development (JAD).

Examples of tools that may be included in the work environment standards include:

- QSM (project management estimation tool);
- SERENA® (configuration management tool);
- Telelogic DOORS® (requirements management tool).

Methods should be documented, trained, and supported during real project use and objectively evaluated. Work environment tools should be trained, coached, and supported. Use of new methods and tools on a project should be accompanied by an understanding that productivity will go down before it goes up. It is the training, coaching, and support of the methods and tools on pilot projects that reduce the risk of them being inadequately applied and producing poor results.

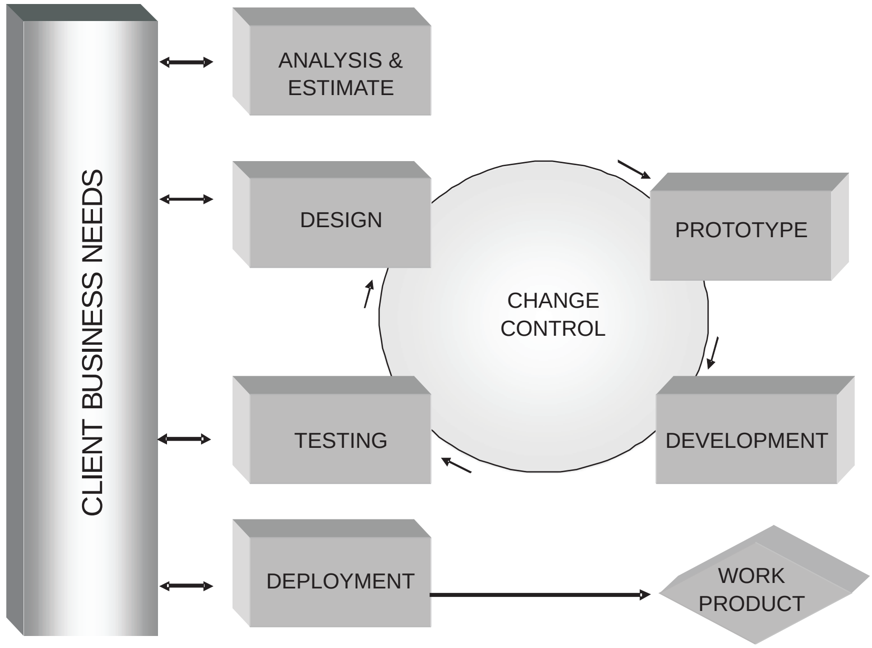


Figure 15.11 Rapid application development life-cycle model.

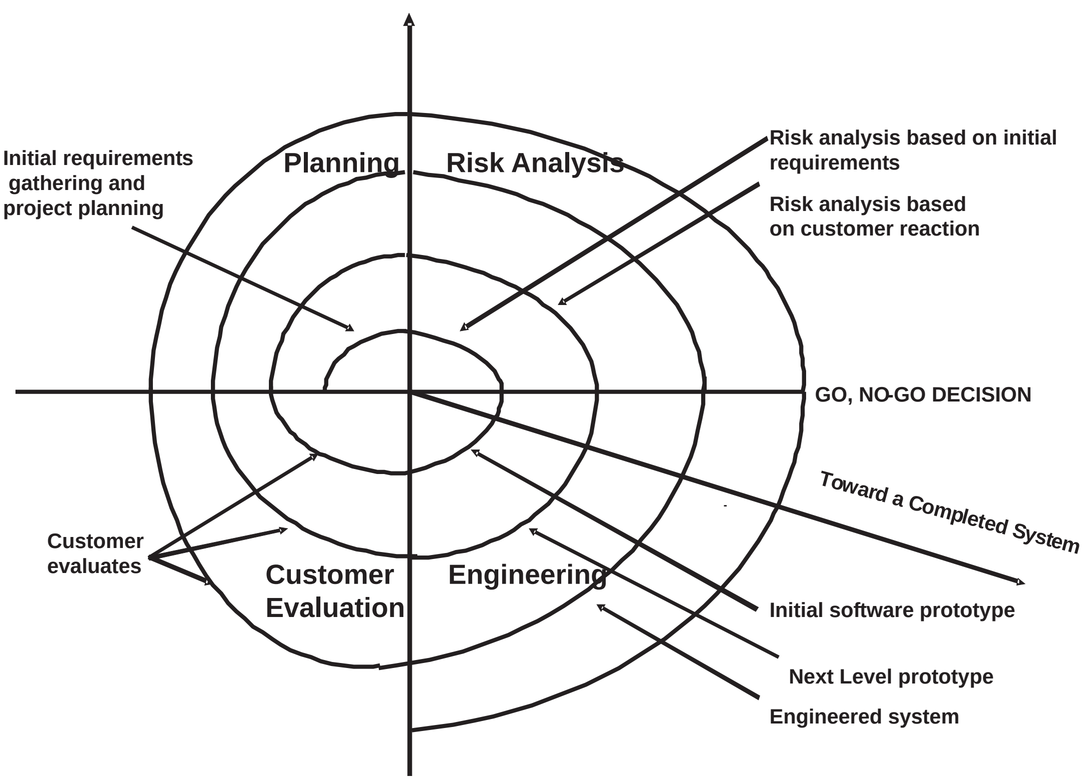


Figure 15.12 Evolutionary life-cycle model.

Figure 15.13 illustrates the relationship of the organization's set of standard processes and the organization's work environment standards to the project's defined process and the project's work environment, respectively.

Tailoring Guidelines

With the variety of projects, application domains, product lines, and constraints placed on projects, it is difficult—if not impossible—to develop a set of standard processes that will satisfy each and every project with no adaptation. It is recommended that each organizational set of standard processes be complemented with an established and maintained set of tailoring guidelines.

Tailoring guidelines should describe the following:

- How the organization's process assets are to be used to create the project's defined processes;
- Mandatory requirements that must be satisfied by the defined processes;
- Options that can be exercised and criteria for selecting among the options;
- Procedures that must be followed in performing process tailoring.

There will always be ongoing debate about the amount of tailoring that should be allowed versus the amount of detailed description that is included in the description of the process elements. These issues must be addressed by tailoring guidelines that are both flexible and consistent. Flexibility in tailoring and defining processes must be balanced with ensuring consistency in the processes across the organization. Flexibility is needed to address the customer, costs, schedule, quality trade-offs, technical difficulty of the work, and the experience of the people implementing the process. Consistency is needed so that organizational standards, objectives, and strategies are appropriately addressed and process data and lessons learned can be shared.

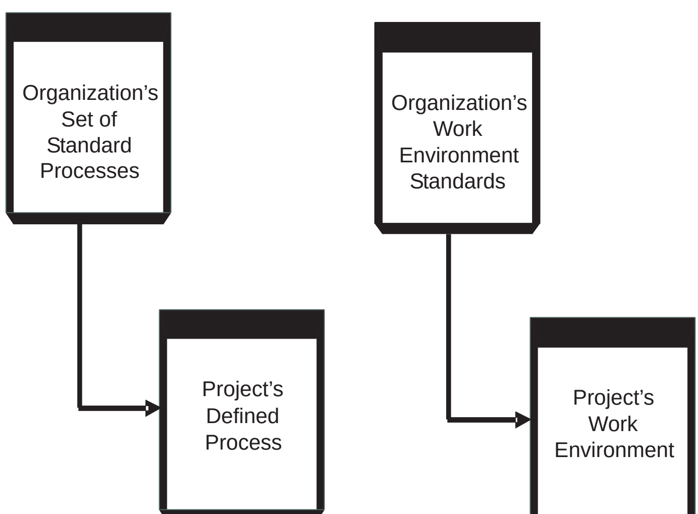


Figure 15.13 Organization's set of standards processes and work environment standards.

Organizational Measurement Repository

The organization's measurement repository contains a common set of measures for both processes and products. The organization's measurement repository should contain:

- Product and process measures that are related to the organization's set of standard processes;
- The related information needed to understand and interpret the measurement data and assess it for reasonableness and applicability.

Operational definitions for the measures must specify the point in the process at which the data will be collected and the procedures for collecting valid data.

Examples of classes of commonly used measures include:

- Size of work products (lines of code, function or feature points, complexity);
- Effort and cost;
- Actual measures of size, effort, and cost;
- Quality measures;
- Work product inspection coverage;
- Test or verification (peer review) coverage;
- Reliability measures.

Summary

Process improvement requires the support of all levels of management and the practitioners. Senior managers, middle managers, project managers, and practitioners must be committed and visibly participating to make process improvement work. The EPG is a focal group for action planning, process improvement, technology insertion, training, and awareness and expectation setting. EPGs are the organization's "champion of change" and its members "change agents." An EPG must facilitate the process improvement efforts at the organization, project, and individual level. Each organization must establish and maintain a usable set of organizational process assets to help implement the practices of a given process area and achieve its goals. These organizational process assets or artifacts are developed or acquired to meet the business objectives of the organization.

References

[1] *Software Engineering Process Group Guide*, CMU/SEI-90-TR-24, Pittsburgh, PA: Software Engineering Institute, September 1990.

[2] Wikipedia, Definitions, February 2008: V-Model, http://en.wikipedia.org/wiki/V_model; Incremental Life-Cycle Model—SoftDevTeam, http://en.wikipedia.org/wiki/Iterative_and_incremental_development; Waterfall Model—Level A Software, http://en.wikipedia.org/wiki/Waterfall_model; Spiral Model—Level A Software, http://en.wikipedia.org/wiki/Spiral_model; Incremental Model—Level A Software, http://en.wikipedia.org/wiki/Spiral_model; Project Lifecycle Models—Business eSolutions, http://en.wikipedia.org/wiki/Product_life_cycle_management.

PART V

Individual Knowledge and Skills to Integrated Teams

CHAPTER 16

The Knowledge and Skills Base

As stated in the CMMI®'s Organizational Training process area, the real need that every organization must address is the need to continue to develop the knowledge and skills of its people so that they can continue to perform their roles effectively and efficiently. This includes all levels of management and practitioners. It includes developers of all engineering disciplines and support groups. It even includes departments such as human resources, finance, marketing, and procurement.

Knowledge and skills may be technical, organizational, or contextual:

- Technical skills affect the individual's ability to use the equipment, tools, materials, data, and processes required by a project or a process.
- Organizational skills focus on behavior within and according to the employee's organization structure, role and responsibilities, and general operating principles and methods.
- Contextual skills are the self-management, communication, and interpersonal abilities needed to successfully perform in the organizational and social context of the project and support groups.

The identification of *process training needs* is primarily based on the skills that are required to perform the organization's set of standard processes. These are, in turn, based on business objectives and core competencies.

Organizational Training Focus

Organizational training programs must support the organization's strategic business objectives and tactical training needs that are common across projects and support groups.

Core Competencies

The focus on organizational training programs and their link to building core competencies should not be taken lightly. Given the increasingly complex software and hardware systems being built today, finding and keeping good people is also becoming increasingly difficult. Two managers have been quoted as saying: "The most important ingredient on this successful project was having very smart people" and "The only rule I have in management is to ensure that I have good people—real good people—and that I grow good people, and that I provide an environment where good people are happy and can produce."

As previously stated, however, finding and developing these good people can and will be a difficult task. Therefore, organizations need to take on a more aggressive and "people-focused" approach to providing training for their workforces. First and foremost, it is imperative that the senior management team not only be able to state current business objectives, but also be able to articulate what business the organization is in. This is not as trivial as it may sound. One definition of the organization's business may result in the workforce becoming frustrated with the management team and may not result in the workforce understanding what is expected of them. Another definition of the organization's business may make it clear for all employees such that some will leave the organization immediately, but others will increase their efforts to gain the necessary core competencies and become a more valuable asset to their organization.

Once it is clearly defined what business the organization is in, the organization must identify and define the *core competencies* required to perform the organization's business and still remain competitive. The management team and the practitioners must understand just what these core competencies are and why they are so critical. All efforts to increase the core competencies of the workforce must be totally supported by management and practitioners alike. Departments that are responsible for training and career development, such as human resources, must work closely with project managers, lead engineers, quality managers, and process improvement specialists to find ways to develop and/or acquire the core competencies that support the business.

It is appropriate at this point to bring forth some of the significant points from a classic article that appeared in the *Harvard Business Review* in 1990 [1] titled "The Core Competence of the Corporation." The most powerful way to prevail in global competition is to identify, cultivate, and exploit the core competencies that make growth possible. In the long run, competitiveness is derived from an ability to build, at lower cost and faster than competitors, the core competencies that spawn unanticipated products. A focus on the core competencies empowers individual businesses to adapt quickly to changing opportunities.

Core competence is communication, involvement, and a deep commitment to working across organizational boundaries. Core competence should not diminish with use. But competencies still need to be nurtured and protected. Knowledge fades if it is not used. When competencies become imprisoned or are outsourced, the people who carry the competencies do not get assigned to the most exciting opportunities and their skills soon diminish. If core competencies are not recognized, individual business units will only go after those opportunities that are close at hand. Competencies are the glue that binds existing business. They are also the engine for new business development.

At least three tests can be applied to identify core competencies in a company:

1. A core competence provides potential access to a wide variety of markets.
2. A core competence should make a significant contribution to the perceived customer benefits of the end product.
3. A core competence should be difficult for competitors to imitate.

Core competencies are built through a process of continuous improvement and enhancement that may last a decade or more. Focusing on core competencies creates unique, integrated systems that reinforce fit among your company's diverse production and technology skills—a systemic advantage that cannot be copied by your competitors.

Following the definition of the core competencies, it is now appropriate and necessary for the organizational training focus to determine just what makes up the organizational knowledge and skills base. Personnel records and training records are inputs for this activity. Strategic and near-term plans must be developed to build up the necessary knowledge and skills and thus the core competencies. Career development must be planned to support the core competency development for each individual; in doing so, core competencies are also developed for the organization. Workforce practices must be adapted to reward the knowledge and skills growth that leads to higher core competency.

Recruiters must also change their focus, looking first for candidates who either have the necessary knowledge and skills that fulfill the core competency needs, or to a base of knowledge and skills combined with a proven ability to learn that could be used to develop the critical core competencies for the organization.

Organizational and Project-Level Training

The organization's strategic business objectives and improvement plans should be analyzed in order to plan for current, intermediate, and future training needs in order for the organization to remain competitive. Examples of sources of strategic training needs include:

- The organization's set of standard processes;
- The organization's strategic business plan;
- The organization's process improvement plan;
- Enterprise-level initiatives;
- Skill appraisals;
- Risk analyses.

Once the strategic training needs are captured, it must be determined how this will affect the training needs of the organization versus project training needs. Steps to accomplish this include the following:

- Identify the training needs required at the organizational level.
- Analyze the project and support group's needs that can be most efficiently addressed organization-wide.
- Negotiate specific training needs with the various projects and support groups.
- Consider "economies of scale" at every opportunity when planning for organizational versus project-level training. Economy of scale refers to providing organizational training if the training need is shared by multiple projects.

Training Capability

To establish the required organizational training capability, three areas must be defined:

1. The instructor's or trainer's proficiency;
2. The training approach;
3. Descriptions of training (syllabuses).

All instructors must be required to demonstrate current proficiency in the topics they have been identified with and intend to teach. There are many learning approaches. It is appropriate, therefore, for different training approaches to be developed and used, some in combination with others, to teach or train as well. Some training approaches that can be effectively used by an organization include:

- Classroom teaching;
- Computer-aided instruction;
- Guided self-study;
- Formal apprenticeship and a mentoring program;
- Facilitated videos;
- Structured on-the-job training.

It is equally important for detailed syllabuses to be provided that describe training opportunities so that the organization's workforce members will know what is available. This will enable individuals to select appropriate courses and participate in other training opportunities with the assurance that these opportunities will help them gain the desired knowledge and skills. Such a detailed outline/syllabus should contain:

- Course abstract;
- Objectives;
- Intended audience;
- Length of the training;
- Prerequisites and preparation for participation in the training;
- Format of the training;
- Training topics;
- Criteria for determining the participant's satisfactory completion.

Training Delivery

Increasing the knowledge and skills of a target group of participants is so much more than just assigning an instructor and then telling him/her to deliver the training material. Indeed, my experience in providing quality management and process improvement training during the past 20 years points to ensuring that guidelines and expectations are clearly set, and then reset again, as appropriate. Participants should also be carefully selected through documented procedures and checklists, rather than by just determining who needs training hours logged and who is available to participate.

A starter kit list of participant selection guidelines is shown here:

- Do they have the prerequisite background?
- Do they have the skills and abilities to perform their roles?
- Do they have the desire to use the information provided in the course?
- Will they have the opportunity to put in place the training learned or to at least influence process improvement?
- Is there a need for cross-discipline technical and management training?
- Do the managers need training for their level as well?
- Is there a need for training in basic engineering principles?
- Is there a need for training in the project support functions such as quality assurance or configuration management?
- Is there a core competency that is needed immediately?

In addition to properly screening participants, it is important to plan for each training session, including the time each participant must have to take advantage of the learning opportunity. Nonstop interrupts, daily crises to be taken care of, and electronic leashes (mobile phones) are not conducive to learning. It is also critical to choose experienced instructors that can not only present the material, but can also speak from their experiences in order to enhance the value of the material.

Effectiveness of the Training

Most organizations hand out a participant evaluation form. They ask that it be filled out at the end of the course in order to determine the effectiveness of the course and of the instructor. Personal experience has shown that this offers little insight into the true effectiveness of the training or, even more importantly, the knowledge and skills gained, which was the objective of the training. Following the lead and examples of Tom Gilb [2] in his book on software management, a quantitatively measurable "effectiveness of training" definition is offered here:

- *Experience and knowledge of the participants prior to the training.* What background do the participants really have before they go to the training? Having little to no background in the training topic does not mean the training will not be valuable, but if the course is oriented toward those who already have a strong foundation in the topic, beginners will not get much value and those who seek advanced concepts will be held back and become frustrated with the beginners.
- *Preparation by the participants prior to the training.* Often courses suggest or require that the participants read some material for background to establish a more equal playing field. However, it is my experience that little preparation is done by the participants or, in some cases, not allowed by their manage-

ment. This automatically forces the instructor to calibrate and change the emphasis for the first day if not the entire training class.

- *Expectations of the participants (overview course or in-depth study of a subject compared to the course objectives).* If the participants are attending a 1-day seminar that the instructor has cut down from a 3- to 4-day workshop and are expecting the same value, the disappointment will be great on both sides. I actually had the opportunity to teach 3-day workshops on quality management topics to a large company with very good results. Suddenly, the company stated they could no longer afford to send their employees for a full 3 days and asked if a 1-day seminar was possible. I said yes but clearly stated that the participants would get only an overview of the topics that were normally covered in the 3-day class. The training manager said that was all right. But this was not conveyed to the participants. The result was a classroom evaluation rating that was the lowest of my career and the near loss of a significant customer. Eventually it was resolved, but not without a lot of negotiation. We also had to reach an understanding about the expectations accepted by the customer.
- *Training materials.* The training materials contribute toward the overall learning of the participants. Poorly organized training materials or poor-quality training materials can quickly result in participant dissatisfaction.
- *Experience and knowledge of the instructor.* Without a doubt, the experience and knowledge of the instructor on the topics being presented are important. A weak instructor makes the class uncomfortable for everyone.
- *Ability of the instructor to train adults.* This point deserves more elaboration! Adults generally have experience and somewhere during the class will want to verbally exhibit that experience and get recognition for it. Instructor's who feel that all participants should be like young students at university or high school will eventually insult these adults and turn off their interest in learning anything from the class. Most of the time, I have found that not being willing to acknowledge the experience and knowledge of the adult participants results in repeated interruptions and sometimes class-stopping arguments between the participant who wants recognition and the instructor.
- *Ability of the instructor to add value to the materials from his/her experience.* Many people feel they need only to "read" the slide material verbatim and tell a joke or two and they will have fulfilled their obligation as an instructor. My experience is that the ability and willingness of the instructor to add value by sharing his/her experiences through stories and examples keeps the participants interested and gets the best results.
- *Time lapse after the training before the participant has an opportunity to put the training concepts to use.* Training that will not be used for months following the training session may end up being wasted for the participant who attend the training. Sometimes the timeliness of training is not always under the control of the participants or their managers, but the risk of the training not being effective and useful should be understood up-front. Refresher training may even be required.

- *Mentoring and coaching available to participants after the training.* Whether this is external or internal is not as relevant as whether or not the participants have access to coaching and mentoring. I have often been told, and rightfully so, that the course material and examples I provided were good but the participants needed help in figuring out how to apply the concepts learned during the training to their daily work. Management should be prepared to provide the necessary mentoring and coaching, in a timely manner, for participants following the training, especially if the training is critical to the project at hand and the core competencies of the organization.
- *The management support the participant has after the training for attempting to use the new concepts he/she learned.* This point is near and dear to my heart. I have often been told that the training was great, but the participants' management was not going to allow the participants to use the concepts following the training. Of course, my brain starts to scream "Then why send them to the class in the first place?"
- *Was the training on a subject that was part of the participant's job or not?* This is another point that makes a significant impact to the perceived effectiveness of the training. I have taught a class where 9 of the 10 participants were not interested in attending the class because it was not part of their job description, was never going to be part of their job description, and they were not interested in attending an overview to complement their current job assignment. One more time, the question must be asked: "Then why send them to the class in the first place?"

Training, Mentoring, and Coaching

The gaining of knowledge and skills leading to increased core competencies requires an approach that is focused on successful technology transfer. This may be accomplished by these means:

- Provide *training* in order to convey technical and organizational change concepts to individuals and groups who need to have an in-depth knowledge of the topics. Training should not be the only mechanism to transfer years of experience to the participants.
- Provide *mentoring* in order to share with a select group of individuals the psychology and philosophy behind the concepts of training or of processes, procedures, guidelines, templates, and so forth. Mentoring sessions should be conducted with an expert or a senior experienced person, who may even be approaching retirement, and the person being mentored. Experiences and war stories need to be shared in order to convey a sense of reality and understanding to those who are being mentored. Lessons learned over the years can be passed along to the next generation.
- For many companies, training is really reduced to *on-the-job-training*. This usually translates into "trial by fire." This form of training should be avoided. Its effectiveness has been proven extremely low over the decades.

- Provide coaching together with on-the-job training. This has been proven to be a very effective combination and is highly recommended. *Coaching* of individuals and small groups while they are working on the project should be provided so that they can see the practicality of the ideas. Organizational experts may provide such coaching in one of three consulting modes:
  - *Expert*. The consultant leads the development of a plan or leads the peer review or other artifact with minimal input from the project manager or members.
  - *Collaborative*. The consultant owns 50% of the problem to be solved and the client owns 50% of the problem to be solved. This assumes a certain level of knowledge on the part of the client's personnel.
  - *Observer*. The consultant basically reviews work that has been done and provides direction and/or guidance on implementation.

Training, mentoring, and coaching may be the only way available for many organizations and even countries to remain competitive or remain in business. I have had the privilege of training, assessing, and consulting in more than 25 countries worldwide. The attitude toward training, mentoring, and coaching the next generation varies greatly. One country ensures that its senior managers and engineers spend up to 50% of their time mentoring and coaching the next generation so that their experience will not be lost when they retire. This same country supports its highly motivated individuals by giving them a sabbatical to work toward a master's or doctor's degree in other countries to increase their knowledge base with different thinking and help keep innovation alive. A second country not only does not do what I have described for the first country, but they do not allow their senior managers and engineers to spend any time mentoring and coaching the next generation. What type of company do you work in? How will your lessons learned over the years get passed on to the next generation?

When an organization is focused on building core competencies and is willing to provide the necessary training, coaching, and mentoring it takes to help its workforce gain the knowledge and skills that will lead to the core competencies and successful business, we also say that this organization has evolved to the point where they view their workforce as critical corporate assets. The last section of this chapter offers some questions that are designed to help each person determine if they feel his/her organization considers them to be critical corporate assets or not.

Treating the Workforce as Critical Corporate Assets

If organizations are to succeed in the face of increasingly complex tasks that demand large amounts of "good people" they must view their people as critical corporate assets. The following set of questions should be used in an organizational survey to determine if personnel believes they are viewed as critical corporate assets or not.

To help determine if you are considered a corporate asset in your own organization, answer each of the following questions:

1. What is your educational background?
2. What job experience do you have that helps you do your current job?
3. Do you know what the organization's business is and what core competencies are required to support that business?
4. Do you have the skills you need to do your job?
5. What training have you received in the past year related to your current job?
6. What mentoring have you received in the past year related to your current job?
7. What coaching have you received in the past year related to your current job?
8. Is your career development path defined and in line with the organization's competency needs?
9. Are you able to conduct self-study?
10. Are you willing to spend the time and the effort necessary to make a real difference in your life? Or do you sit and wait for someone else to tell you it is time to make an improvement? Be aware that 99% of success is based on desire and the willingness to take charge of your own life. After all, you are your only commodity. That is the bottom line.
11. How do you respond to classroom training?
12. How well do you absorb training on the job?
13. What training related to the management of people has your manager or supervisor had in the past year?
14. How would you rate your manager's people skills? (circle one)
    - a. Doesn't care about people.
    - b. Shows some care, but not much and not often.
    - c. Tries to incorporate concern for people in everyday management.
    - d. Balances focus on people and technical tasks.
    - e. Is a real people person.
15. Do you feel you have an equal chance to get promotions in your application areas?
16. Are there opportunities for you to test your skills in other areas?
17. Do you feel you are adequately compensated at present?
18. Do you think your company's overall compensation plan is acceptable? Why or why not?
19. Do you think your actions are aligned with the direction your organization is going?
20. Do you believe that others in the organization are aligned with the direction your organization is going?
21. Do you feel like you have grown technically and emotionally since you have joined the organization?
22. How long have you been with this organization?
23. How long were you with the organization in your previous job?
24. What motivated you to stay or leave?
25. Are you planning on a long-term relationship?

26. Do most people and most projects in the organization have the capability to deliver high-quality products?
27. What motivates you to enjoy working for a company and stay with that company?
28. Is the company prepared to hire talented people as well as help them grow internally?
29. Is your company prepared or preparing for future competition?
30. Are people in your organization considered to be "human capital" and treated as corporate assets?
31. Finally, are you ready to take charge of your life? Are you willing to step forward and say "I am ready for that training, I want it, and I will do what it takes to use it successfully!"?

The bottom line is that people count. Too often people are viewed as commodities. They are, in fact, the most important assets of a corporation. Their knowledge, their ability to grow, and their belief in the honesty and integrity of the organization are invaluable. Getting the maximum output from employees means investing the maximum in their potential and affording them opportunities for growth.

Summary

The most powerful way to prevail in global competition is to identify, cultivate, and exploit the core competencies that make growth possible. This happens with vision, a clear set of business and measurement objectives, and a continuous focus on the knowledge and skills of an organization's workforce. To accomplish this, organizations must minimally do the following:

- Clearly understand, define, and communicate what business the organization is in.
- Identify and define what core competencies are needed to support the organization's business.
- Identify the knowledge and skills necessary to be considered competent in the core competencies.
- Determine the knowledge and skill levels of the people for each department throughout the organization.
- Set up training, mentoring, and coaching programs for the employees in the organization's core competencies.
- Assist employees in their career development planning in order to enhance their capability to perform their assigned tasks and responsibilities along with the identified core competencies.
- Hire new employees based on how well their background matches the core competencies and how well their knowledge and skills complement the knowledge and skills base that already exists in the organization or project.

- Provide incentives in the form of promotions, money, time off, and so forth, based on an employee's ability to grow in the organization's core competencies while producing high-quality processes and products.
- Coordinate all of the workforce activities with the current and future business needs.
- Align the motivation and growth of the people with that of the organization.
- Treat people like critical corporate assets.

References

[1] Prahalad, C. K., and G. Hamel, "The Core Competence of the Corporation," *Harvard Business Review,* "On Point Article," 1990.

[2] Gilb, T., *Principles of Software Engineering Management*, Reading, MA: Addison-Wesley Longman, 1988.

CHAPTER 17

Integrated Teams

An integrated team is composed of members with complementary skills and expertise who are collectively responsible for delivering the work product. Team members include empowered representatives from both the organization's technical disciplines and business functions involved with the product. The team members have a stake in the success or failure of the work products produced. Within defined boundaries, these representatives have decision-making authority and the responsibility to act for their representative organizations. These integrated teams may be viewed as a microversion of the company or business unit itself.

This chapter describes the conditions under which integrated teams are considered, built, and managed. It includes the CMMI® process areas of Organizational Process Definition and Integrated Project Management.

The Concept of the Integrated Team

> Establishing self-managed and empowered teams, whose members are collectively responsible for delivering the work product in order to support clear business objectives, could be equivalent to achieving CMMI® Maturity Level 3.5.

Integrated product teams are only established to satisfy specific business objectives that the project manager or higher level managers of the organization believe cannot be reached without the skills and abilities of a special group of people. These special people are to be provided with necessary tools, equipment, training, and so on, that may not normally be provided for standard project members. They should be highly skilled in their own disciplines or functions and have demonstrated the ability to learn. They should also have demonstrated the ability to get along with and work cooperatively with others. These integrated team members do not have to be solicited to share their skills. They will do it whenever the integrated team needs them too.

The concept of an integrated team is not the same as a project team as normally defined at CMMI® Maturity Level 2 or even integrated project management at CMMI® Maturity Level 3. It does not concentrate on team-building skills, although those skills may certainly be taught, and it is not related to the team software process although, again, those skills may be taught as part of the integrated team scope of things to do. An integrated team may consist of the entire project but more often it is a subset of the project. Integrated teams are more like highly trained special forces units that are brought together at considerable expense and asked to perform specific tasks with significantly higher expected results. To decide to put such an integrated team together requires understanding the business objectives and determining if the effort will be cost effective.

We now examine some of the requirements brought out in the CMMI® with respect to integrated teams.

Shared Vision

The most important characteristic of having successful integrated product teams is to establish and maintain the organization's shared vision. A *shared vision* is a common understanding of guiding principles including mission, objectives, expected behavior, values, and final outcomes that are developed and used by the organization, project, or integrated team.

The purpose of creating a shared vision at any level in the organization is to achieve a unity of purpose. The value of a shared vision is that people understand and can adopt its principles to guide their actions and decisions. Shared visions help team members to focus on the end state while still allowing room for personal and team innovation, creativity, and enthusiasm.

The shared vision of a project's integrated team should be consistent with the shared vision of the project, which in turn must be consistent with the shared vision of the organization as illustrated in Figure 17.1. It is important that the individual's vision be aligned as well.

Organizational Process Definition

Along with the shared vision, successful integrated teams will have organizational environments established for them to promote the highest possible productivity, quality products, and services. The organization's set of standard processes and organizational process assets need to be augmented to support the integrated teaming concepts.

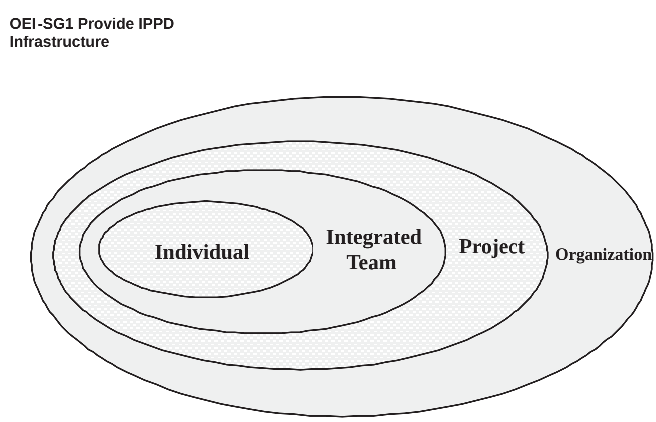


Figure 17.1 Shared vision context.

Integrated Work Environment

In addition, an integrated work environment must be established and maintained according to the needs of the integrated team. The integrated work environment includes the physical infrastructure such as facilities, tools, and equipment to allow the integrated team members to perform their jobs effectively.

Integrated work environments are and should be viewed as capital assets that:

- Are often expensive;
- Have unique implementations;
- Are irreversible (their implementation can destroy or make unusable the assets being replaced);
- When modified can disrupt ongoing activities.

The integrated work environment comes from the organizational work environment standards that are tailored for the integrated team normally sponsored and managed by the project (Figure 17.2). *Integrated work environments must be evaluated to determine if the performance improvements are worth the costs and risks;* hence the earlier reference to the business objectives.

Examples of integrated work environment technologies, tools, and resources include:

- Computing resources;
- Software productivity tools;
- Communication systems, tools and resources;
- Engineering or simulation tools;
- Proprietary engineering tools;
- Prototyping or production equipment.

Examples of communication tools include meeting rooms, e-mail, Web sites, and videoconferencing capabilities.

Integrative Leadership and Interpersonal Skills

The management of integrated teams demands a different view of product development and requires integrative leadership and interpersonal skills beyond those typically found in traditional environments where people tend to work alone. Some of those skills were mentioned earlier. Others include:

- The skills to integrate all appropriate business and technical functions;
- The interpersonal skills to coordinate and collaborate with others;
- The leadership skills to act, influence others, and achieve the shared vision.

Higher leadership skills for those who will lead an integrated team are also required. These include:

- Ensuring that all team members mutually understand their roles and responsibilities;
- Actually using the people in their intended roles;
- Effectively accessing the expertise that exists in the organization and integrating it to strengthen the team effort.

Implementing integrated teams requires cultural changes as people and integrated teams are empowered and decisions are driven to the lowest level as appropriate. However, empowerment does not necessarily mean that every decision must occur at the lowest level—decision making is influenced by the decision type that the team agrees will be used to resolve issues and that the project manger and higher level managers have agreed to support. Examples of decision types include:

- *Command*. The leader examines the issue and makes the decision alone.
- *Consultative*. The leader receives and examines inputs on the issue from relevant stakeholders and then makes the decision.
- *Collaborative*. Issues are raised, discussed, and voted on. Rules are established to determine when this vote is binding on the leader.
- *Consensus*. Issues are discussed among all members of the integrated team until the entire team agrees that it can live with and support the decision.

While integrated teaming normally corresponds to team-related incentives, individual excellence must also be valued. To purposefully support the use of integrated teams, the recognition and reward systems need to shift from a focus on the success or failure of an individual (program manager) to an integrated team success or failure. Care should still be taken to continue to recognize individual excellence as long as it is not achieved at the expense of the established integrated team behaviors.

Integrated Project Management: IPPD Addition

As indicated in Figure 17.1, a project does not operate in isolation. If a project understands organizational expectations and constraints, it can align its direction, activities, and shared vision with the same in the organization, and it can help create a common purpose within which its activities can be coordinated. A project's shared vision context has both an external and internal aspect:

- The external aspect deals with the interfaces outside of the project.
- The internal aspect deals with aligning project members' personal aspirations and objectives with the project's vision and purpose.

To support integrated teaming, the project manager must determine the integrated team structure within the guidelines of the organization's set of standard processes that will best meet the project objectives and constraints. Figure 17.2 shows that IPPD rules and guidelines must be part of the organization's set of standard processes and the project's defined process in a fashion similar to that for the organiza-

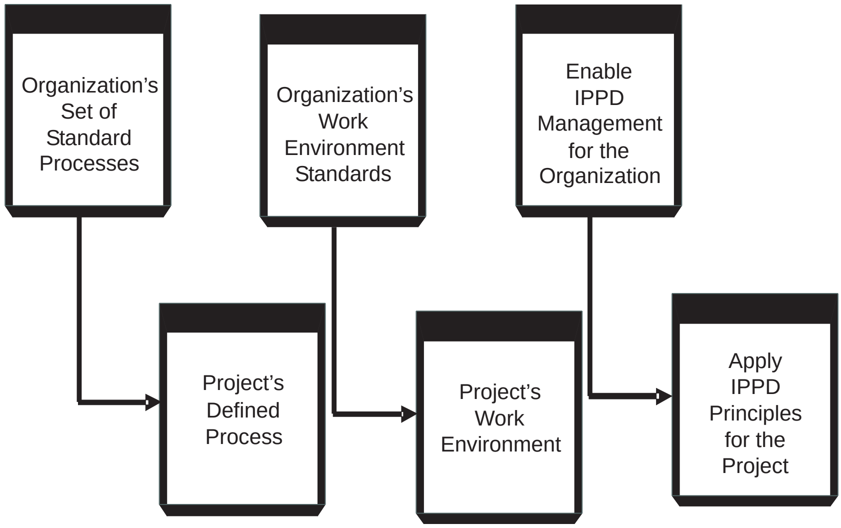


Figure 17.2 Integrated teams organizational and project view.

tional set of standard processes described in Chapter 15 and the organization's work environment standards described in Chapter 10.

The basis for defining integrated teams, their responsibilities, authorities, and interrelationships should come from the evaluation of factors such as these:

- Product requirements;
- Cost;
- Schedule;
- Risk;
- Resource projections;
- Business processes;
- Project's defined process;
- Organizational guidelines.

The team structure should be tied into the work breakdown structure used by the project, allowing each integrated team to be responsible for its own specific tasks and work products. The integrated team structure should facilitate the partitioning of responsibilities, requirements, and resources so that the right expertise and abilities are available to produce the assigned products. As the project evolves, the integrated team structure must be reevaluated for applicability. This reevaluation may result in the integrated team being reorganized, including the appointment of a new integrated team leader.

Preliminary Distribution of Requirements

The requirements should be preliminarily distributed to integrated teams even before the teams are officially formed as a sanity check to verify that the selected team structure is workable and covers all the necessary requirements, responsibilities, authorities, tasks, and interfaces. Potential representatives from the required disciplines should be identified even though the integrated team has not yet been formed to ensure an objective evaluation of the ability to satisfy the requirements that were preliminarily assigned to the integrated team.

Responsibilities and Authorities

The team leader should be chosen together with the assignment of planned responsibilities and requirements for each team. Integrated teams require a great deal of autonomy that necessitate a high degree of confidence at the organizational and project level in the team leader. Organizational and project influence on selecting the team leader should be used judiciously. When a new team leader and/or new team members come into the team, a review of the match between the new composition and the current responsibilities should be made and appropriate changes made as necessary. When defining the integrated team responsibilities and authorities, consider the following factors:

- Authority of the integrated team to pick its own leader;
- Authority of the team to implement subteams;
- Reporting chains;
- Reporting requirements such as cost, schedule, and performance status;
- Progress reporting measures and methods.

The Integrated Team

An integrated team focuses on the product life cycle to the extent required by the project. The sponsor, usually the project manager/leader, typically provides the integrated team with the product requirements they will be responsible for implementing, initial technical and business interfaces, and high-level tasks from the work breakdown structure and work packages. The project manager also needs to clearly define the relationship between the integrated team and the project and organization.

Selection Criteria for Integrated Team Members

Team members must be selected and positioned according to established criteria including:

- Knowledge, skills, and functional expertise related to tasks and responsibilities associated with the team's assigned work products;
- Interpersonal skills and ability to work in a team environment;
- Ability to complement the mix and knowledge and skills in the team;
- Potential to fulfill a significant responsibility in the team;
- Ability to acquire additional knowledge and skills or expertise related to the team's tasks;
- Educational and cultural background;

- Personal self-motivation.

The functional knowledge and related job skills within the integrated team are directly related to specific team tasks and responsibilities. Organizational business objectives must be identified, the core competencies required to support those business objectives must be defined, and the knowledge and skills profiles required for each core competency must be established.

Integrated Team Charter

Once the team has been selected, it is important to establish the team charter. The team charter is the contract among the team members and between the team and its sponsor for the expected work effort and level of performance. It is meant to solidify the rights, guarantees, privileges, and permissions for organizing and performing the team's objectives and tasks. It should establish the team's level of empowerment and independence. It should also identify how the team and individual performance and accomplishments are going to be measured.

Operating procedures and ground rules must also be developed to define and control how the team will interact and work together. They define the expectations and rules that will guide how the team works collectively, the degree of collective decision making, the level of consensus that is needed for team decisions, and how conflicts will be addressed and resolved.

Although it is important for the integrated team members to be chosen for their ability to get along with other team members, they must also be chosen for their highly crafted technical skills and possibly their individual contributions on other projects. Establishing the team charter, operating procedures, and ground rules simply serves to remind each individual that he or she has been chosen for an integrated team and must abide by that team's decisions. To complete the "rules of the game," it is necessary to clearly establish the roles and responsibilities of each member of the integrated team including each team member's anticipated contributions, level of involvement, and realm of influence each member is expected to have on the success and functioning of the team. Setting of the roles and responsibilities also includes these tasks:

- Determine how assignments are accepted.
- Determine how resources and input are accessed.
- Determine how works gets done.
- Determine who checks and reviews work.
- Determine how work is approved.
- Determine how work is delivered and communicated.

Integrated Product Team Used by the Department of Defense

Before we summarize the integrated team points made in this chapter, it is appropriate to provide an overview of the integrated product team (ITP) used by the U.S.

Department of Defense so those who have been involved with IPTs can compare them to the concepts presented in this chapter.

An IPT is a cross-functional team formed for the specific purpose of delivering a product for an external or internal customer. These cross-functional teams are made up of individuals who:

- Have complementary skills;
- Are committed to a common purpose, approach, and performance objectives;
- Hold themselves mutually accountable.

IPTs should be used for these reasons:

- To better achieve performance goals;
- To integrate information, knowledge, and experience;
- To create synergy through team dynamics;
- To develop more alternatives;
- To broaden team members' perspectives and knowledge;
- To develop team learning and flexibility;
- To facilitate culture change.

The basic principles of IPTs include:

- Open discussions with no secrets;
- Qualified empowered team members;
- Consistent, success oriented, proactive participation;
- Continuous up-the-line communications;
- Reasoned disagreement;
- Issues raised and resolved early.

IPTs should be used for oversight, management, program implementation, engineering, production, testing, and support. IPTs require a careful assessment of the context and business objectives needed. They also require collaboration and support from other departments or disciplines. Barriers to successful IPTs include:

- Lack of top management backing;
- Fuzzy team vision, mission, and direction;
- IPT Leader manages the team as individuals;
- Lack of team member empowerment;
- Assuming everyone has the competence to work well as a team.

The success of IPTs depends on recognition and management of the required culture change. It requires strong and continuous communication. It also requires a common positive proactive attitude for all team members.

The characteristics of a high-performing IPT include the following:

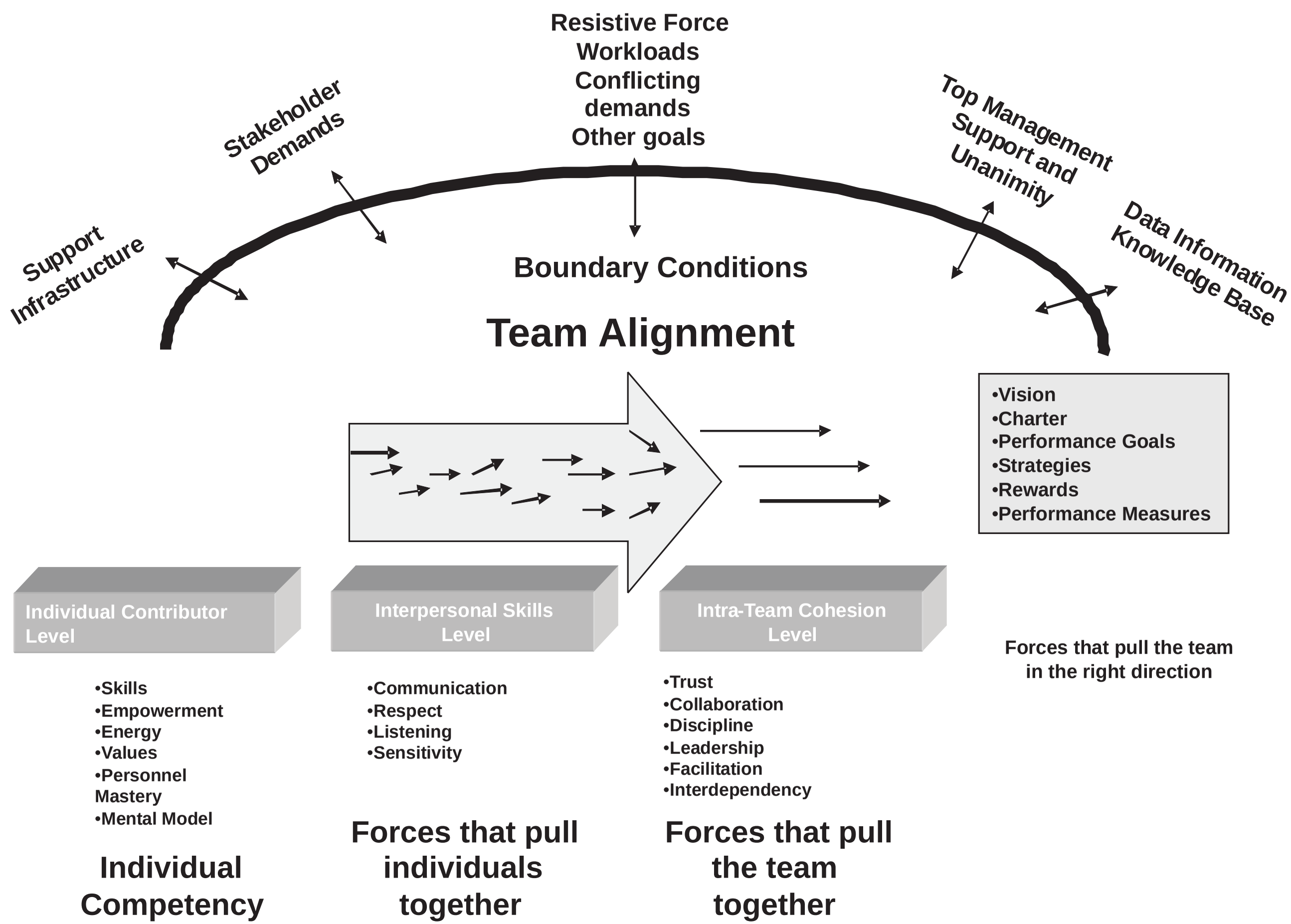


Figure 17.3 High-performance team model. (*After:* [1].)

- The focus is on collective performance, not isolated individual performance.
- Members share a commitment to a common purpose.
- Members have complementary skills.
- There is high trust and open communication among the IPT members.
- There is a high degree of mutual interdependence.
- Individual's goals are lined up with the team's goals.
- Members feel strongly about the team.
- Synergy is created among the team members to find win–win results.
- Disagreements are encouraged in discussions.
- Members are deeply committed to each other's personal growth and success.

The critical success factors for an IPT include:

- Team effectiveness;
- Collaborative culture of the IPT members;
- Team alignment;
- Team empowerment;
- Team motivation;
- Team learning capability.

Figure 17.3 summarizes the main points of an IPT by showing the necessary components for a high-performance team model.

Summary

Deciding to use an integrated team requires an understanding of business objectives and the ability to determine if the results will be cost effective. The use of an IPT can be expensive and politically charged unless an organization's culture has evolved to the point where it supports and nurtures integrated teams. Not everyone is a candidate to be a member of an integrated team. Those who are chosen will become a part of the "vision chain." They will align their personal aspirations with the vision of the integrated team, which is aligned with the vision of the project, which is, in turn, aligned with the vision of the organization.

Reference

[1] Bennet, D., "Integrated Product Teams and Integrated Product and Process Development," *Tutorial, Applied Concurrent Engineering Conference,* Seattle, WA, November 1996.

PART VI

Measurement and Analysis to High Maturity

CHAPTER 18

Establishing the Measurement Foundation

This chapter illustrates the strong measurement focus that is found within the CMMI® starting at Maturity Level 2. It discusses measurement needs at the organizational as well as project levels. As stated in the CMMI® introductory notes for the Measurement and Analysis process area [1], "The initial focus for measurement activities is at the project level. However, a measurement capability may prove useful for addressing organization- and/or enterprise-wide information needs. To support this capability, the measurement activities should support information needs at multiple levels including the business, organizational unit, and project to minimize rework as the organization matures. Projects may choose to store project-specific data and results in a project-specific repository. When data are shared more widely across projects, the data may reside in the organization's measurement repository."

It is my claim that a project *must* take into consideration the organization's measurement objectives, which it might inherit, as well as its own project's measurement objectives, which may come from customer demands, competition, new technologies, past project quality defects, and quality goals (Figure 18.1).

Why a Separate Measurement and Analysis Process Area?

The Measurement and Analysis Common Feature found in the CMM® for Software was one of the weakest parts of the original CMM®. In an effort to avoid presenting specific metrics, yet somehow give guidance about the measuring status and effectiveness of processes, the measurement requirements of the CMM® were not very clear and ended up not being of much use to organizations that were trying to improve their process capability. In the past, many individuals, projects, and organizations have even mistakenly believed, when embarking on their journey to achieve CMM® ML 2 and ML 3, that measurement was not necessary until the organization had actually accomplished CMM® ML 3. They believed that measurement only played a significant role once they began a Maturity Level 4 initiative and embarked on an implementation of the Quantitative Process Management and Software Quality Management key process areas. These same organizations focused on the use of status information to achieve their CMM® Maturity Level 2 objectives. They added a few measures once they developed their organization's standard software process, but frequently found out they did not have the right data to actually consider putting their process or subprocesses under statistical process control.

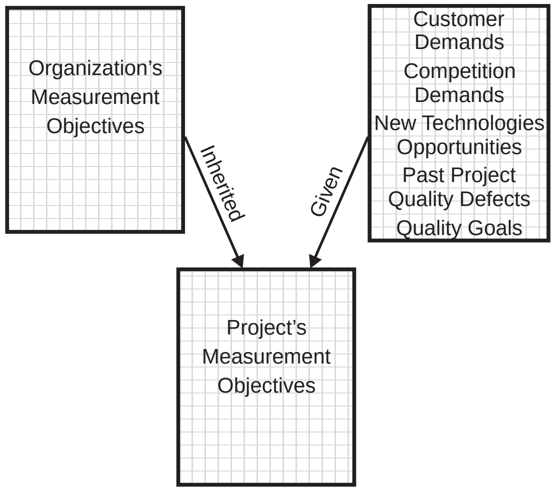


Figure 18.1 A project's measurement objectives.

The CMMI® authors created a new process area called Measurement and Analysis that assists organizations in getting their measurement programs started and simultaneously creating a measurement foundation that can be evolved as the organization moves from CMMI® ML 1 to ML 2 to ML 3 and beyond to higher maturity.

Will Hayes, of the Software Engineering Institute, developed material that was part of the CMMI®-DEV v1.2 upgrade material and part of the Intermediate CMMI® course material. His material clearly indicates that an organization starting on its measurement journey should understand the demands and benefits of quantitative measurement at Maturity Level 4 to understand the measurement path suggested by the process areas of Maturity Levels 2 and 3. Hayes's ideas are examined in more detail later in this chapter.

Measurement: Is It Really Necessary?

Before we delve into the contents of the Measurement and Analysis process area, let us better understand what motivates measurement in general. Is measurement really necessary? The following are questions that a project manager must consider and be able to answer to higher level management's satisfaction when a project is starting:

- Can it be done?
- How long will it take?
- How much will it cost?
- How many people will it take?
- What are the risks?
- What are the trade-offs?
- How many potential errors?
- How much does your current development process cost?

- How much value does each piece of the process add?
- What would the impact be of deleting, modifying, or adding a procedure to the process?
- What activities contribute the most to the final product cost?
- Have you tried to improve the current development process?
- What changes in cost/value resulted from that improvement effort?
- What processes represent the greatest potential for return on improvement investment?
- How would you quantify the value of the process improvement investment?
- Do you really want to know where the money is going in your development projects?
- What value do you think you are delivering to your customers? Do they agree?

These questions must be answered for every project. The demands for more quantitative answers goes up as the organization strives to distinguish itself from its competitors or show its clients its ability to manage quantitatively and be able to satisfy higher customer demands for quality and process performance. Higher level maturity organizations, as we will come to understand in Chapters 21 and 22, will have command of statistical techniques to predict with confidence their ability to satisfy those demands for quality and process performance.

Information Needs and Measurement Objectives

The Measurement and Analysis PA purpose states that an organization should develop and sustain a measurement capability that is used to support management information needs. What information needs are we talking about? The CMMI® provides us with some examples of sources of information needs including:

- Project plans;
- Monitoring of project performance;
- Established management objectives at the organizational level or project level;
- Strategic plans;
- Business plans;
- Formal requirements or contractual obligations;
- Recurring or other troublesome management or technical problems;
- Experiences of other projects or organizational entities;
- External industry benchmarks;
- Process improvement plans at the organizational and project level.

Again, we must ask, what is it about the project plans or technical problems or experiences of other projects or external industry benchmarks such as CMMI® appraisals that suggests an information need? Perhaps our ongoing project has not been meeting its delivery dates. Perhaps other projects have not been able to meet the functionality promises that were made. Perhaps technical problems that have reached production are causing significant rework and customer dissatisfaction. These "information needs" might result in project measurement objectives such as these:

- Reduce time to delivery
- Reduce total life-cycle cost.
- Deliver specified functionality completely.
- Demonstrate ability to implement increased customer requirements demands.
- Improve prior levels of quality.
- Improve prior customer satisfaction ratings.

Quantitative Quality and Process Performance Measurement Objectives

It is my assertion that a focus on quantitative quality and process performance measurement objectives are necessary for an organization striving to achieve CMMI® Maturity Level 2 as well. It is even clearly stated in Specific Practice 1.2 that measurement objectives should be specified as precise, quantifiable measures. The difference between ML 2, ML 3, and ML 4 is not in the statement of quantifiable measurement objectives, but in the capability of the organization's processes and people to come up with quantifiable measures that lead to predictability of quality and process performance.

Let us look at an example. An organization's CEO states that the organization and its projects should meet the delivery time, decrease the cost of poor quality, and meet the functionality promised with each delivery. Some individuals may believe that these statements meet the intent of measurement and analysis, but perhaps a second look is in order. Of course, any project in any organization should strive to do this. What is the information need? Better yet, the question that must be asked and answered is "Why should these be measurement objectives for the project and/or organization?" Another question that should be asked is "Why is there a focus on deliver on time"? Have the projects not been delivered on time in the past and it has become a problem? Is there a market window that when not met causes great financial loss? Are other business units dependent on our project's prompt delivery? Why make an organizational measurement objective to deliver with the promised functionality? Have projects in the past not been able to deliver on the agreed-on date with the full promised functionality? Is this causing customer dissatisfaction? Is this causing the organization to fall behind its competitors?

Understanding the organization's business objectives and the project's information needs is the first major requirement for establishing the organization's measurement foundation. Without this, measurement gets reduced to status information that is normally collected through project monitoring and control. Organizations and projects within them must continuously maintain traceability of the measurement objectives to the project's and organization's measurement objectives. The question of "Why are we measuring this?" must always be able to be answered. In addition, any project and organization must keep in mind that the measurement objectives can and should change to reflect the changing and evolving business and measurement objectives and understanding of the capability of the processes the project's are supposed to follow.

Different Project's Business Objectives Supporting a Common Organizational Business Objective—An Example

It is also important to understand that each project is not required to have the same project's business objectives or information needs to support the organization's business and measurement objectives. An example might help here. Let us assume that the organization's business objective is to reduce the cost of poor quality. Project A might have as an information need "How many defects are found and fixed using peer reviews?" Project B might have as an information need "How many defects are found and fixed in unit and integration testing?" Project C might have as an information need "What is the compliance percentage of its project members in following the organization's set of standard processes?" Quantitative measures corresponding to those projects' business objectives would then be developed to give the project an indicator of whether it was going to succeed in reaching its measurement objectives and, hence, support its project's business objectives and subsequently the organization's business objectives.

Organizations that quickly put down standard business objectives and expect each project to provide a standard measure (most often in the form of a stoplight chart) should not expect to receive the benefits that a measurement initiative based on the guidance of the CMMI® can truly provide.

The Value of the Measures

It is strongly recommended that this question be continually asked: "What value will this measurement be to the people who will be asked to supply the raw measurement data and to those who will receive the analyzed results?" Some years ago, I was working for a business unit of a large company in Europe. The CIO of the company wanted to establish a peer review culture in his organization. He turned over the implementation of this program to his deputy CIO. The deputy CIO instructed the quality assurance manager to go to each project and demand to know the percentage increase in the number of peer reviews conducted this past month compared to previous months. Project managers and members alike saw no value in this exercise but realized they had to find a way to "show" compliance each month. Many of the projects would hold at least one more peer review at the local tavern with a round or two of beers a day or two before the quality assurance manager was to come around so that they could say the percentage had increased. During a period of about 6 months, each project was reporting an approximately 70% increase in the use of peer reviews. I was serving as the lead assessor and during one assessment, the practitioners told me that they really did not conduct peer reviews. I reported this during the senior management presentation. The CIO was very upset and the deputy CIO angry at being put in a compromising situation. When the CIO demanded that I substantiate this claim, 65 practitioners stood up before I could speak and said, "That is what we told the assessment team. We really don't conduct peer reviews." Measures should always be useful to those who collect the data and use it!

Basic and Derived Measures

Basic Measures

A measurement program should always include the basic project management measures of monitoring and control and include (see discussion in Chapter 6):

- Attributes of the work products:
  - Size (lines of code, function points, pages of documentation, number of requirements, number of interfaces, number of logic gates, number of test cases),
  - Complexity,
  - Weight,
  - Form, fit, or function;
- Cost and expended effort;
- Schedule;
- Technical performance (completion of activities and milestones against the schedule);
- Staffing profiles;
- Resources:
  - Physical facilities,
  - Computers, peripherals,
  - Networks,
  - Security environment,
  - Processes;
- Knowledge and skills acquisition of project personnel;
- Training needs;
- Commitments;
- Quality (defects or quality attributes).

Derived Measures

A measurement program should also include derived measures. Derived measures are often expressed as ratios or composite indices and are based on combinations of data that are collected for the defined basic measures. *They are often more quantitatively reliable and meaningfully interpretable than the base measures used to generate them*. Examples of commonly used derived measures include:

- Earned value—actual cost of work performed compared to the budgeted cost of work performed:
  - Cost performance index (CPI),
  - Schedule performance index (SPI);
- Defect density—number of defects found per thousand lines of code;
- Peer review coverage;
- Test coverage;
- Defect escape rates;
- Defect removal efficiency;
- Usability;
- Reliability measures—mean time between failure.

The operational definitions of the measures should be stated in precise terms:

- What has been measured?
- How was it measured?
- What are the units of measure?
- Can the measure be repeated given the same definition and yield the same results?

Statistical Analysis Techniques and Data Types

Even though Measurement and Analysis is a Maturity Level 2 process area it does not mean that the use of statistical techniques should not be applied. Specific Practice 1.4 gives the reader guidance about when data analysis methods and tools are to be selected and about some of the issues to be considered, including the choice of descriptive statistics such as arithmetic mean, median, or mode.

Descriptive statistics are normally used to examine distributions on the specified measures such as central tendency, dispersion, or extent of variation or data points exhibiting unusual variation. Central tendency examples include mean, median, and mode. Dispersion refers to the spread of data or how much the values tend to differ from one another. Statements about the sampling criteria used to select a sample from a given population should be made and decisions documented along with the sampling choice rationale.

The use of visual displays and other presentation techniques such as pie charts, bar charts, histograms, scatter diagrams, and spider or Kiviat diagrams should become common as organizations establish their measurement and analysis foundation.

It should now be clear that measurement and analysis is not just about collecting basic measures and tracking an actual delivery schedule against the planned schedule.

The measurement and analysis process expects projects and organizations to not only establish the measures but to evaluate how well the measurement and analysis satisfied project and organizational assumptions. For example, did the statistical assumptions about the distribution of data turn out to be correct and how did this affect the performance of the project?

For completeness, we now look over the major points of the Measurement and Analysis process area not yet covered.

Specify Data Collection and Storage Procedures

The procedures that will be used to collect and store the data for each measure must be established and maintained. Explicit specifications of how, where, and when the data will be collected must be defined. Procedures for ensuring the data collected is valid must be developed, and the data must be stored in such a way that it is easily accessed, retrieved, and restored as needed. The following questions should be considered when an organization sets up its measurement collection and storage procedures:

- Has the frequency of data collection been established?
- Have the points in the process where the measurements will be made been determined?
- Has the time required to move measurement results from the points of collection to databases or end users been estimated?
- Has it been decided who is responsible for collecting the data?
- Has it been decided who is responsible for data storage, retrieval, and security?
- Have the necessary supporting tools for data collection, storage, analysis, and feedback been developed or acquired?
- If a database is used for storage of metrics data, would there be a value for real-time generation of graphics?

It is important to remember that effective data collection is always dependent on the actions of people. Therefore, consideration must be given to how easy it is for people to record metrics and to generate the required graphics. Initially, usage of a spreadsheet may be sufficient. However, as an organization transitions to the use of an intranet to facilitate communication, a spreadsheet may not be the most efficient way to generate graphics. Use of a database and active server pages to enable publishing of real-time graphics may be a better approach. If the interface between the individual recording the metrics data is easy enough to find and use, then the likelihood that data will be recorded increases. Also, if generating a graphics display requires little to no effort on the part of the person recording the data, then the likelihood that the metrics displayed will be current and actually used increases substantially. After all, metrics are of no use if all of the data is not recorded and is not translated into an output, suitable for analysis, within a reasonable time frame.

Specify Analysis Procedures

Analysis procedures must be defined and agreed to in advance. This was discussed earlier when we discussed statistical techniques. Consideration must be given to how the results will be fed back to the project manager and project members and what the most appropriate method of display of the results should be to ensure maximum understandability and usefulness.

It is always important to factor in the audience that will receive these measurement results. The feedback, content, and interpretation will differ if the audience is an intended user, a sponsor, a data analyst, or a data provider.

Collect and Analyze the Measurement Data

The measurement data should be collected as defined at the points in the process that were agreed on. Derived measures should be generated from the basic data and initial analyses should be conducted. The results from the initial analysis should be interpreted and fed back to the stakeholders to determine if the results are understandable and decisions can be made from them. It is most important for those who collect and analyze the measurement data to follow up and coach those who receive it to ensure their understanding and interpretation of the measurement results. Creation of a common metrics style guide is suggested. This ensures that expected output will be reasonably consistent and professional in appearance.

Store the Measurement Data and Analysis Results

A major point that the authors of the CMMI® made regarding the storage of the measurement data is repeated in the definition of the organizational measurement repository, and repeated again in the description of the quantitative project management process area. That point states: *Measurement-related information should be stored together with sufficient context so that the measures can be understood and interpreted for reasonableness and applicability*. Throughout the years, I have assessed many companies around the world that collected a lot of data but without any context at all. In addition, this data was normally handled by the "metrics expert" and was almost never used by the project manager or members for decision purposes.

It is strongly suggested that for any graphics displayed, one or two sentences should address what the graphics are displaying. For example, the sentences could say "A rising line means this… and a falling line means this...." Use of sentences like this should also be a mandatory part of the metrics style guide recommended earlier. It should also be part of the organization's metrics policy in order to ensure that no metrics will be published in the organization without some form of explanation of what the graphics mean. After all, the purpose of metrics is to facilitate communication of data in a visual manner so that the data can be more easily understood. Metrics are nice, but if no one can interpret them, they are useless.

Effectiveness of Processes

It is important for each project manager to know if the defined processes for the project are being followed by his/her project members. It is even more important for the project manager to know if the processes that his/her project members are following are working in the manner that they were expected to work. In other words, are the processes *effective*?

When we discuss effectiveness of processes, we want to know how well the process or processes are working for those who are following them. Many measures related to measurement and analysis focused on status. Here are a few examples that should illustrate the difference between the status of following a process and its effectiveness.

We will look at the process area of Requirements Management. If a project had 1,000 requirements, how effective would the requirements management process be? The status of 1,000 requirements is just that—a status number. It does not give any more information than if we said we had 10,000 requirements. We might expect the project to be more complex if we had 10,000 requirements compared to 1,000, but there is nothing to be said of the effectiveness of the Requirements Management process itself. What if we had the input that the project had 100 requirements change requests? Would that be enough information to give us reason to believe we could discuss the effectiveness of the requirements management process? The answer is still no. We do not know what type of requirements change requests are included in that 100. But if we had the data that told us we had 100 requirements change requests and 25 of them were "show stoppers," we would be able to state that we had an indicator of effectiveness of the requirements management process.

We would not consider that process to be very effective and would start the search to determine what was not working right. Perhaps the requirements elicitation process was faulty. Perhaps the requirements analysis process was not adequate. Perhaps the impact analysis was inadequate. If the data stated that of the 100 requirements change requests, 100 of them were "nice to have," then we would believe that our requirements management process was sufficiently effective.

Like all measures, effectiveness of processes measures produce indicators that must be evaluated if they are good or not. Here are a few other examples of effectiveness of requirements management measures that could be useful for a given project:

- Time spent on change requests up until a yes/no decision is made by the senior contract group;
- Impact of the change requests on project progress (i.e., effort spent on the change requests versus the amount of effort to execute the original project);
- Actual cost of processing a change request compared with budgeted or predicted costs:
    - Actually make the change;
    - Fill in the forms;
    - Impact analysis;
    - Authorization;
    - Replanning;

- Rescheduling;
- Renegotiating commitments;
- SQA effort;
- SCM effort;
- Review effort;
- Test effort.

Organization's Set of Standard Processes

Measurements needed for performing quantitative management may or may not be different from measurements needed for analyses performed with defined processes. This is part of the message of the CMMI®-DEV v1.2, and SEI instructors such as Will Hayes have developed and illustrated this message in the Intermediate CMMI® class and exemplified it in the new and enriched higher maturity class. In other words, the measures that are found and discussed in the Measurement and Analysis process area may indeed be basically the same at Maturity Level 3 as they were at Maturity Level 2, only now, the measures are made on the organizational set of standard processes. Organizations are encouraged to try not to limit what they measure simply to satisfy process areas such as Measurement and Analysis.

To perform quantitative management, analysis of a history of measurement data is required. It is easier to collect statistically valid data that can be used to benefit most of the projects in an organization if the measures are based on the use of common or standard processes.

Besides being a CMMI® Maturity Level 3 requirement, developing a set of organizational standard processes builds a common vocabulary and allows others to anticipate the behavior of other groups and be more proactive in their interactions. Having a set of organizational standard processes allows the organization to measure a *controlled set* of processes to gain economy of scale. Trends can be seen and predictability can start. Process performance baselines can be economically developed to support quantitative management later.

One of the major components of the Organizational Process Definition process area is the organization's measurement repository that was defined and discussed in Chapter 15. The CMMI® description found at Maturity Level 3 is based on the foundation laid by the Measurement and Analysis process area at Maturity Level 2.

Organization's Measurement Repository

The organization's measurement repository contains:

- Product and process measures that are related to the organization's set of standard processes;
- The related information needed to understand and interpret the measurement data and assess it for reasonableness and applicability;
- Operational definitions for the measure that specify the point in the process where the data will be collected and the procedures for collecting valid data;

- Examples of classes of commonly used measures:
  - Size of work products (lines of code, function points, complexity);
  - Effort and cost;
  - Actual measures of size, effort, and cost for software;
  - Quality measures;
  - Work product inspection (peer review) coverage;
  - Test or verification coverage;
  - Reliability measures.

Slightly More Advanced Measures

When discussing defects discovered during peer reviews, it is much more interesting to think about defects in terms of *major* or *minor* defects where the boundary line between the two classifications depends on the influence of the defect on the cost, schedule, performance and quality, functionality, risk, and customer satisfaction of the product that is received by the end user.

Even the classification of defects into major and minor defects is not as informative as one would need to focus limited resources and care for product quality at the same time. Placing major and minor defects into categories (segmenting) allows the peer review team and the measurement team that will analyze the defects to see trends and focus energy on improving processes that will result in a greater benefit to the organization's business as a whole. Categories of peer review defects may include imprecisely stated requirements, ambiguous requirements, performance variables that are not quantified, requirements that are not testable, missing items, interface errors, and logic errors.

Peer review data can also reveal the effectiveness of the peer review calculated by comparing the number of major defects found in a life-cycle stage compared to the total number of defects found so far. After an organization has collected peer review data for life-cycle work products throughout the entire life cycle and can start to see trends, it is possible to mathematically calculate the remaining defects in a given life-cycle work product. Hypothesis testing may also be used to compare the results of two different sets of peer reviews after a process improvement change.

A caveat about peer review metrics is to ensure that the metrics are not used in a negative fashion against the person who authors the peer-reviewed document or code. Remember, a hallmark of effective peer review is that the results cannot be attributed to any person.

Defects identified through the Performance Process and Product Quality Assurance (PPQA) process area activities and audits after an item has completed peer review are a rich source for potential defect prevention activities. If the peer review process has been effective, then the number of defects in the product reviewed following the peer review should be little to none. If significant defects are identified, then the result of the PPQA audit of the product will point to a very inefficient peer review process. Remember, that is the purpose of peer review: to catch the defects before the product is approved for release. Quality assurance is not a substitute for an ineffective peer review process. So if a QA responsible or team is finding defects after a product has completed peer review, the peer review process may not be as effective as the project or organization thinks it is. This requires a corrective action on the part of PPQA personnel.

Testing defects can also be subdivided into categories and testing effectiveness can also be calculated. The defects found in testing can be compared to those found in peer reviews, leading to a more focused causal analysis on the origin of the defects and the effectiveness of the engineering processes that are allowing the defects to be injected into the evolving system. Test coverage measures are another measurement that helps project members to more accurately report the extent of the testing that is carried out and understand the product quality that is being shipped more thoroughly.

Goal–Question–Metric Paradigm

The goal–question–metric (G-Q-M) paradigm developed by Vic Bacilli and Dieter Rombach documents the purpose for which *any* measurement and analysis is done (Figure 18.2). G-Q-M is the principal concept on which the Measurement and Analysis process area is based.

- Goals are issues of importance for the organization.
- Questions define the issues in such a manner that their answers indicate progress toward achieving the goals.
- Metrics supply the data that provides the answers to the questions that indicate the status of efforts to achieve the goals.

Summary

The strong measurement focus that can be found and utilized from the CMMI® has been presented. Starting with basic project management measures and an understanding of getting a measurement program started through the implementation of the concepts found in the Measurement and Analysis PA, it guides the reader to the

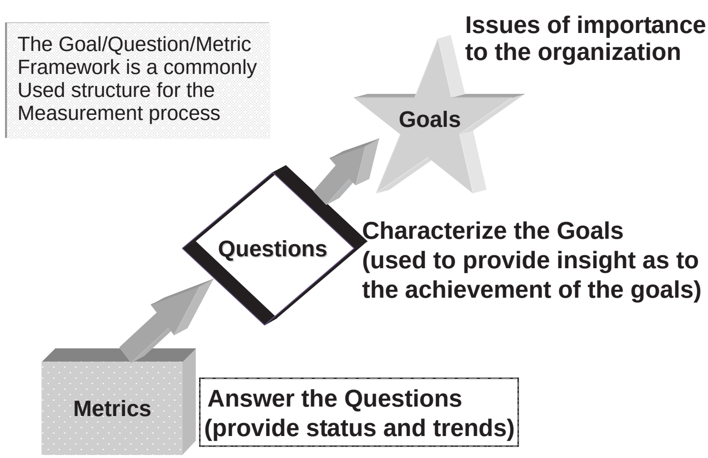


Figure 18.2 G-Q-M paradigm.

Level	Process Characteristics	Measurement Implications
5 Optimizing	Process improvement is institutionalized	Continuing improvement is based on business objectives and cost-benifit analysis.
4 Managed	Product and process are quantitatively controlled	Data analysis is based on the principles of statistical process control. Actuals are compared to expected values of mean and variance.
3 Defined	Technical practices are integrated with management practices and institutionalized	Consistent definitions exist across projects. Management and Quality data is collected across the organization.
2 Repeatable	Project management practices are institutionalized	Projects collect management data about cost, effort, size, schedule, etc. Different projects may use different definitions.
1 Initial	Process is informal and ad hoc	Measurement is haphazard, but may yield cost and effort data.

Figure 18.3 Measurement implications of process maturity.

establishment of an organizational measurement repository, the collection of peer review and test data, and the evolution of the organizational process measures that provide the building blocks for effective statistical process control and quantitative project management.

The measurement implications shown in Figure 18.3 (taken from the Kasse initiatives “Multiple Views of the CMMI®” presentation) sums up this powerful feature of the CMMI®.

- *Initial.* Measurements are hard to make and data is difficult to collect due to the chaotic processes.
- *Managed.* The project can collect measures based on the actuals to estimates they collect including size, effort, schedule, technical activities, risks, and staffing.
- *Defined.* The organization’s standard software process allows the start of quantitative measurement data to be collected and placed into the process database.
- *Quantitatively managed.* As stated, the data collected allows trend analysis to be carried out. Statistical process control methods are used to manage the project quantitatively.
- *Optimizing.* Organization can quantitatively decide how much improvement to try for and its cost benefit. Organizations can look into new businesses and determine the feasibility of doing it from a technical and now financial point of view.

Reference

[1] CMMI® Product Development Team, *CMMI® for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1 Continuous Representation,* CMU/SEI-2002-TR-011, ESC-TR-2002-011, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, March 2002.

CHAPTER 19

Basic Statistics and Probability Distributions

High Maturity and High Performance

The focus on high maturity and just what that means has increased dramatically during the past few years motivated in part by the large number of high-maturity appraisal results that have not always translated into equally high process performance. There is ongoing discussion and debate as to whether the SEI has "raised the bar" regarding high maturity or whether the industry has evolved together with the SEI to be able to understand the principles behind high maturity. In my opinion, the bar has been raised, but not in a way that was meant to be negative to those organizations that have followed the guidance of the CMM® and now CMMI®.

Total quality management is all about continuous process improvement in order to achieve greater degrees of product and service quality. The SEI's CMM® and CMMI® are offshoots of TQM and so follow that philosophy as well. To be fair, the SEI has simply responded to industry's cry for high performance to go along with the reported high-maturity levels. The concepts were always part of the description of the high maturity level process areas. But now, examples from related disciplines, techniques such as Six Sigma and Team Software Process, and basic statistics are overtly being offered along with the simple text of the CMMI®. Perhaps the examples are causing organizations' consternation resulting in a defense of their level. I hope that most of the organizations that have used the SEI's CMM® and CMMI® guidance are pleased with the clearer and more useful explanations of high maturity that the CMMI® v1.2 [1] and the SEI's related trainings are providing.

Measurable results, stabilized processes, predictability, quantitative project control, more effective causal analysis, and innovations applied to quantitatively understood processes are now easier to focus on and easier to recognize once they have been obtained.

This chapter is not intended to teach anyone about statistics. The industry abounds with more than sufficient books and articles to do that. It is intended to bring basic statistics concepts into the forefront of the reader's consciousness with regards to building a measurement and analysis system. It also is intended to illustrate the importance of a basic statistics vocabulary in order to understand the guidance the CMMI® provides to those organizations who wish to reach a high maturity level.

Techniques to Control Processes in High-Maturity Organizations

To support the point just made—that the statistical concepts being presented today by the SEI have always been a part of the description of the high maturity level process areas—some very basic statistical definitions will be presented in this chapter. The topics list was taken from a presentation made by Mark Paulk when he was still part of the Software Engineering Institute. His presentation was titled "Basic Statistics for Software Engineers" and was presented at the ESEPG in 2001.

High-maturity organizations typically use control charts and other statistically rigorous methods to control their processes. Control charts included XmR, XbarR, *u*, and *z* charts. Some high-maturity organizations use techniques such as:

- Cost of quality to determine the effectiveness of their process improvement activities;
- Prediction intervals;
- Confidence intervals;
- Pareto analysis;
- Process modeling and simulations;
- Designed experiments (DOE);
- Six Sigma;
- Analysis of Variance such as ANOVA or MANOVA.

Statistics

In Chapter 18 we introduced the topic of statistics. Words such as *central tendency, dispersion, extent of variation,* and *distribution* were mentioned. Central tendency implies location, the balance point or middle of a group of values such as the mean, median, or mode, as listed in Chapter 18. Dispersion refers to the spread of data or how much the values tend to differ from one another. The terms *range* and *standard deviation* are used to describe dispersion. Used together, they provide an understanding of the likelihood of a particular value. Statistics, specifically hypothesis testing, enables us to place a confidence interval or range on the central tendency. Let us examine these basic statistical terms a bit more.

Mean

Suppose you were given five numbers and asked to find the average or *mean* of those five numbers: 1, 4, 5, 8, and 2. The average of these five numbers $\overline{x} = 1/5\ (1 + 8 + 3 + 6 + 2) = 4$. Λετ x_j = an individual value of x. Then the mean of any number of values $x_1, x_2, \ldots, x_n$ can be represented by:

$$\overline{x} = 1/n\left(x_1 + x_2 + x_3 + x_4 + \ldots + x_n\right) \text{ or}$$
$$x = 1/n \sum_{i=1} n, x_i$$

Median

The *median* is another measure of central tendency. If one sorts the data by magnitude then the median is the value in the middle. There are as many numbers bigger than the median value as there are smaller. The median is often more illuminating than the average where the occasional value might distort the average.

For example, we are given the following auction figures for buying a house in thousands of dollars:

185 190 145 220 1060 200 170

Sorting the numbers yields

145 170 185 190 200 220 1060

The mean would be 310. The median would be 190—a much better indicator. If there are an even number of values, you should average the middle two values to get the median.

Mode

The *mode* is another measure of central tendency. It means the most common number in the data set. Consider this data set:

7 8 7 8 8 9 6 5 10 8

Sorting the data into order yields

5 6 7 7 8 8 8 8 9 10

If the numbers represented shoe sizes, the shop owner would probably want to stock the most commonly sold shoes rather than the average or even the median foot size of their customers.

Variance

Looking at the diagram in Figure 19.1, we can see that the shooter on the left has the tighter group of shots but is off target. The shooter on the right seems to be more on target, but is not as good a shot. The shooter on the left can probably fix his/her problem quite easily by adjusting their his/her sites. The shooter on the right may find the problems more difficult to correct.

Two separate concepts are being represented in Figure 19.1:

- *Accuracy:* the distance between the process average and the target;
- *Precision:* the tightness of the grouping.

The shooter on the left is more *precise,* but the shooter on the right is more *accurate*. We use the term *variance* to measure the precision: $s^2 = 1/n - 1 \sum_{i=1,n} (x_i - \overline{x})^2$.

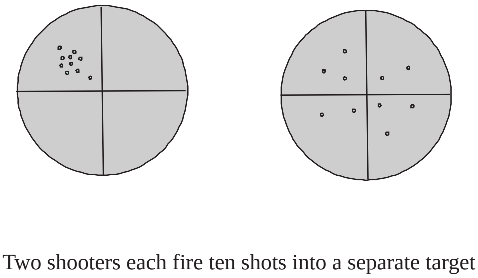


Figure 19.1 Variance.

This indicates the distance of each individual value from the mean; in other words, it is a measure of the precision of each of the individual data values. We divide by $n - 1$ to compensate for a slight bias that occurs when we use small samples to predict the behavior of a large population. If we were using the total population, we would divide by n. The *standard deviation* is the square root of the variance: $-\sigma = (s^2)^{1/2}$, where the Greek sigma represented a measure of variation and quality (Figure 19.2).

Probability Distributions

A *probability distribution* is a mathematical model that is used to describe the characteristics (shape, center, and spread) of a population. *The characteristics are normally expressed as an "assumption" for the procedure and can be represented using an equation.* A question that many of us might ask is "Where do probability distributions come from?" The National Graduate School provides us with the graphical

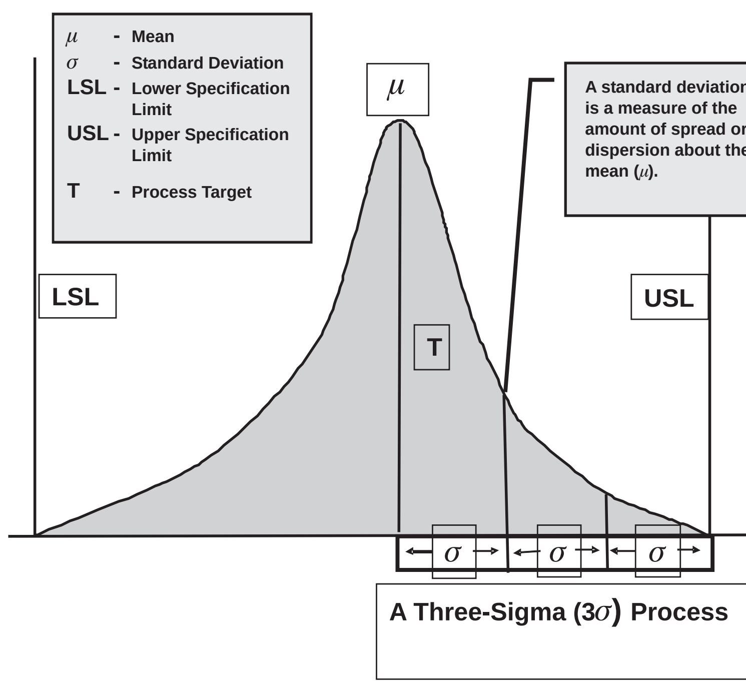


Figure 19.2 Sigma is a measure of variation and quality.

view shown in Figure 19.3 to answer this question. As you can see from the figure, many types of distributions exist. Each one is used to determine the probability of occurrence for a specific value or range of values contained within a population. Examples of distributions that you may come across in statistical process control are discussed next.

Discrete Probability Distributions

Variables represented by a discrete distribution (Figure 19.4) are integers, either binomial (0 or 1, yes or no, and so on) or counts (number of days, number of defective items, and so on). Normally the x-axis is used for the variable of interest and the y-axis is the corresponding probability of occurrence for that variable. An example of a discrete x-variable is the type or category of defect, and the y-variable is the probability or likelihood that each type or category of defect type will occur.

Binomial Probability Distribution

The binomial distribution (Figure 19.5) is used when items are evaluated using *attributes*, such as in quality control settings where the proportion of defective product is evaluated. An attribute is present or not with independent outcomes. The probability of an attribute being present is constant in each trial. The probability function for the binomial distribution is:

$$P_{(x)} = (n/x)p^x(1-p)^{n-x}$$

where

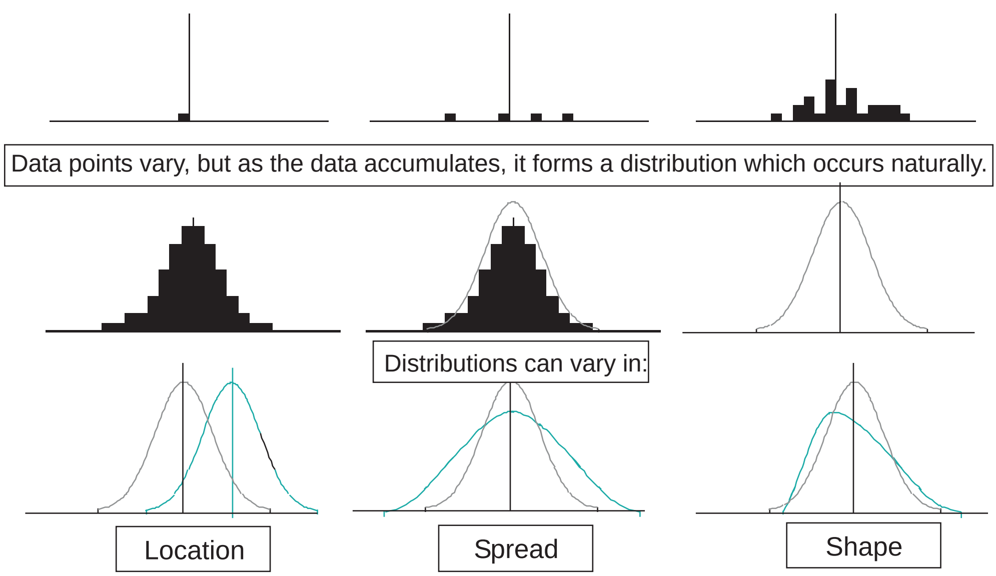


Figure 19.3 Probability distributions. (*From:* [2]. © 2008 National Graduate School of Quality Management. Reprinted with permission.)

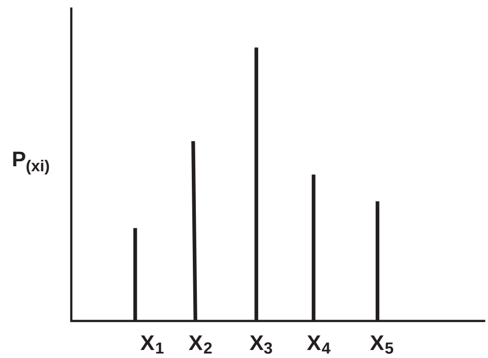


Figure 19.4 Discrete distribution. (*From:* [3]. © 2004 McGraw-Hill, Inc. Reprinted with permission.)

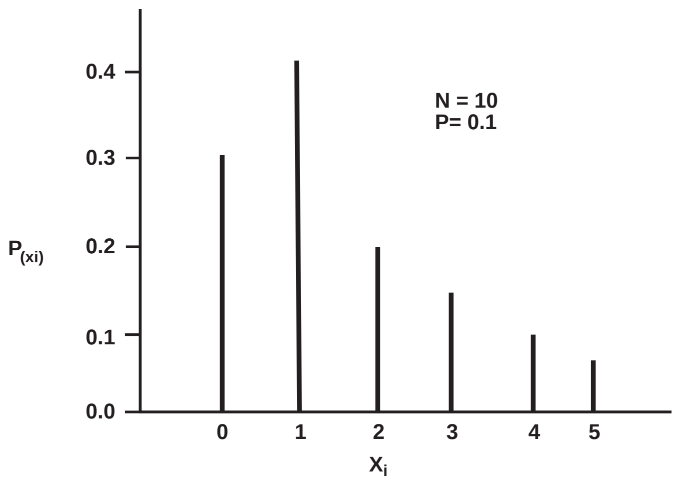


Figure 19.5 Binomial distribution. (*From:* [3]. © 2004 McGraw-Hill, Inc. Reprinted with permission.)

$P(x)$ is the probability of exactly x successes;

P is the probability of success;

x is the number of successes, and $x = 0, 1, 2, 3, ..., n$;

n is the number of trials;

mean or $\mu = np$;

variance $= np\ (1 - p)$.

Poisson Distribution

The Poisson distribution (Figure 19.6) is used when we want to determine the probability of the number of occurrences on a per-unit basis such as per-unit time. For example, how many test defects show up in 40 hours of testing? The resulting data is a count of events. The events are rare and independent of each other. The probability function for the Poisson distribution is:

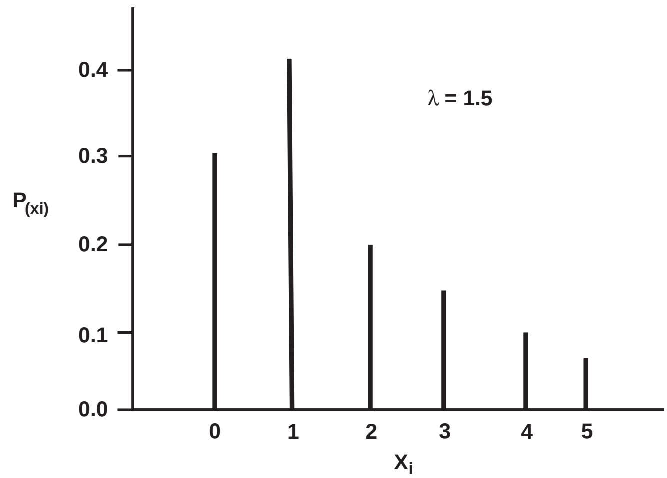


Figure 19.6 Poisson distribution. (*From:* [3]. © 2004 McGraw-Hill, Inc. Reprinted with permission.)

$$P_{(x)} = \left(e^{-\lambda}\lambda^{x}\right)/x!$$

where

e is the base of the natural logarithm = 2.718;

x is the number of occurrences per unit, and $x = 0, 1, 2, 3, \ldots, n$;

mean = λ;

variance = λ.

Continuous Probability Distribution

Values in a continuous distribution are measured on a continuous scale, as shown in Figure 19.7. Examples of continuous variables are length, temperature, and volume.

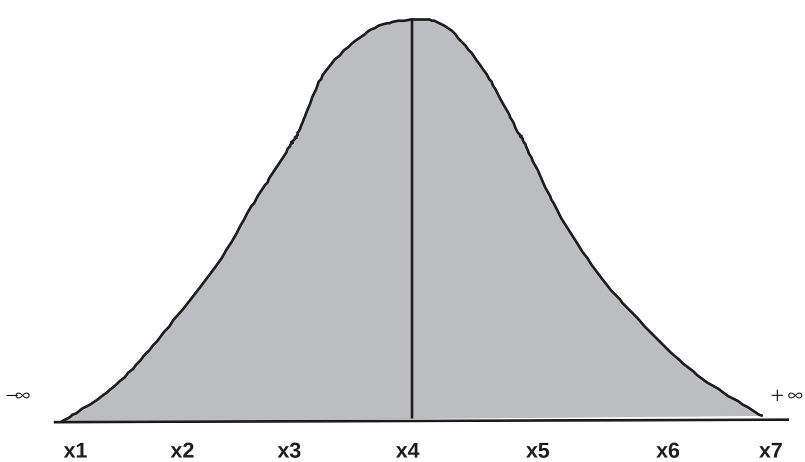


Figure 19.7 Continuous distribution. (*From:* [3]. © 2004 McGraw-Hill, Inc. Reprinted with permission.)

Standard or Normal (Gaussian) Distribution

The normal distribution (Figure 19.8) is both the most important and most commonly applied probability distribution. The normal distribution is often referred to as a *normal curve* and has the following characteristics:

- Smooth and continuous;
- Bell shaped and symmetrical;
- Both tails are asymptotic to the x-axis,
- The total areas under the distribution curve is 1;
- The mean, median, and mode have the same value.

A normal distribution where $\mu - 0$ and $\sigma = 1$ is called a standard normal or z distribution. The area under the normal curve is distributed as follows:

$-1 < z < +1 = 68.3\%$ of the total area;

$-2 < z < +2 = 95.4\%$ of the total area;

$-3 < z < +3 = 99.7\%$ of the total area.

If a data set is normally distributed, then the standard deviation and mean can be used to determine the probability of observations within a selected range.

Exponential Distribution

The exponential distribution (Figure 19.9) is used when we want to determine the probability of an event occurring over space or time intervals. Typical applications are in reliability engineering (product life expectancy) and in customer service settings (waiting time at a bank). The mean of an exponential distribution is referred to as mean time to failure, but can also be thought of as mean time to service in customer service applications.

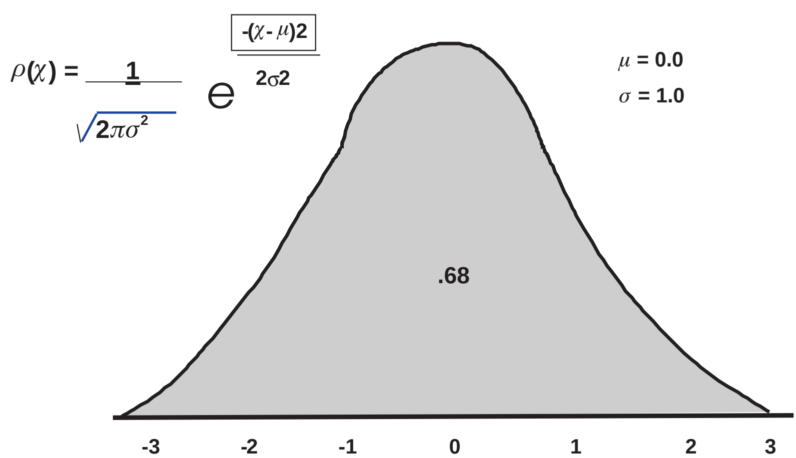


Figure 19.8 Normal distribution. (*From:* [3]. © 2004 McGraw-Hill, Inc. Reprinted with permission.)

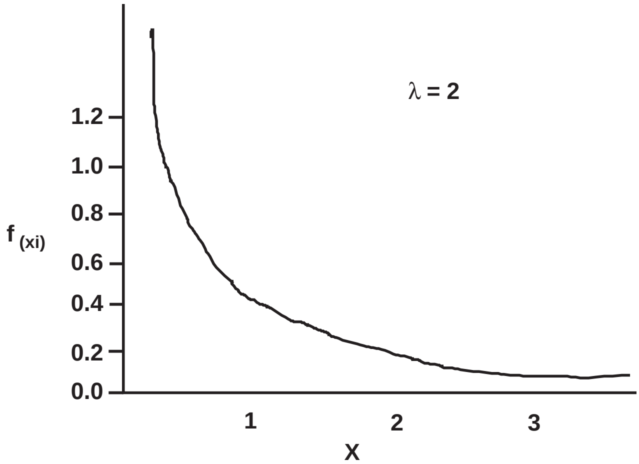


Figure 19.9 Exponential distribution. (*From:* [3]. © 2004 McGraw-Hill, Inc. Reprinted with permission.)

Central Limit Theorem

The central limit theorem (Figure 19.10) states that for almost all populations, the *sampling distribution* of the mean can be approximated closely by a normal distribution provided the sample size is sufficiently large. What this means is that no matter what kind of distribution we sample, if the sample size is big enough, the distribution for the mean is approximately normal. Furthermore, this normal distribution will have the same mean as the parent distribution, and variance equal to the variance of the parent divided by the sample size. This is the key theoretical link between the normal distribution and sampling distributions. The central limit theorem underlies all statistic techniques that rely on normality as a fundamental assumption.

Sampling Distribution

The sampling distribution mentioned in the central limit theorem is used to get from the basic statistic to an estimate of the population parameter. To understand it, we

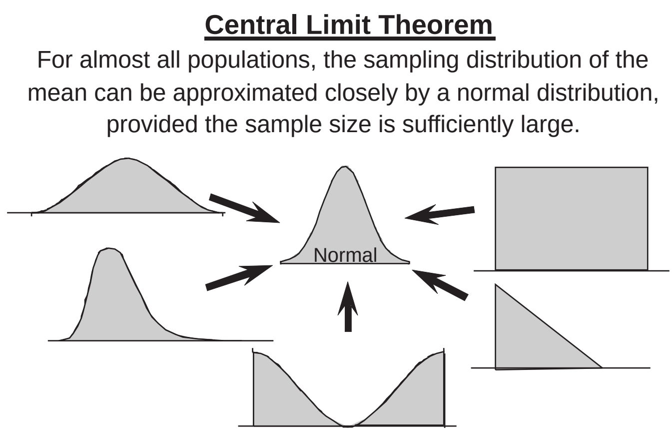


Figure 19.10 Central limit theorem. (*From:* [2]. © 2008 National Graduate School of Quality Management. Reprinted with permission.)

need only imagine an experiment in which we take three independent samples instead of the one typical of most studies. For all three independent samples, we will compute a single statistic, the mean of the response. Even though all three samples came from the same population, you would not expect to get the same mean from each sample. They would differ due to the natural fluctuations of drawing a sample. We would, however, expect to yield a similar estimate for the mean because the samples were drawn from the same population. Now simply imagine taking an infinite number of samples from the same population and computing the mean for each one. If we plotted these on a histogram, we would find that most of them converged on the same central value and fewer samples would have averages significantly away, up, or down from that central value. In other words, the histogram would show a normal distribution as described earlier. The distribution of an infinite number of samples of the same size as the sample in our study is known as the *sampling distribution*.

Importance of the Data

As you might have guessed, the real effort is not in analyzing the data statistically; it is in understanding how the data was collected, over what period of time, what the constraints were, what the population of people was that existed when the data was collected, what the ability of the people was who participated in the activities from which the data was collected, and so forth. In other words, it is about the validity and usefulness of the data.

Some of the top Six Sigma consulting companies have reported taking up to 6 months to review a company's data. During that process they had to determine which data was valid and which was not before they could arrive at the point where the population's characteristics were understood and the probability distribution of the data recognized and they could finally think about which statistical analysis techniques would be appropriate.

Attributes or Discrete Data

Data as counts are not very interesting. They may support status reporting, but they do not support measurement-based decision making very well. Six Sigma courses and books and the SEI's Understanding High Maturity Practices course provide us with another way to view our data. Processes are easier to understand and improve if we segment or stratify our data. *Segmentation* is the act of grouping the data according to one of the data elements. This implies that the categories or buckets of data are nominal or, in other words, *not ordered*. Nominal data includes:

- Types of customers;
- Document types;
- Types of projects;
- Types of defects;
- Types of verification techniques.

Stratification is the grouping of the data according to the value range of one of the data elements. This implies that the categories or buckets of data are *ordered*. Ordinal data includes:

- High, medium, or low rating;
- Severity ratings such as critical, intermediate, or low;
- Priority ratings;
- Criticality ratings.

Nominal and ordinal data are referred to as attribute or discrete data. When we want to examine attribute data, the focus is on learning about one or more specific nonnumerical characteristics of the population being sampled. When working with attribute data, the key drivers of sample size are sample precision or how close the estimated value is to the actual population value and the expected value of *p*, the population proportion used is the sample size calculation.

Variable or Continuous Data

Another type of data is called variable or continuous (see the continuous probability distribution discussion earlier in the chapter). With variable data, an actual numerical estimate is derived for one or more of the characteristics of the population being sampled. Variable data can be classified as interval or ratio data. Interval data has data that has equal intervals; ratio data is interval data that has an absolute zero value.

Examples of variable data include:

- Defect density;
- Cycle time;
- Peer review effectiveness;
- Cost and schedule variance measures;
- Temperature.

When working with variable data, sample precision and population variance have the greatest influence on sample size. *Those populations with high variance require larger sample sizes than those having low variance for any given sample precision.*

With both attributes and variables data, if greater degrees of sample precision are desired, larger sample sizes will be required.

Figure 19.11 taken from the SEI's Designing Products and Processes Using Six Sigma course summarizes the information presented on attributes and variables data.

Influence of Data Types

With this background let us reexamine basic measures and derived measures. The basic measures are normally captured as counts, whether they are nominal or ordinal, implying that the nominal data has some sort of ordering. Attributes data is counted and plotted as discrete events. Examples of attribute data include:

Types of Data		Description	Examples
Attribute or Categorical Data	**Nominal**	Categories or buckets of data with **no ordering**	Defect types Language types Customers Document types
	Ordinal	Categories or buckets of data with **ordering**	Severity ratings Priority ratings Customer Satisfaction ratings High, Medium, or Low ratings
Continuous Data	**Interval**	Data measured on a scale that has **equal intervals**	Productivity Defect Density Preparation Rate Cycle Time Size Test Hours
	Ratio	Interval data that has an **absolute zero**	

Figure 19.11 Data types. (*From:* [4]. © 2007 Software Engineering Institute. Reprinted with permission.)

- Shipping errors;
- Percentage waste;
- Number of defects found;
- Number of people with certain skills on a project;
- Percentage of projects using formal inspections;
- Team size;
- Staff hours logged per task.

Variable data or ratio data increases the choices that may be made for hypothesis testing. It may be becoming clearer to the reader that, although basic measures have their place in project management and measurement, derived measures, which are normally represented as ratio data, lend themselves to a wider variety of statistical techniques that can help us make comparisons and help with hypothesis testing.

Regardless of what type of data we have, we must determine if our measurement system yields accurate, precise, and reproducible data. Frequently encountered problems include wrong data and missing data. We must be able to determine if any values are out of bounds or if the frequency of each value makes sense.

Measurement System Error

One of the main topics found in the Understanding CMMI® High Maturity Practices course offered by the SEI is the understanding of the impact of poor quality data that typically results in measurement errors. Measurement errors come from:

- Allowing different operational definitions such as what is a line of code or how do we determine the duration of a project;
- Double duty, or overtime that is not tracked because the effort of data collection is owned by the accounting department and not the project manager;
- Trying to "pass the test," which occurs when the project team members produce a process description to satisfy what they think the CMMI® is asking for;
- Failure to provide adequate resources;
- Failure to provide adequate and timely training because the highest skill needed is assumed to be available on the project;
- Lack of consequences if poor quality data is used in predictions that have negative results.

Measurement error components consist of precision, accuracy, and stability over time. Precision and accuracy were previously illustrated. Measurement system variability can be further partitioned into:

- *Measurement system repeatability.* The variation that results when repeated measurements are made under identical conditions. In other words, the ability of one operator using one gauge to measure one part multiple times with minimal variability. How consistent is a person with him- or herself?
- *Measurement system reproducibility.* The variation that results when different conditions are used to make the measurement. In other words, the ability of multiple operators to produce similar average measures for multiple parts with minimal variability in the average measured values. What is the variation between different people, different environmental conditions, and different software modules?

A test known as a gauge repeatability and reproducibility (GRR) test can be conducted to determine whether excessive variability exists in the measurement system. The measurement error guide for continuous data states:

- If the %GRR is < 10%, the measurement error is acceptable.
- If the %GRR is between 10% and 30%, the measurement effort is unacceptable except for critical measurements and the measurement process should be improved.
- If the %GRR is > 30%, the measurement error is unacceptable.

Hypothesis Testing

Hypothesis testing is a formal way of making a comparison and deciding whether or not the difference is significant based on statistical analysis. It is the application of statistics to confirm or deny a theory about the value of some population parameter. Hypothesis tests can be used to make decisions about any process that we can sample. Examples of hypothesis testing applications include:

- Average waiting time in an airport security check line,
- Types of defects through structured walkthroughs compared to unit testing,
- Projects being implemented within 5% of the promised functionality.

Hypothesis tests are used to evaluate process performance relative to a standard or specification to determine if differences exist between processes or to verify process improvements by comparing before and after data.

Hypothesis tests use two opposing conditions:

- *Null hypothesis:* There is no difference between the two samples of data.
- *Alternative hypothesis:* There is a difference between the two samples of data.

An example set of hypotheses would be:

- *Null hypothesis:* The size of small effort projects is equal to the size of large effort projects.
- *Alternative hypothesis:* The size of small effort projects is less than the size of large effort projects. (The alternative hypothesis is the difference we seek to learn about.)

However, the type of hypothesis testing we might use is dependent on the data type we are analyzing as we might expect. Various types of hypothesis testing are shown in Figure 19.12.

Statistical Intervals

The three types of statistical intervals—confidence, prediction, and tolerance intervals—are discussed next. See also Figure 19.13.

Confidence Interval

In statistics, a confidence interval (CI) is an interval estimate of a population parameter, or the expected interval for the central tendency of a process or subprocess.

- Confidence intervals are used to indicate the reliability of an estimate. For example, a CI can be used to describe how reliable survey results are. All other things being equal, a survey result with a small CI is more reliable than a result with a large CI.
- The confidence interval establishes a way to test whether or not a significant change has occurred in the sampled population. This concept is called significance or hypothesis testing.
- Being able to tell when a significant change has occurred helps in preventing us from interpreting a significant change from a random event and responding accordingly.
- Confidence intervals are the most prevalent form of interval estimation.

Data Types	Interval or Ratio (Parametric Tests)		Ordinal (Non-Parametric Tests)		Nominal	Proportion
# Samples (Data Groups)	Mean	Variance	Median	Variance/Fit	Similarity	Similarity
1 Sample	1 Sample 1 Test	1 sample Chi-Square test	1 sample Wilcoxon Signed Ranks Test	Kolmogorov-Smirnov Goodness of Fit test	>2cells Chi-Squares Binomial Sign Test =2 cells	1 Proportions Test
2 Samples	Independent 2 Sample 1 Test Paired 1 test Paired	Normal F Test Levene Test Not Normal	Independent Mann Whitney U Test Wilcoxon Matched Paired	= Medians Siegel Tukey Test Moses Test ≠ Medians	Fisher Exact Test (1 way ANOVA): Chi-Square Test	2 Proportions Test
3+ Samples	ANOVA (1 & 2 WAY ANOVA; Balanced ANOVA; GLM) MANOVA (General and Balanced)	Normal Bartlett Test Levene Test Not Normal	Independent Krukal Watis 1 Way ANOVA Friedman 2 way ANOVA Paired	Van der Weardan Normal Scores Test	Chi-Square Test	ANOM (Analysis of Means)

Figure 19.12 Hypothesis testing decision matrix. (*From:* [4]. © 2007 Software Engineering Institute. Reprinted with permission.)

Prediction Interval

Expected interval for a future data point of a process or subprocess. Prediction intervals predict the distribution of individual points, whereas confidence intervals estimate the true population mean or other quantity of interest that cannot be observed.

Tolerance Limits

Arbitrary limits established for the x-factors or subfactor so that an overall confidence or prediction interval of a process or subprocess may be realized. Tolerance limits may be established via statistical tolerance analysis by developing a statistical interval within which, with some confidence, a specified proportion of a population falls. In simpler terms, the tolerance limits may be determined from statistical tolerance interval analysis, and then used to set internally adopted specification limits for internal parameters of a process. Adhering to the tolerance limits will increase confidence in acceptable performance outcomes that depend on the process parameter associated with the tolerance limits.

Summary

If we take a step backwards and review the CMMI®-DEV v 1.2 and think about what it should contribute to our organization, we might be surprised at the simplicity of the CMMI® and how powerful its concepts can be. What do we really want for our organization? We want to have processes that our workforce follows, be able to measure those processes, and determine which processes are weak and which are strong and improve those that are weak. We want the biggest subset of our organization population we can get to have the confidence our measures or data are valid. We want to determine what our business objectives are, what measures we need for what reasons to support those business objectives, and we want those measures at both the organizational and project level.

We want to build a database of our process performance and resulting product quality so that we have an idea of what quantitative measurements we can strive for. We want to be able to predict whether we have the ability to meet our customers' requirements, keep up with our competition, or take a path to highlight our organization's strengths that will put us in a world leadership position. Statistical techniques, like those described in this chapter, help us to achieve all of this.

The SEI's Introduction to CMMI® training class on the ML 4 and ML 5 process areas sums things up quite nicely [1]:

> With defined processes, measures are collected and analyzed to understand and manage activities and results:
>
> - Thresholds are set but not using statistical and other quantitative methods
> - Exceeding thresholds triggers actions.
>
> With quantitative management
>
> - Analyses are concerned with addressing special causes of process variation;

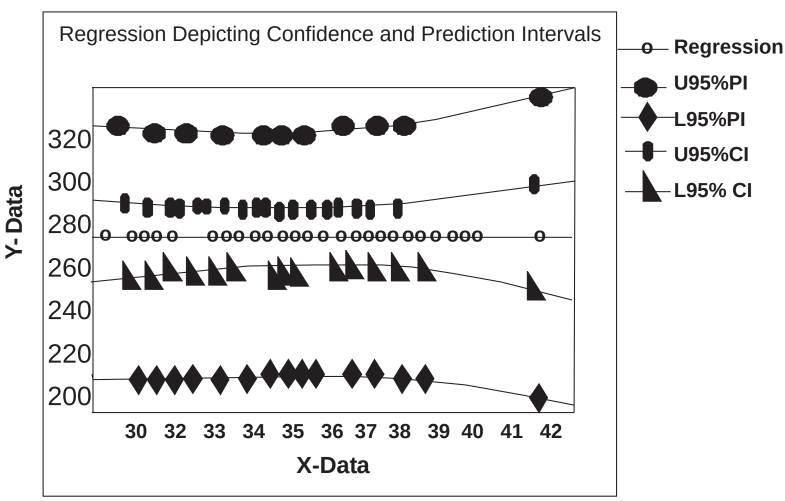


Figure 19.13 Confidence and prediction intervals. (*From:* [5]. © 2008 Software Engineering Institute. Reprinted with permission.)

- Measurements are analyzed quantitatively to:
    - Understand process performance;
    - Predict the achievement of product and service quality objectives.

With quantitatively managed processes the behavior of the process is predictable and quantitatively understood. In addition to qualitative information, a quantitative basis exists for decisions to achieve established product quality, service quality, and process-performance goals.

Chapters 20 and 21 will focus on the CMMI®'s process areas that make up ML 4 and ML 5 and expand on these ideas.

References

[1] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI® Second Edition, Guidelines for Process Integration and Product Improvement*, Reading, MA: Addison-Wesley, 2007.

[2] National Graduate School of Quality Management, February 2008; available at http://www.nationalgradschool.org.

[3] Stagliano, A. A., *Rath and Strong's Six Sigma Advanced Tools Pocket Guide,* New York: McGraw-Hill, 2004.

[4] Software Engineering Institute, "SEI Designing Products and Processes Using Six Sigma Basic Statistics," course, Pittsburgh, PA, 2007.

[5] Software Engineering Institute, "SEI Understanding High Maturity Practices: Statistical Thinking," course, Pittsburgh, PA, 2008.

CHAPTER 20

Quantitative Management

Quantitative Management Overview

The CMMI®-DEV v1.2 describes process performance and quantitative management quite well in the Introductory Notes to "Specific Goals and Specific Practices of Organizational Process Performance and Quantitative Project Management." A few of those definitions as well as those found in the CMMI®-DEV v1.2 glossary are borrowed here to inspire this chapter.

When higher degrees of quality and performance are demanded, the organization and projects must determine if they have the ability to improve the necessary processes to satisfy the increased demands. This is an important indicator of higher maturity organizations. They make decisions based on data and know their limitations due to the analysis of that data. Quantitative management *is tied to* the organization's strategic goals for product quality, service quality, and process performance. Achieving the necessary quality and process performance objectives requires *stabilizing the processes or subprocesses that contribute most to the achievement of the objectives*. Assuming the technical requirements can be met, the next decision a project or organization must make is to determine if it is cost effective. An organization could, in fact, have the technical capability to achieve higher product quality or process performance and go out of business proving it. That is not the goal of any business, nor the guidance of the CMMI®.

Process performance is a measure of the actual process results achieved when a project follows its defined processes derived from the organization's set of standard processes. Process performance is characterized or defined by both *process measures* and *product measures*. Process or subprocess measurements include effort, cycle time, and defect removal efficiency. Product measurements include reliability, usability, and defect density. Service measurements include capacity and response times.

Applying quantitative management concepts on a project utilizes statistical and other quantitative methods at both the organizational and project levels (Figure 20.1). The objective is to understand past process performance, past product quality, and past service quality and to be able to predict future process performance, future product quality, and future service quality.

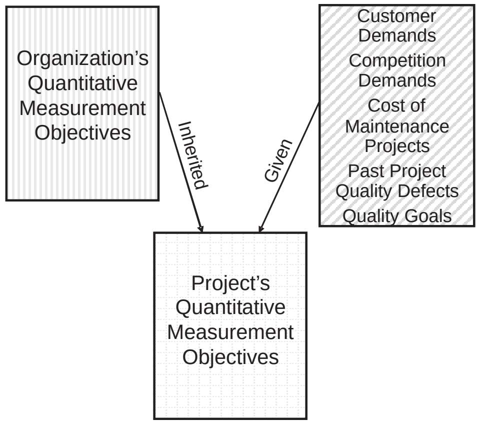


Figure 20.1 Quantitative measurement objectives.

Relevant Terminology

It is appropriate to discuss a few of the CMMI® definitions to build a more complete picture of quantitative management. See Appendix C for a more complete quantitative management vocabulary.

- *Statistical management:* Management involving statistical thinking and the correct use of a variety of statistical techniques such as control charts and confidence intervals and prediction intervals.
- *Quantitative management:* The process of using data from statistical management to help the project predict whether it will be able to achieve its quantitative, quality, and process performance objectives and, if not, determine what corrective action, if any, should be taken.
- *Predictability:* A process is said to be predictable when through the use of past experience, we can describe, at least within limits, how the process will behave in the future. An unpredictable process will display exceptional variation that is the result of special causes. A predictable process will display routine variation that is characteristic of common causes. Confidence in estimates provides the ability to predict the extent to which the project can fulfill its quality and process performance objectives.
- *Statistical control:* Statistical control supports statistical management and is based on collecting and analyzing process and product measures so that special or assignable causes of variation can be identified and addressed to achieve predictable performance.

Organizational Process Performance

An Everyday Life Example

To determine if a requirement for a higher degree of quality or performance can be achieved, it is necessary to know what the process capability of the existing organizational processes and subprocesses is across the organization's projects and then determine if it can be improved to meet the organization's business and measurement objectives.

Let us look at an example that involves swimming to provide a practical example before we delve further into the theory. I have used this example for a number of years while teaching CMMI® and ML 4 process areas. It was easy for me to understand and present this example as I have watched two sons follow most of this swimming scenario during their teenage years. Here is the scenario:

- You have a son or daughter that you placed into a kids' swimming program because you did not want them to be afraid of water and wanted them to be able to help themselves out of danger if they ever fell into a pool and perhaps even to like swimming as exercise.
- Pretty soon, you discover that they like swimming and are pretty good at it and you put them into a swimming club program to further develop their skills.
- Soon enough after that, you are encouraged to push your son or daughter into more of the club's competitive events. Of course, this requires more practice time and going to those swim meets on the weekends.
- During the next few years, your child has shown an aptitude for competitive swimming and has collected enough trophies to keep both the child and you, the parent, interested enough to keep supporting it.
- The clock advances to the time when the son or daughter is 17 or 18 years old and has won swimming events at the high school level, the state level, and even the national level and now the talk starts about world competition or even the Olympics.
- But there is something else that has happened during the past years. The coach has worked with your son or daughter and has built up a database of times that he/she has excelled in each of the swimming events. Your son or daughter has swam those events so many times, especially during the past 3 years, that for each event, the average time and the range of performance results is known within a range of five one-hundredths of a second. Based on all of the major constraints known from swimming history and a few more the coach has observed about your son or daughter, this range of performance results holds consistent for each event.
- The database of performance results also shows something more than the stability of your son or daughter's performance. It shows that the best time, much less the mean of the range of performance results, is not good enough for someone aspiring to be an Olympic champion.
- The questions of course are what can be changed, what can be done to improve the swimming performance in one or more events to succeed in becoming an Olympic champion, not just get honorable mention for participation? Your son and daughter have come so far and you know how much sacrifice you, as parents, have made as well.
- It is a known swimming fact that most swim meets are not lost in the middle of the pool. Precious seconds are lost on the dive (the start), the turn, and some-

what on the finish, when potential winners relax and find out they lost the race by that one one-hundredth of a second instead of winning it.

- The other factor that causes swimmers to lose time is the coefficient of drag. For years men shaved their heads and bodies in an attempt to reduce that friction of their hair with the water and therefore reduce the coefficient of drag. But a few years ago, we saw the introduction of a full body suit for men and women that was made of a special shark skin to help the swimmer glide a bit more easily through the water.
- So, instead of trying to look at the entire swimming process and even trying to get it under statistical control, it makes more sense to concentrate on the swimming subprocesses of diving, turning, finishing, and even searching for innovative technology in an attempt to improve their performance and therefore succeed in the goal of winning the Olympics in one or more events.

This swimming example provides us with a scenario that follows the measurement guidance CMMI® offers us from low to high maturity. It takes us from basic swimming activities or processes (life-cycle processes) to an organizational set of processes to guide a set of swimming hopefuls in a swimming club (organizational set of standard processes), to collecting measures on the organizational swimming processes for each participant (measurement and analysis), to building up a process performance database by collecting performance data for each participant within each separate event (OPP—process performance baselines), to establishing being an Olympic champion (OPP—business objective for the swimming club), to focusing on improving the swimming subprocesses to enable the swimmer's stable process to improve such that the goal of becoming an Olympic champion (QPM—project's quantitative measurement objectives) is predictable and reachable.

Process Performance

The CMMI® includes a defined process area called Organizational Process Performance that guides an organization to maintain a quantitative understanding of the performance of the organization's set of standard processes in support of quantitative quality and process performance objectives. Process performance data, baselines, and models are established and maintained to quantitatively manage the organization's projects.

First let us define process performance. The introductory notes for the Organization Process Performance process area provide excellent insight: "Process performance is a measure of the actual results achieved by following a process" [1]. Process performance is characterized by both process measures and product measures. Typical process measures include effort, cycle time, and defect removal effectiveness. Typical product measures include reliability and defect density. "The common measures for the organization are composed of process and product measures that can be used to summarize the actual performance of processes in individual projects in the organization" [1].

To support quantitative project management, the organization needs to:

- Establish and maintain quantitative objectives for quality and process performance for the organization;
- Determine which processes/subprocesses are suitable to be measured;
- Determine which measures are useful for determining process performance.

It is important to remember the specific practices do not always, if at all, represent sequential activities. As stated in the CMMI®, these specific practices are often interrelated and may need to be performed concurrently.

Quantitative Objectives for Quality and Process Performance

As the adage goes, in real estate, the three most important things to focus on are location, location, and location. In similar fashion, the three most important things to focus on when an organization focuses on measurement and especially quantitative objectives for quality and process performance are business objectives, business objectives, and business objectives.

There is no reason to even use the CMMI®, much less strive for higher levels of maturity, unless the business objectives are clearly stated, understood by everyone in the organization, and monitored on a continuous basis. Thus, the statement "The organization's quality and process-performance objectives should be based on the organization's business objectives" should not be a surprise to anyone. In addition, the organization's quality and process performance objectives should be based on other factors such as:

- Past performance of projects;
- Past quality defects of delivered products;
- Competition;
- Demands of the customers and end users;
- New technological opportunities;
- Strategic vision of the senior management.

Examples of business objectives may include the following:

- Achieve a development cycle of a specified duration for a specified release of a product.
- Achieve an average response time less than a specified duration for a specified version of a service.
- Deliver functionality of the product to a target percentage of estimated cost.
- Decrease the cost of maintenance of the products by a specified percent.
- Decrease the number of passes through system test by a targeted number while holding constant or decreasing the number of defects that escape the system test phase.
- Decrease the number of defects that find their way into production.
- Achieve a development cycle of 22 calendar months for a specified release of a product beginning June 2008 without increasing the current level of delivered defects.

- Decrease the cost of maintenance by 30% beginning in January 2009 without reducing service availability or current product usability.

Examples of quantitative quality and process performance objectives that are often established at the project level to support those business objectives include the following:

- Deliver work products with no more than a specified number of latent defects.
- Shorten time to delivery to a specified percentage of the process performance baseline.
- Reduce the total life-cycle cost of new and existing products by a percentage.
- Reduce the defect density of the next product release by a percentage compared to the previous release.
- Improve the mean time to failure of a critical product component compared to the existing reliability measure of a field product.
- Reduce the number and severity of defects of type A in the next release of a previously released product to increase product quality.
- Reduce the amount of rework time by a percentage of the total project life-cycle time by a specified percentage.
- Improve prior customer satisfaction ratings by a specified percentage compared to previous ratings.

These quantitative quality and process performance objectives may be contrasted with the quality and process performance objectives found in the Measurement and Analysis process area when the measurement foundation is being formed (see Chapter 18). Selection of the processes and/or subprocesses to be included in the organization's process performance analyses is based on the needs and objectives of both the organization and projects.

Examples of criteria that can be used for the selection of a process or subprocess for organizational analysis like the swimming scenario include the following:

- The relationship of the subprocess to key business objectives;
- Current availability of valid historical data that is relevant to the subprocess;
- The current degree of variability of this data;
- Subprocess stability.

Examples of criteria used to select measures that are to be included in the organization's process performance analysis include:

- Relationships of the measures to the organization's business objectives;
- Coverage the measures provide over the entire life of the product;
- Visibility that the measures provide into the process performance;
- Frequency at which the observations of the measures can be collected;
- Extent to which the measures represent the user's view of effective process performance.

Process Performance Baselines

Process performance baselines and models are used to:

- Analyze the process performance associated with the processes in the organization's set of standard processes to target areas for innovation;
- Assess the potential return on investment for process improvement activities (see the description of the CAR and OID process areas in Chapter 21).

Process performance baselines and models predict what kind of outcome will result from selecting particular subprocesses from the organization's set of standard processes. The use of process performance baselines and models helps to determine whether the project can achieve its objectives and assesses the risks of not doing so.

The organizational data from process and product measures is analyzed to establish a probability distribution and range of results that characterizes the expected performance for selected processes and subprocesses when used on any individual project in the organization. The expected process performance can then be used in establishing the project's quality and process performance objectives and can be used as a baseline against which actual project performance can be compared. Figures 20.2 and 20.3 provide examples of using a box and whisker plot to arrive at a process performance baseline.

It is expected that an organization will establish multiple process performance baselines to characterize performance for subgroups of the organization. These subgroups include:

- Product line;
- Application domain;
- Complexity;
- Team size;
- Work product size;
- Development or maintenance projects.

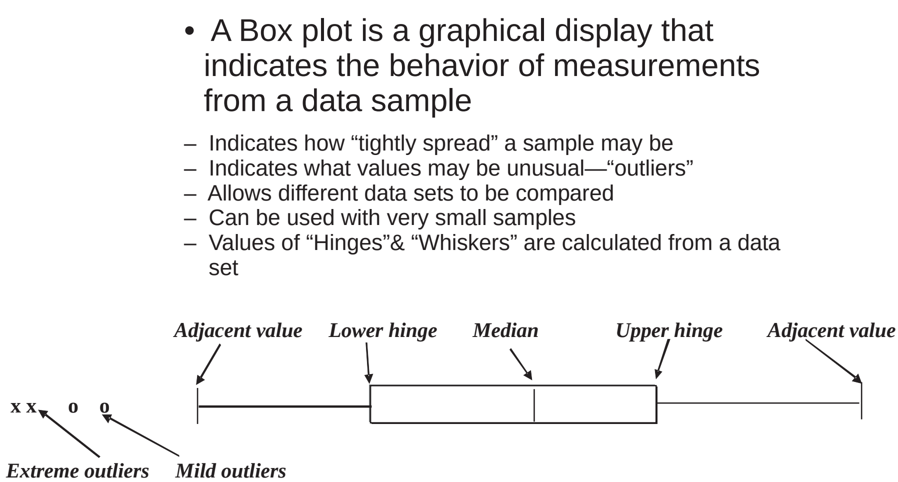


Figure 20.2 Box and whisker plot. (Courtesy of Dave Hufton, Institute of Systems Science, National University of Singapore.)

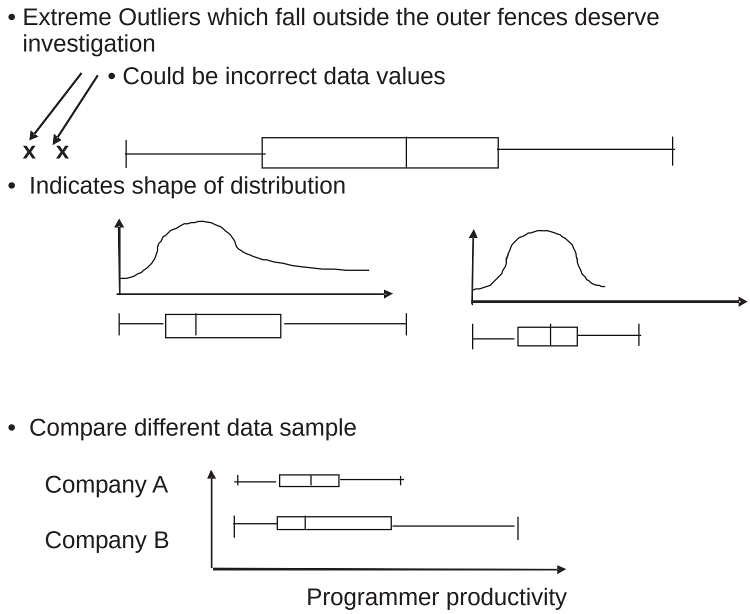


Figure 20.3 Uses of box and whisker plots. (Courtesy of Dave Hufton, Institute of Systems Science, National University of Singapore.)

When process performance baselines are being established, it is recommended that only measurements from stable processes be used. However, even data from unstable processes can be used. One only needs to recognize that using unstable processes contributes additional uncertainty. It is also recommended that the process performance baselines used be taken from time-ordered data.

One example of using statistical methods such as hypothesis testing makes itself apparent here. Hypothesis testing can be used to compare process performance baselines to determine if the organizational baseline had shifted or moved to the current baseline. In other words, an organization or project can determine whether a process change has occurred by comparing the before and after process performance baselines. Other statistical methods that support process performance baselines include central tendency and variation along with confidence and prediction intervals.

It is worthy to note here that the Measurement and Analysis process area described in Chapters 18 and 19 provides the foundation for the higher maturity practices. When we start with measurement and analysis at Maturity Level 2, we learn to establish objectives for measurement and analysis, specify the measures to be performed, obtain and analyze the measures, and analyze the data.

Process Performance Models

Process performance models relate the behavior of a process or subprocess to an outcome. They predict future outcomes (Y's) based on possible or actual changes to factors/inputs (X's). They normally use these factors or inputs from one or more subprocesses in the prediction. These process performance models typically use process and product measurements collected throughout the life of the project from the various life-cycle phases to estimate progress toward achieving objectives that cannot be measured until later in the project's life. And they are statistical in nature.

High-maturity organizations normally develop multiple process performance models that go beyond cost and schedule variance. Projects must deal with multiple variables. Organizations and projects will be required to answer different questions that will require different process performance models.

Process performance models are developed based on the organization's set of standard processes and the organization's process performance baselines.

- The results or outcomes are the big *Y*'s that may be decomposed into subordinate outcomes or little *y*'s. These represent the business and measurement objectives from the organization down to the project and from the project itself.
- The process performance models help us to think about and predict the controllable inputs or factors (*X*'s), the interaction of these inputs, and the *X*-*Y* relationship. (See Figure 20.4 for an example of *Y*-results and *X*-factors.)
- Process performance models predict future outcomes based on possible or actual changes to the inputs using "what if" analyses.
- Controllable inputs or factors include:
  - Size of work to do;
  - Process compliance;
  - Extent of unit testing;
  - Checking time for peer reviews.

Examples of Process Performance Models and Related Statistical Methods

Statistical methods that support process performance modeling are discussed next.

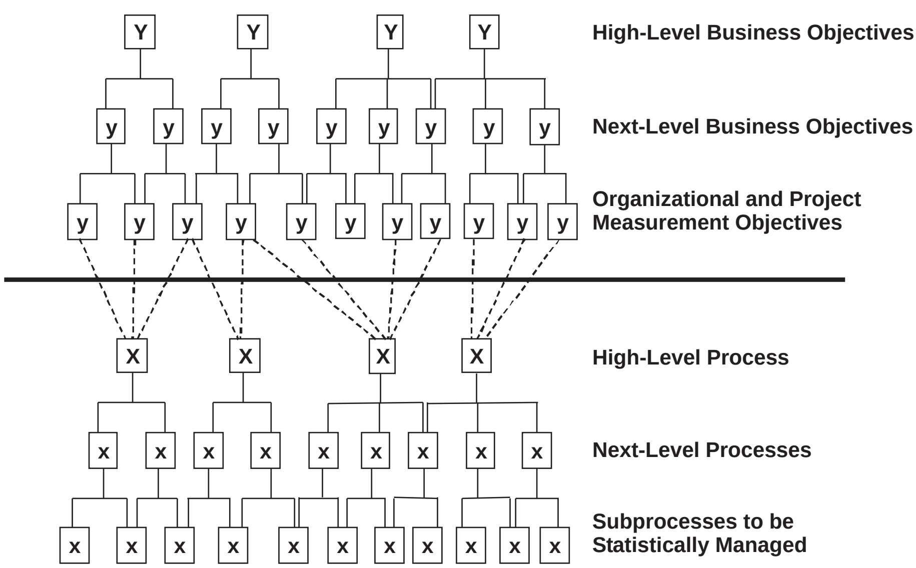


Figure 20.4 Organization and project measurement flow down to critical process and subprocesses. (Used with permission of the Software Engineering Institute.)

Hypothesis Testing

Used to evaluate actual process performance (mean and variation) relative to a standard or specification to:

- Determine if differences exist between processes;
- Verify process improvements by comparing before and after data.

As stated previously, hypothesis testing can be used to compare process performance baselines to determine if the organizational baseline had shifted or moved to the current baseline. See Chapter 19 for more details about hypothesis testing.

Analysis of Variance (ANOVA)

ANOVA is used to test for significant differences on more than two group means and estimate the 95% confidence interval of each group mean. It is used together with dummy variable regression for predictive modeling.

Chi-Square

The chi-square test is used to test for significant differences with attribute or categorical data. This test:

- Is used to verify that data fits into a particular distribution or belongs to a family of distributions;
- Enables you to see if knowledge of one discrete factor is useful in predicting a separate discrete outcome;
- Is used together with logistic regression for predictive modeling.

Process Performance Model Examples

Linear Regression Analysis

Linear regression analysis is used to define the mathematical relationship between an output variable (y) and one or more input variables (x).

- Regression models are used to predict the value of the outcome or dependent variable (y) as a function of the value of the input or independent variables (x).
- Consider this example: Reduction of defects shipped = 0.524 + 0.51 Defect density from peer reviews + 0.48 Defects from unit testing + 0.1 Defects from systems testing – Accuracy of schedule estimation.

Logistic Regression

Logistic regression is used to predict a discrete or attribute (y) outcome using either continuous or discrete (x) factors:

- Nominal;

- Ordinal;
- Binominal.

Monte Carlo Simulation

A Monte Carlo simulation allows modeling of variables that are uncertain:

- Can put in a range of values instead of a single value.
- Analyzes simultaneous effects of many different uncertain variables creating a more realistic analysis.
- Establishes confidence levels for outcomes.

Design of Experiments (DOE)

The DOE structured methodology provides us with a mechanism to observe how the output of a process is affected by specific changes that we make in the setting of the process inputs. Often used to determine the effect that two or more factors have on our process.

Process Model Simulation

Process model simulations are used to describe how things must/should/could be done instead of the process itself, which describes what really happens. Such a simulation provides gives us a rough anticipation of what the process will look like.

System Dynamics

System dynamics is an approach to understanding the behavior of complex systems over time:

- Uses internal feedback loops;
- Often utilizes simulation to study the behavior of systems and the impact of alternative policies.

Probabilistic Networks

The joint probability distribution of the variables is specified by a probabilistic network's graph structure. Bayesian networks are examples of probabilistic graphical models.

Reliability Growth Models

Reliability growth models account for corrective actions in order to estimate current and future reliability and other metrics of interest.

Neither the CMMI® nor many of the other related references make a clear distinction between quantitative models or techniques and process performance models. Figure 20.5 was developed by Bob Stoddard of the Software Engineering Institute during an Understanding CMMI® High Maturity Practices training class and helps put these concepts into perspective.

Summary for Organizational Process Performance

The best summary for our discussion on OPP comes from the notes that provide the high-level description for SP 1.5, Establish Process Performance Models. Process performance models provide the ability to predict critical process and product characteristics that are relevant to business and measurement objectives. Process performance models are used by both the organization and the projects as follows:

Organization

- For estimating, analyzing, and predicting the process performance of the processes in the organization's set of standard processes;
- To assess the potential return on investment for process improvement activities.

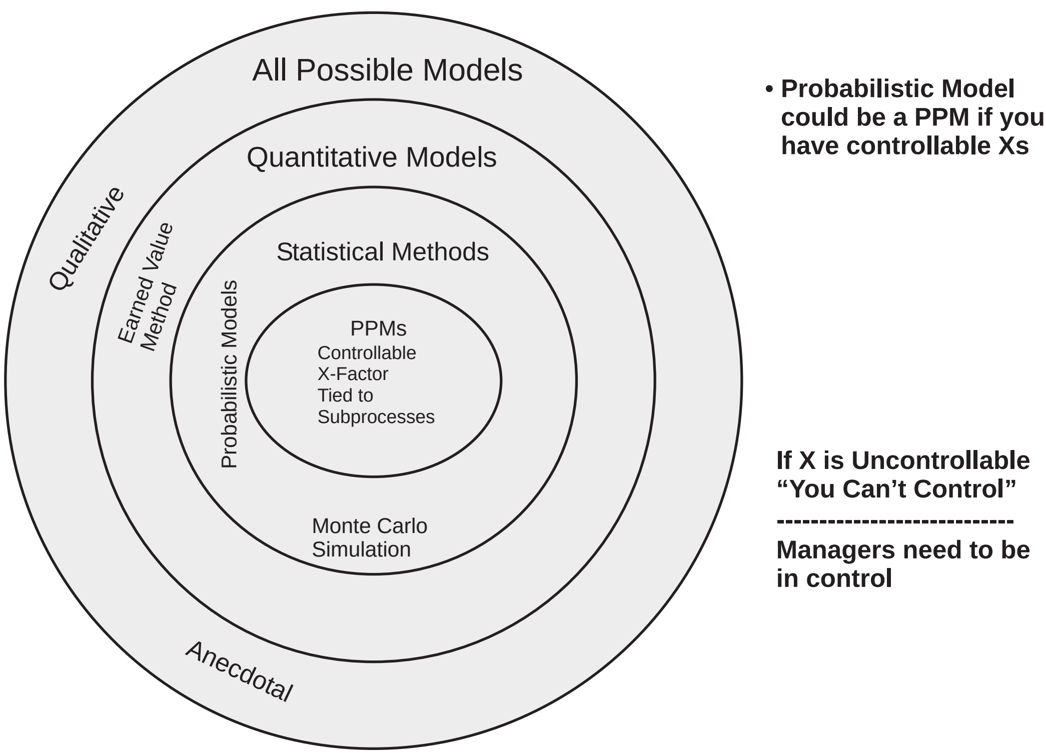


Figure 20.5 Qualitative and quantitative models to process performance models. (Courtesy of Bob Stoddard, Software Engineering Institute. Used with permission of the Software Engineering Institute.)

Project

- For estimating, analyzing, and predicting the process performance for the project's defined process;
- For selecting subprocesses for use.

Process performance models are statistical or probabilistic and can account for statistical variation. High-maturity organizations generally must develop a set of process performance models. Each organization and project will have different questions that will require different models for the answers.

Quantitative Project Management

What is quantitative project management? It still has something to do with project management but the descriptive word of *quantitative* is now the focus. From the SEI's Intro to CMMI® Workshop, we get a first level understanding.

Quantitative project management involves the use of statistical and other quantitative methods at the organizational and project levels to:

- Understand past process performance, past product quality, and past service quality;
- Predict future process performance, future product quality, and future service quality.

Projects use measurable objectives to meet the needs of customers, end users, and the organization. Managers and engineers use the data with statistical and other quantitative techniques in managing the processes and results.

Statistical management involves statistical thinking and the correct usage of a variety of statistical process control tools such as control charts, confidence intervals, prediction intervals, and tests of hypotheses for comparison of results. Quantitative management, then, uses data from statistical management to help the project predict whether it will be able to achieve its quality and process performance objectives and identify when corrective action should be taken.

Process performance is a measure of the actual process results achieved and is characterized by both process measures and product measures as indicated earlier. Organizations are encouraged to use statistical management to have the ability to predict the extent to which their projects can fulfill their quality and process performance objectives.

Quantitative project management (QPM) is project oriented to complement organizational process performance (OPP), which is organization oriented. The essence of QPM is to manage the project's defined process to achieve the project's established quality and process performance objectives. This constitutes its quantitative measurement objectives for quality and process performance. QPM provides an understanding of the variation expected in process performance. (See Chapter 22 for a generic discussion about reducing variations at all maturity levels of the

CMMI®.) The quality and process performance objectives, measures, and baselines developed for the organization as described by OPP can be used to help projects:

- Establish their quantitative measurement objectives.
- Analyze and compare their process performance.

Projects within high-maturity organizations have the following:

- Processes that are tailored from the organization's set of standard processes;
- Individuals who use data and are at least educated users of the results of common statistical analysis;
- The capability to measure, track, and report performance against organizational objectives;
- Support from the organization's quantitative measurement objectives, process performance baselines, and process performance models for managing quality and process performance.

Project's Quality and Process Performance Objectives

The project's quality and process performance objectives come from both the organization as well as the customer, end users, and relevant stakeholders (refer to Figure 20.1 earlier in this chapter). Figure 20.6 shows the relationship among the organization's and project's business and measurement objectives and the quality and process attributes, measures, and quantitative objectives for improvement.

Candidate attributes and measures to support business, organizational unit, and project needs include:

- Functionality;
- Reliability;
- Maintainability and expandability;
- Security and safety;
- Usability;
- Predictability;
- Development cycle time;
- Accuracy.

Based on the concept provided by Figure 20.6, product quality measures for which we could define quantitative objectives would include:

- Critical resource utilization;
- Number, category, and severity of defects in the released product;
- Number, category, and severity of customer complaints;
- Reduced percentage on time to delivery;
- Reduced total life cycle on cost;
- Mean time to failure.

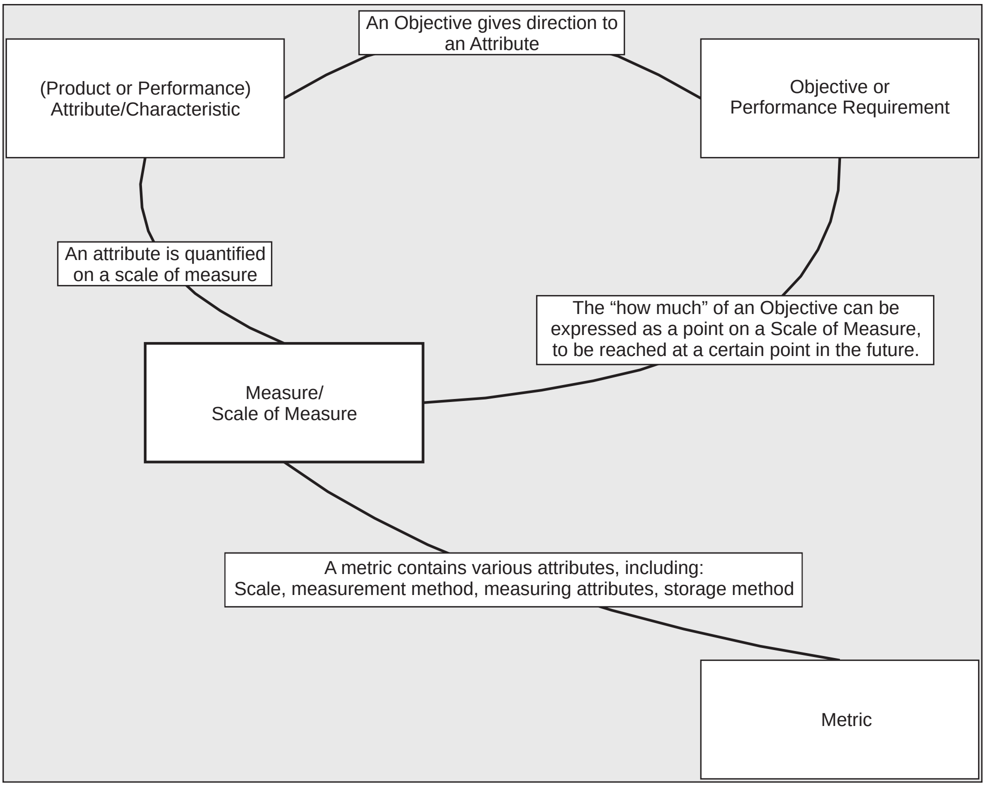


Figure 20.6 Attribute, scale, objective. (Courtesy of Simon Porro and Tom Gilb. Reprinted with permission.)

Process performance measures for which we could define quantitative objective would include:

- Earned value;
- Cycle time;
- Percentage of defects removed by type of verification activity;
- Peer review coverage;
- Test coverage;
- Peer review effectiveness;
- Test effectiveness;
- Defect escape rates;
- Defect removal efficiency;
- Number and density of defects (by severity) found during the first year following product delivery;
- Rework time as a percentage of total project life-cycle time.

In addition to the organization’s and project’s quantitative measurement objectives, it is important to select “interim measurement objectives” for each life-cycle phase as appropriate to assist in monitoring the project’s quantitative objectives, as defined earlier. Process performance models can be used to focus the project on which interim measurement objectives (*X*'s) would lead to desired outcomes (*Y*'s).

Project's Defined Process

The ability to create the "project's defined process" is the *gate* to all that is Maturity Level 4 and Maturity Level 5 or Capability Level 4 and Capability Level 5, for that matter. With the process assets being defined at the organizational level, a project operating in a Maturity Level 3 organization uses the guidance provided by the Integrated Project Management (IPM) process area. (See Chapter 10 for a detailed description of IPM and its similarities and contrasts to project planning and project monitoring and control.)

It is the combination of OPD and IPM that supports the creation of the project's defined process, as most readers well know. But let us look at a few of the concepts provided in IPM. Specific Goal 1 states "The project is conducted using a defined process that is tailored from the organization's set of standard processes." The project's defined process consists of defined processes that must form an integrated, coherent life cycle for the project. It must include those processes necessary to acquire or develop and maintain the product. It should satisfy the project's contractual and operational needs, opportunities, and constraints. It should be based on:

- Customer requirements;
- Product and product component requirements;
- Commitments;
- Organizational process needs and objectives;
- Organizational set of standard processes and tailoring guidelines;
- Operational environment;
- Business environment.

In addition, the standard processes selected should be those that "best fit" the needs of the project. These requirements set down in IPM are exactly the same as those needed for quantitative project management.

The processes and—most importantly for quantitative management—the subprocesses that we have identified as those we would like to statistically manage due to their contributions to our project's and organization's quantitative business and measurement objectives must come from the processes we select and tailor from the organization's set of standard processes. Of course, we must choose the most suitable processes; the ones that are the most appropriate to statistically manage. In other words, which controllable factors/inputs (*X*'s) could lead or help drive the project to the desired outcomes or results (*Y*'s). The reader may want to reexamine Figure 20.4 at this point. It is also important to remember that processes and subprocesses *interact with each other*. Examining the interaction of the critical subprocesses is equally important to understanding their relationships and how those interactions might affect or impact the processes or project downstream. Statistical techniques that support understanding the effects of subprocess interaction include process model simulation and Monte Carlo simulation.

Another approach to developing the project's defined process is to make use of the process performance baselines and process performance models. High-maturity organizations have found that these can assist coming up with process trade-offs such as choosing a formal inspection process compared to an informal buddy check. We all know that even if 100% inspection were possible, we would still not be able to be 100% confident in our results. Process performance models may guide a project to the appropriate sample size to help meet the customer's requirements with a 90% to 95% confidence level.

Which Subprocesses and How Many Will Be Statistically Managed?

The preceding discussion on a project's defined process included criteria for selecting the appropriate and suitable subprocesses. Here is a more complete list captured in one place of criteria for selecting subprocesses:

- Customer requirements related to quality and process performance;
- Quality and process performance objectives established by the customer;
- Quality and process performance objectives established by the organization;
- Stable performance of the subprocess on other projects;
- Organization's performance baselines and models;
- Processes that have the greatest sources of variation;
- Analysis of process interaction;
- Major contributors to the achievement of the project's quantitative measurement objectives. *(Remember the Olympic swimming example and the subprocesses of the dive and turn that would contribute the most toward a performance needed to win the Olympics.)*

So the most frequently asked question remains: How many subprocesses need to be statistically managed for an organization to achieve Maturity Level 4? For that answer, I invite you to read the hint provided by the CMMI® book authors that is next to Specific Goal 1.3 on page 447: "…An answer is to select at least one 'main contributor' subprocess per project lifecycle phase, at least one project management subprocess, and at least one support subprocess" [1]. Hints like these, the introductory material, the notes, and the subpractices that define the process areas at Maturity Level 4 and Maturity Level 5 are so very important for anyone to truly understand what high maturity, according to the CMMI®, really means. A presentation by Rusty Young of the SEI produced the following quote, which sums up the intent of the introductory material, notes and subpractices: "This material is informative, not *ignorative*." These high-maturity process areas are not about what is the minimum my organization can do to get a ML 4 or ML 5 certificate; they are all about describing what an organization must understand and have in place to answer the highest demands from its customers and competitors with quantitative management.

Of course, it should be recognized that determining the project's quantitative measurement objectives, that is, composing the project's defined process with the properly selected or suitable subprocess, is a nonstop, iterative process that continues throughout a project's life.

Managing the Project's Quantitative Objectives for Quality and Process Performance

It is in this section where we see the organizational, project, and statistical factors come together, as summarized here by the bulleted points:

- Organization's set of standard processes along with its critically identified subprocesses;
- Interim measurement objective results per life-cycle phase;
- Organization's quantitative measurement objectives;
- Process performance baselines;
- Process performance models;
- Project's quantitative measurement objectives for quality and process performance measures;
- Project's defined process;
- Subprocesses identified to be statistically managed that have been identified as contributors or predictors to achieving the project's quality and process performance objectives;
- Attributes of subprocesses;
- Subprocesses measures;
- Special and common causes of variation in selected processes and subprocesses;
- The statistical analysis techniques available and expected to be useful;
- Visualization techniques and tools to support quantitative management needs.

Additional Ideas for Quantitatively Managing the Project's Objectives for Quality and Process Performance

The following points are taken from related QPM practices, subpractices, and notes to help the reader in understanding what is needed to perform effective quantitative project management:

- Periodically review the performance of each subprocess and the capability of each subprocess selected to be statistically managed to determine progress against the project's quality and process performance objectives.
- Periodically review the actual results achieved against the interim objectives for each phase of the life-cycle phase to determine progress against the project's quality and process performance objectives.
- Use process performance models filled in with obtained measures of critical attributes to estimate progress toward achieving the project's quality and process performance objectives.
- Be prepared to use more than one measure to characterize the performance of the selected subprocess and make trade-offs. Examples of subprocess measures can be found in the CMMI® within the description of SP 2.1:
    - Requirements volatility;

    - Ratios of estimated to measured values of the planning parameters (e.g., size, cost, and schedule);
    - Coverage and efficiency of peer reviews;
    - Test coverage and efficiency;
    - Effectiveness of training (e.g., percent of planned training completed and test scores);
    - Reliability;
    - Percentage of total defects inserted or found in the different phases of the project life cycle;
    - Percentage of total effort expended in the different phases of the project life cycle;
    - Percentage of rework time compared to the total time expended on the project so far.
- Compare the project's quality and process performance objectives to the natural bounds of the measured attribute. See the discussion on variation and natural bounds in the section titled, "Understanding Variation Using Statistical Techniques."
- Use histograms to determine if the selected subprocess is capable of meeting the customer's requirements or specification limits.
- Use process performance models to predict how subprocesses downstream may be affected to get a feeling for what is allowable in terms of variation and back into the variation of the subprocesses in the upstream life-cycle phases.
- Use control charts to get a visual on subprocess improvements.
- Record the quantitative data along with its context to ensure the continuous maintenance of the organization's process performance baselines and models.

Understanding Variation Using Statistical Techniques

In order for a project to be managed quantitatively, it must be able to predict its critical subprocess performance. This in turn implies that the subprocess is "stable" or has removed all special causes of variation. This section examines the critical subject of variation in closer detail.

Any necessary corrective action is based on understanding the nature and extent of the variation experienced in actual process performance and recognizing when the project's actual performance may not be adequate to achieve the project's desired quality and process performance. Reducing process variation is an important aspect to quantitative management. Some variation is routine, run-of-the-mill variation and is to be expected even when the process has not changed. Other variation is exceptional, outside the bounds of routine, and therefore to be interpreted as a sign of a process change.

To distinguish or separate variation into these two components, Dr. Walter Shewhart created the control chart. The control chart illustrated in Figure 20.7 is created by plotting data in a time series. A central line is added as a visual reference for detecting shifts or trends, and limits are computed from the data. These limits are placed on either side of the central line at the distance that will allow them to filter out virtually all of the routine variation. The key to effectiveness of the control chart is contained in the way in which the limits are computed from the data.

Dr. Donald Wheeler describes these calculations in a clear manner in his book *Understanding Variation—The Key to Managing Chaos* [2]. By characterizing the extent of routine variation, the limits on a control chart allow you to differentiate between routine variation and exceptional variation. If, over a reasonably long period of time, all of the points fall within the limits of a control or process behavior chart, then the process can be said to display nothing but *routine* or *common cause variation*. When this happens, the process can be thought of as being *predictable* within those limits, and it is reasonable to expect that, unless something is changed, it will continue to operate that way in the future. Figure 20.7 illustrates a stable process with routine or common causes of variation.

Thus, the limits of a control chart allow you to characterize the behavior of your process as predictable or unpredictable and define how much routine variation you should expect in the future. However, when the points fall outside of the limits of the control chart, they are interpreted as signs of exceptional or assignable causes of variation, as illustrated in Figure 20.8.

Understanding variation is a fundamental premise of statistical control. Understanding variation is achieved through the collection and analysis of process and product measures so that special causes of variation can be identified and addressed to achieve predictable performance.

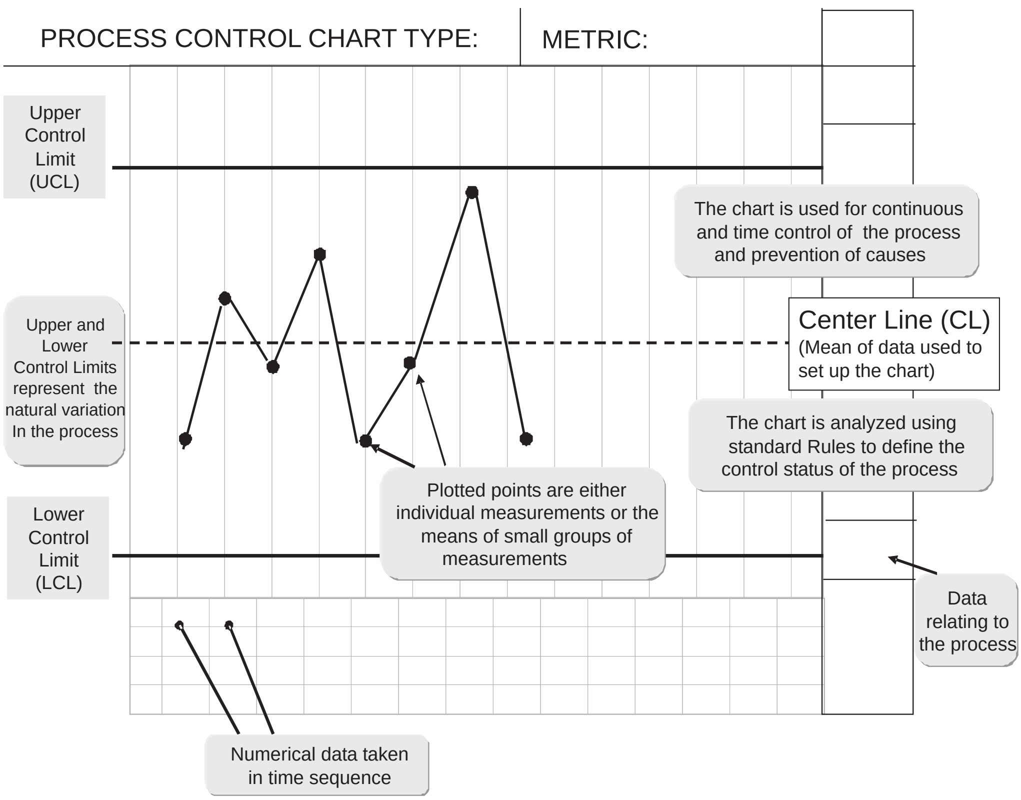


Figure 20.7 Control chart—common cause of variation. (*After:* [3].)

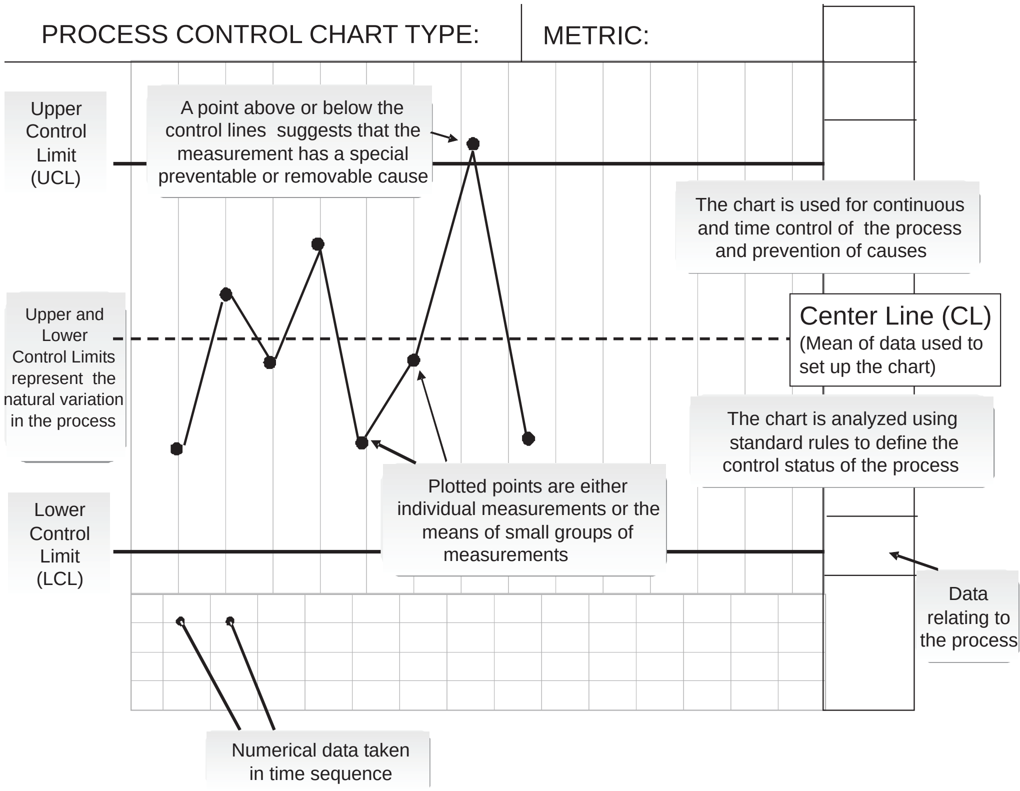


Figure 20.8 Control chart—special cause of variation. (*After*: [3].)

Wheeler used the term *routine variation* instead of exceptional variation. Routine or "common" causes of variation are variations in process performance due to the normal interaction among the process components (people, machines, material, environment, and methods). Variation due to common cause is random but will vary within predictable bounds. Special or assignable causes of variation arise from events that are not part of the normal process. These special causes of variation represent sudden or persistent abnormal changes due to one or more of the process components. The special cause may manifest itself as a faulty input to the process environment or through the way the processes themselves are executed. Examples of assignable causes of variation include inadequately trained people, tool failures, and failures to follow the process.

Assignable Cause and Exceptional Cause

Assignable causes of variation dominate the many common causes of routine variation; thus, it is worthwhile to try to identify the assignable cause of variation so its effect on your process can be eliminated. It is worthwhile assuming that if a project manager or project member knows about an assignable cause of variation, the person would not allow it to dominate the process. Thus, control charts allows a project to learn about cause-and-effect relationships that may have not been realized in the past and to help the project to realize a more predictable and reliable process.

It is important to remember to focus on subprocesses that can be controlled to achieve a predictable performance as well as the overarching processes.

Subprocesses should be selected from the process elements in the organization's set of standard processes. It is helpful to know if the subprocesses showed stable performance in previous comparable instances or if the performance data for that subprocess showed satisfaction of the project's quality and process performance objectives. It is also useful to analyze the interaction of the subprocesses to understand the relationships that exist among the subprocesses. This can certainly have an effect on your ability to bring them under statistical control.

Remember: It may not be possible to statistically manage some processes nor economically justifiable to apply statistical techniques to certain subprocesses.

When the selected subprocesses are brought under statistical control, their capability of achieving quality and process performance objectives can be determined. It is, therefore, also possible to predict whether the project will be able to achieve its objectives.

Statistical process control is then based on collecting and analyzing process and product measures so that the special or assignable causes of variation can be identified and addressed in order to achieve predictable performance.

Trial Bounds

So how do we obtain the initial control limits or bounds for our subprocesses? The subpractices of SP 2.2 in QPM give us some hints. Trial bounds may be established if suitable data is available from prior instances of the subprocess or subprocesses.

When a subprocess is initially executed, suitable data for establishing trial natural bounds are sometimes available from prior instances of the subprocess, comparable subprocesses, process performance baselines, or process performance models. These trial bounds may need to be recalculated when:

- There are incremental improvements to the subprocess.
- New tools are deployed for the subprocess.
- A new subprocess is deployed.
- The collected measures suggest that the subprocess mean has permanently shifted or the subprocess variation has permanently changed.

Natural bounds of process performance for each measured attribute should be calculated using techniques such as:

- Control charts;
- Confidence intervals (for parameters of distributions);
- Prediction intervals (for future outcomes);
- A requirement to understand process variation in the selected subprocesses.

Voice of the Process—Voice of the Customer

Before you can improve any system or system process, you must listen to the voice of the process defined by the natural bounds and variation within those bounds of process performance. You must be able to change the inputs in order to achieve the desired results.

However, comparing numbers to specifications will not lead to the improvement of the process. You must also listen to the voice of the customer. This means being able to achieve the goals established for the product and process performance such as:

- Product specifications:
  - Amount of downtime;
  - Mean time to failure;
  - Response time;
- Management specifications:
  - Meeting the schedule;
  - Meeting the budget.

Capable Processes

From a statistical control point of view, a capable process is one that satisfies both the voice of the process and the voice of the customer (Figure 20.9). The focus of this description of quantitative management has mainly been on the control chart, but many other continuous improvement and quantitative project management tools, sometimes called visualization tools and techniques, are available:

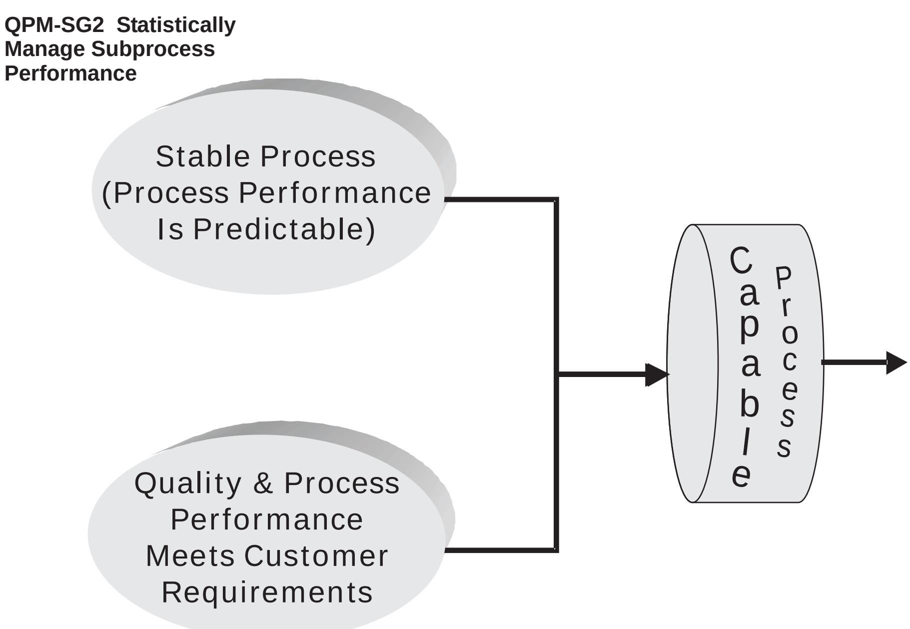


Figure 20.9 Capable process.

- Cause and effect (fishbone diagrams);
- Pareto analysis;
- Scatter diagrams;
- Run charts;
- Interrelationship diagraphs;
- Check sheets;
- Histograms.

Cause-and-effect or fishbone diagrams, Pareto analysis charts, scatter diagrams, run charts, and interrelationship diagraphs are examples of data visualization techniques or tools that are used to help projects organize and summarize data, look for patterns and trends, and determine cause-and-effect relationships. (Some of these techniques are described in more detail in Chapter 22.)

Two quantitative management techniques are most useful for quantifying process behavior (voice of the process) and in answering the question "Is the process capable of meeting my customer's requirements (voice of the customer)?" These two techniques are the control chart and the histogram.

Control Charts

As mentioned earlier, control charts:

- Allow a project team to monitor, control, and improve process performance over time by studying variation and its source;
- Distinguish special or assignable causes of variation from common causes of variation as a guide for management decision making;
- Serve as a tool for ongoing control of a process;
- Help improve a process to perform consistently and predictably for higher quality, lower cost, and higher effective capacity;
- Are most effective when they are used within the broader context of established goals and the activities performed to achieve those goals.

Histograms

Histograms are displays of observed distributions. Histograms help in many ways:

- Allow a project team to take measurement data and display the distribution of the observed values;
- Show the frequencies of events that have occurred in ways that make it easy to compare distributions and see central tendencies;
- Illustrate quickly the underlying distribution of the data;
- Help indicate if there has been a change in the process;
- Are used to characterize the observed values of almost any product or process attribute:

  - Module size;
  - Defect repair time;
  - Time between failures;
  - Defects found per test or inspection;
- Provide useful information for predicting future performance of the process;
- Help answer the question "Is the process capable of meeting my customers' requirements?"

Histograms are created by grouping the results into cells and then counting the number in each cell (Figure 20.10). Figure 20.11 shows histograms that illustrate capable processes.

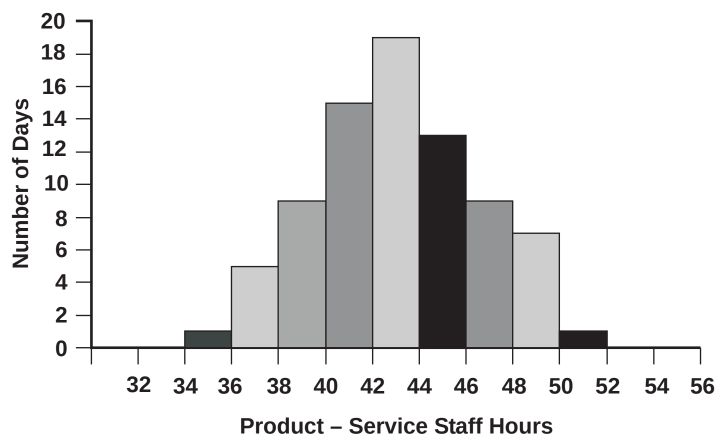


Figure 20.10 Histograms.

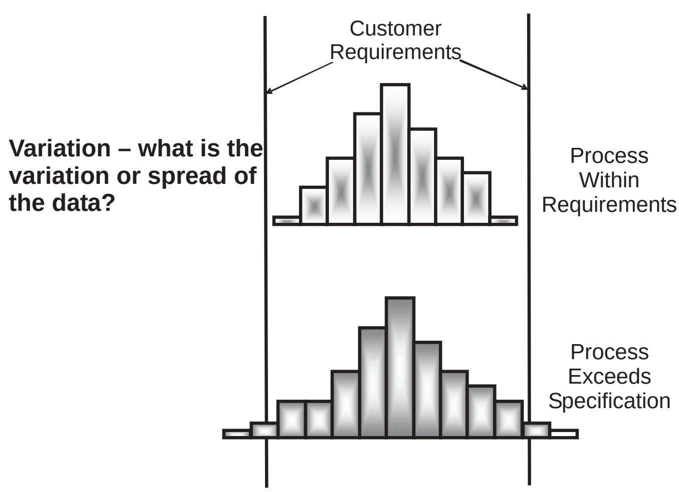


Figure 20.11 Histograms showing capable processes. (*After:* [4].)

Summary

Projects use measurable objectives to meet the needs of customers, end users, and the organization. Managers and engineers use the data with statistical and other quantitative techniques in managing the processes and results.

If we combine some of the higher maturity ideas with the project management, quality management, engineering, and process management ideas found among the process areas of the CMMI® at Maturity Level 2 and Maturity Level 3, the CMMI® can be seen as a simple, but elegant model that helps organizations and their projects to position themselves to answer the current and future demands of customers, competitors, and the organization itself.

The foundation for all things "high maturity" is basic. We need to be able to manage and control our projects if we have any hope of delivering products and product components on time and within budget and with any sort of recognized measurable quality. The basic concepts of project management and quality management at Maturity Level 2 help us accomplish this.

Of course, we want to have our engineering practices integrated with our project management practices. But more importantly, we would like to take advantage of "economy of scale" mentioned many times in this book. Our companies are in business, and most of us are in business to make money and to continue to grow into new and rewarding areas. We would like to have a common vocabulary that the organization's set of standard processes helps us to achieve. We would like to have each project follow a tailored version of the organization's set of standard processes because if we establish measures for these processes, determine some of them are not working so well and improve them, then we would like to have all or most of the projects in the organization be able to take advantage of our improvement efforts.

All of this effort expended to achieve a Maturity Level 3 rating is not just to obtain a CMMI® certificate. It is about putting our organization into a position to meet the challenges of our customers and competitors. It is about being able to dream about challenging ourselves to do better and provide the leadership in our competitive environment so when our customers do ask us to reduce the defect density of products we have delivered or achieve a higher degree of reliability, we will have the quantitative capability to understand the problem, and can develop a plan to quantitatively manage our project to reach these new and quantitatively meaningful goals.

A simple scenario follows that illustrates how the process areas of Maturity Level 4 and Maturity Level 5 can help any organization to achieve a superior position in its respective industry:

- An organization is asked to reduce the defect density of its product releases from a particular product line by its customer.
- It examines the process performance baselines for that product line and determines what the capability of the organization's current processes and subprocesses is (OPP).
- It reviews the organization's quantitative measurement objectives for quality and process performance (OPP).
- It compares this with the customer's requirements (QPM).

- It determines it is not able to achieve the customer's requirements today with the current processes and technology (QPM).
- It asks the organization for help to find any innovative ideas either from a technological or process point of view (OID).
- It is asked, "Based on the organization's quantitative business objectives and the customer's requirements, what processes and/or subprocesses must be improved and quantitatively managed (QPM)?"
- Based on developed criteria, it identifies the critical subprocesses (OPP and QPM).
- It develops its project's defined process from the organization's set of standard processes with tailoring guidelines and ensures that the processes that contain the critical processes are included (IPM and OPD). It uses process performance models to help develop the project's defined process by examining the effects of upstream subprocess on downstream processes.
- It realizes that in order for the organization to help it with new technology or improved processes, it must ensure any subprocess in question is stable and predictable. Otherwise, any application of a new technology would not be able to be measured on its own merit (QPM).
- It finds out that one of the subprocesses in question exhibits special causes of variation (QPM).
- It uses techniques described in causal analysis and resolution to help find the cause of the special variation (CAR).
- It sees that the subprocess is stable (QPM).
- It determines through managing the project performance and the use of process performance models that it still cannot reach the customer's specifications.
- It asks again for the innovative technological support but this time with stabilized processes (OID, QPM).
- A state-of-the-art technology is found and it is determined that it could be cost effective if it could be used on more than one project (OID).
- The innovative technology is piloted and found to be effective (OID, OPP, QPM).
- The project monitors its stable processes and determines that the new technology does not have a negative impact (QPM).
- The project determines that the customer's specifications will be met through the use of quantitative techniques such as histograms and process performance modeling (OPP, QPM).
- The organization determines this process improvement could be useful for other projects throughout the organization and decides to update the organization's set of standard processes (OPF, OPD).

The intent of this chapter was to use the concepts found in the process areas of OPP and QPM in the CMMI®-DEV v1.2 model, the high-maturity training courses offered by the SEI, and the insights offered by the SEI instructors. It was not intended to be revolutionary or even evolutionary. All of the quantitative management principles are not here. All of the statistical techniques and process performance models that an organization might need are not mentioned here. It is a good starting place to compare your current understanding of the CMMI® at high maturity and to follow up in the areas where you require or desire more knowledge.

Many sources of information are available for finding out more about statistics, visualization techniques, and quantitative management including Six Sigma Green Belt and Black Belt techniques and tools. I certainly recommend attending the new and revised high-maturity classes now offered by the Software Engineering Institute. I have attended all of the ones mentioned here and some of them more than once. I appreciate the big step the SEI has taken in little over 1 year after being silent for so many years in the past. SEI courses related to high maturity include:

- Improving Process Performance and Processes Using Six Sigma;
- Designing Product and Processes Using Six Sigma;
- Understanding CMMI® High Maturity Practices.

References

[1] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI® Second Edition, Guidelines for Process Integration and Product Improvement*, Reading, MA: Addison-Wesley, 2007.

[2] Wheeler, D. J., *Understanding Variation: The Key to Managing Chaos,* Knoxville, TN: SPC Press, 2000.

[3] Burr, A., and M. Owen, *Statistical Methods for Software Quality*, Boston, MA: International Thomson Computer Press, 1996.

[4] Brasard, M., and D. Ritter, *The Memory Jogger II: A Pocket Guide of Tools for Continuous Improvement and Effective Planning,* Salem, NH: GOAL/QPC, 1994.

CHAPTER 21

Beyond Stability

There is a distinction between achieving an organizational process capability at Maturity Level 4 and performing the prescribed activities to achieve an organizational process capability at Maturity Level 5. However, for many process improvement professionals, the distinction between Maturity Level 4 and Maturity Level 5 is a very fine shade of gray. Reducing the variation of a process or subprocess by eliminating the special causes of variation and stabilizing the process is only the first step in attempting to achieve the quantitative process performance and/or quality objective. The second and most important step is to find an incremental or innovative or improvement solution that will actually result in producing a product that meets the expectations or requirements/specifications of the customer. In Chapter 20, this was called the "voice of the customer." The approach to stabilizing the process within its natural bounds was referred to as the "voice of the process." Chapter 20 showed that *both* are needed to have a capable process. The innovations or incremental improvements needed to achieve customer requirements are explored further in this chapter.

The CMMI® provides us with a number of related definitions to describe the line of demarcation between the quantitative management level (capability level or maturity level) and the optimizing level. However, this chapter will show that the interrelationships are very strong among the process areas of Organizational Process Performance, Quantitative Project Management, Causal Analysis and Resolution, and Organizational Innovation and Deployment (OID). In fact, the practices of Quantitative Project Management (QPM) cannot be satisfied without the support of Causal Analysis and Resolution and typically Organizational Innovation and Deployment. QPM will frequently refer to the process performance baselines and process performance models described in OPP in Chapter 20, which can also be used in support of Causal Analysis and Resolution and Organizational Innovation and Deployment.

Let us examine a few of the definitions and descriptions of the Maturity Level 5 or optimizing process concepts [1]:

- An optimizing process is a quantitatively managed process that is improved based on an understanding of the common causes of variation inherent in the process. The focus of an optimizing process is on continually improving the range of process performance through both incremental and innovative improvements.
- Reaching Maturity Level 4 assumes that processes or subprocesses within a process area have been designated as key business drivers that the organization wants to manage using quantitative and statistical techniques; in contrast, reaching

Maturity Level 5 assumes that you have stabilized the selected subprocesses and that you want to reduce the common causes of variation within that process. "Quantitative process improvement objectives for the organization are established, continually revised to reflect changing business objectives, and used as criteria in managing process improvement. . . . Both the defined processes and the organization's set of standard processes are targets of measurable improvement activities.

- Although processes may produce predictable results, the results may be insufficient to achieve the established objectives. At Maturity Level 5, the organization is concerned with addressing common causes of process variation and changing the process (to shift the mean of the process performance or reduce the inherent process variation experienced) to improve process performance and to achieve the established quantitative process improvement objectives.

The best value an organization can receive by attempting to achieve the objectives described by Maturity Level 5 comes when there is a quantitative and statistical management base, as described in Chapter 20. Objective data is used to continuously manage processes and subprocesses by combining past experiences captured in the organization process performance baselines, and using current experiences to predict the process behavior in the future using process performance models and determining whether the predicted results will match the quantitative quality and process performance objectives for the project and organization.

This chapter describes the causal analysis and process innovations that can be built on the quantitative and predictable knowledge of an organization's processes to solve business needs that otherwise could not be solved simply through hard work and management concern.

Causal Analysis and Resolution

Causal analysis and resolution (CAR) is the process of improving quality and productivity by preventing the introduction of defects into a product. CAR is applicable to both the projects and the organization.

Relying on activities such as peer reviews and testing *to detect and eliminate the defects* after they have been introduced into the system is not cost effective. Based on an understanding of the defined process in use and how it is implemented, the root causes of the defects and the future implications of the defects can be determined.

Causal analysis can also be used *to solve problems unrelated to defects*. For example, causal analysis can be used to improve quality attributes such as usability and reliability. Another example would be to use causal analysis for improving cycle time if it were shown to be inadequate. Implementation proposals, engineering analysis, and process performance models such as process model simulation can be used to support such casual analysis.

CAR helps projects and the organization to understand how to select and analyze defects and problems, identify where those defects were injected into the life cycle, determine their causes, and implement preventive actions to prevent them from reoccurring. Process performance models may be useful in supporting these objectives. What will be seen in Maturity Level 5 organizations is a cultural shift toward a more data-driven mind-set for managers and practitioners.

A question that is frequently asked is "What types of things or events would trigger or motivate the use of causal analysis techniques?" The tips and hints provided in the CMMI® book give us a start on the answer to this question. CAR is triggered by actual defects or problems that have been reported [1]. This suggests that the effective use of CAR can only happen if defect and problem data are routinely recorded and causal analysis meetings are strongly supported by management at all levels. If business and measurement objectives are not first and foremost understood and tracked by management, the commitment to techniques such as causal analysis may be lower than required.

To have an effective process for finding the root cause of a defect so that it can be removed, it is important to gather relevant defect data from multiple sources including:

- Project management problem reports requiring corrective action;
- Defects reported by the customer;
- Defects reported by the end user;
- Defects found in peer reviews and testing;
- Defects found by quality and process performance measures;
- Process capability problems that have been identified.

To select defects to analyze further, other issues might be taken into consideration such as:

- The frequency of occurrence;
- The similarity among defects;
- Cost of analysis;
- Time and resources needed;
- Safety considerations.

Visualization and statistical techniques such as Pareto charts, histograms, and process capability analysis are examples of techniques that may be useful in selecting defects for causal analysis. A histogram may show which subprocess is the potential problem source. Interrelationships of subprocesses may also be useful in selecting the most useful defects or problems for causal analysis. Analysis of the organization's process performance baselines and the use of process performance models may also help identify potential sources of problems.

Quantitative Project Management Techniques for Causal Analysis

There are many continuous improvement and quantitative project management tools and techniques, as described in Chapter 19. A formal decision analysis approach may be useful in determining which tool or technique to use. Examples of methods for determining causes and other relationships that exist among critical issues include:

- Cause-and-effect (fishbone) diagrams;
- Pareto analysis;
- Scatter diagrams;
- Run charts;
- Interrelationship diagraphs;
- Check sheets;
- Design of experiments (see Chapter 20).

Cause-and-Effect (Fishbone) Diagram (Figure 21.1)

- Allows the project team to identify, explore, and graphically display all of the possible causes related to a problem to discover its root cause.
- Helps the team to probe for, map, and prioritize a set of factors that are thought to affect a particular process, problem or outcome.
- Helps with eliciting and organizing information from people who work within a process and know what might be causing it to perform the way it does.
- Focuses the project team on causes, not symptoms.

Pareto Chart—Special Form of Histogram or Bar Chart (Figure 21.2)

- Helps focus investigations and solution finding by ranking problems, causes, or actions in terms of their amounts, frequencies of occurrence, or economic consequences.
- Is based on the proven Pareto principle: 20% of the sources cause 80% of any problem.

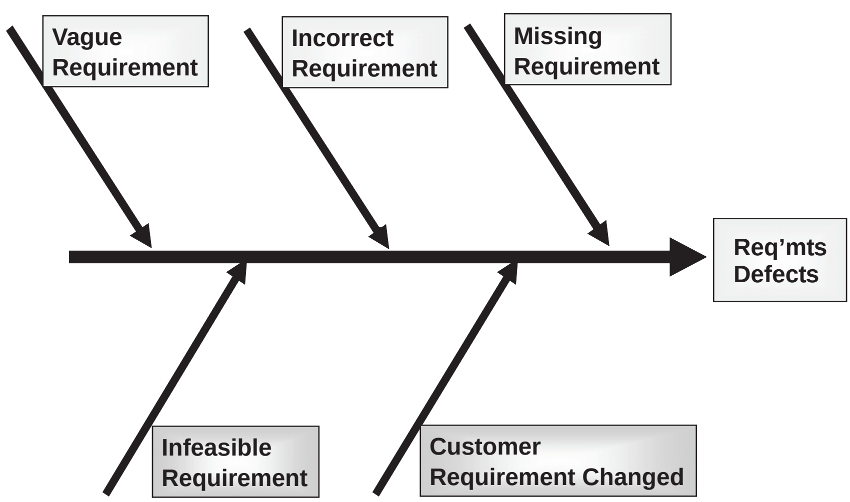


Figure 21.1 Cause-and-effect (fishbone) diagram.

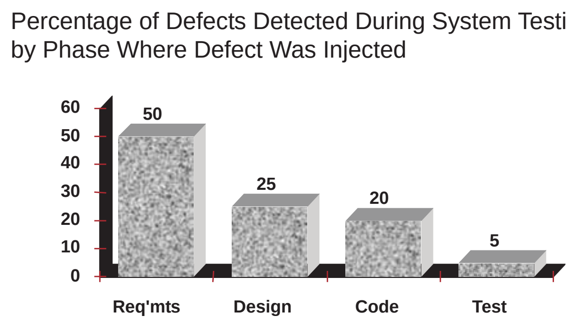


Figure 21.2 Pareto chart.

- Helps prevent "shifting the problem" where the solution removes some causes but worsens others.

Scatter Diagram (Figure 21.3)

- Displays empirically observed relationships between two process characteristics.
- A pattern in the plotted points may suggest that the two factors are associated, perhaps with a cause-and-effect relationship.

Run Chart (Figure 21.4)

- Specialized, time-sequenced form of scatter diagram that can be used to examine data quickly and informally for trends or other patterns that occur over time.

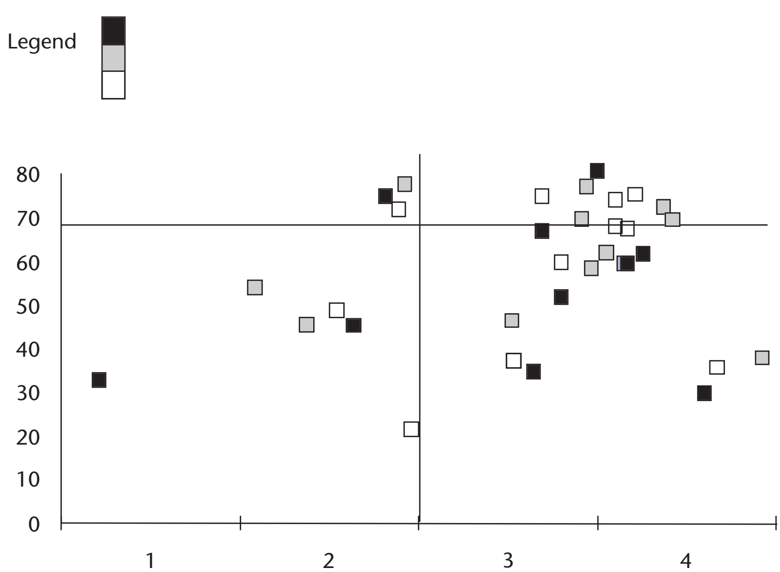


Figure 21.3 Scatter diagram.

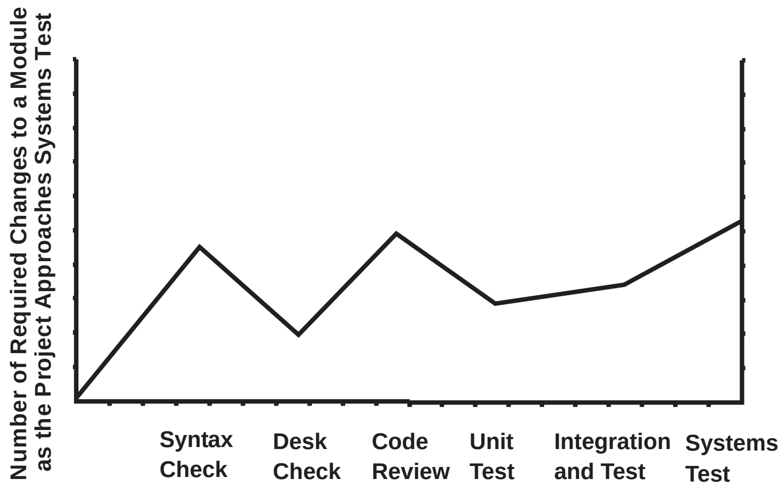


Figure 21.4 Run chart.

Interrelationship Diagraph (Figure 21.5)

- Allows a team to systematically identify, analyze, and classify the cause-and-effect relationships that exist among all critical issues so that key drivers or outcomes can become the heart of an effective solution.
- Encourages team members to think in multiple directions rather than linearly.
- Explores the cause-and-effect relationships among all the issues.
- Allows a team to identify root causes even when credible data does not exist.

Check Sheet (Figure 21.6)

- Allows a project team to systematically record and compile data from historical sources or observations as they happen.

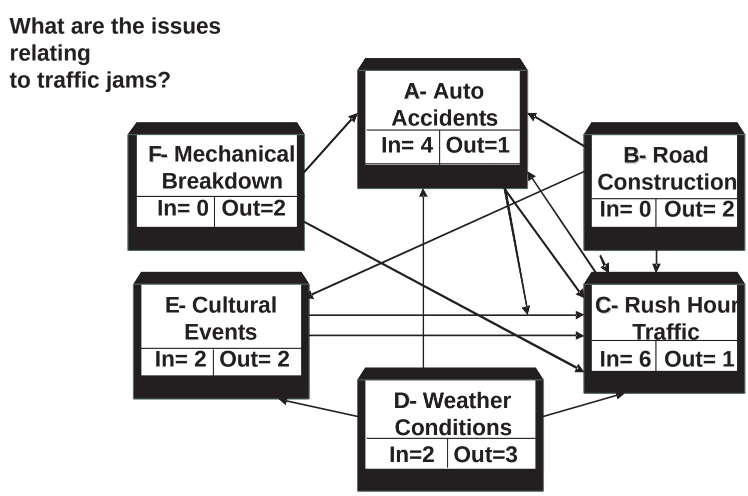


Figure 21.5 Interrelationships diagraph. (*After:* [2, p. 80].)

Errors Classification	Book Chapters					
	1	2	3	4	5	Total
Spelling	////	///	//	////	///	16
Punctuation	//	///	//	//	///	12
Missing Information	/	//	//	/		6
Redundancy	//	///	/	/	//	9
Technical Errors	//	/	//	/	//	8
Format Errors			//	/		3
Incomplete Concepts						
Total	11	12	11	10	10	54

Figure 21.6 Proof and checking errors check sheet. (*After:* [2, p. 35].)

- Patterns and trends can be clearly detected and shown.
- Builds, with each observation, a clearer picture of the facts as opposed to opinions of the team member.
- Ensures that recordings are made consistently.
- Makes patterns in the data become obvious quickly.
- Data must be collected over a sufficient period of time to be sure the data represents typical results during a typical cycle for the business.

Five Whys

"The Five Whys is a question asking method used to explore the cause/effect relationships underlying a particular problem. Ultimately, the goal of applying the Five Whys method is to determine a root cause of a defect or problem" [3]. The following example demonstrates the basic process:

The laptop computer must have a large, round red button for the operator to strike (the problem).

1. *Why?* It has to be big enough and sturdy enough to be struck with medium to strong force (first why).
2. *Why?* The operator has to have a way to stop the program that is running (second why).
3. *Why?* The operator has to have a manual override system to remain in control of the running system (third why).
4. *Why?* Otherwise unanticipated problems could occur (fourth why).
5. *Why?* It is imperative that the running system be able to be interrupted at any time by the operator in order to shut down a critical system to save human lives (fifth why, root cause).

Note that the questioning for this example could be taken further to a sixth, seventh, or even greater level. The *Five in Five Whys* is not a law, but rather, it has been shown that five iterations of asking why is generally sufficient to get to a root cause (Figure 21.7). The real key is to encourage the causal analysis team to avoid assumptions and logic traps. Instead, we want them to follow the chain of causality in increments through the layers of abstraction to a root cause that still has some connection to the original problem.

The use of the Five Whys also illustrates how this technique can support requirements elicitation. Reexamining the preceding scenario again, we can see that the requirement is that the computer operator have the ability to interrupt the running system at any time—whether or not a big red button mounted on the laptop is the solution or not should be up to the design team.

Addressing Defect Causes

Following the analysis of the selected defects and other problems to determine the root causes, it is often helpful to group or segment them into categories for easier causal analysis. We must not forget that the causal analysis, while certainly critical is not the most significant part of the Causal Analysis and Resolution process area. The most significant part of CAR is the action taken to implement and prevent similar defects or problems from occurring in the future. The preventive part of CAR is often overlooked.

Before these actions are implemented it is useful to analyze the actions proposed and determine priorities such as:

- Implications of not addressing the defects;
- Cost to implement process improvements;
- Expected impact on quality;
- What other similar defects may exist in other processes and work products.

If the preventive action of CAR in response to identified defects or problems is shown to be effective on the projects that requested the causal analysis, it may be useful to extend the improvement to the entire organization. Here is one of the

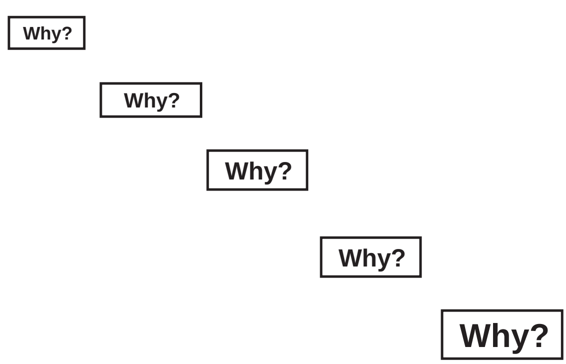


Figure 21.7 Five whys.

places where the concepts found in CAR and OID can be seen to overlap. Due to the quantitative base on which the organization and project can base CAR and OID, processes described in OID would be called on to determine the feasibility and cost effectiveness of deploying the improvements to all projects throughout the organization.

In some cases it can also be imagined where the preventive action specified by CAR would require the incremental or innovative improvement approach of OID. In any case, it is necessary to document and evaluate improvements to the organization's set of standard processes.

Was the Change Successful?

Before declaring victory it is necessary to evaluate the impact of the changes by gathering evidence that the process change has corrected the problem. Has the process performance improved? Has the capability of the project's defined process improved? Will the customer's expectations and requirements be satisfied? Process performance baselines and process performance models can be useful not only for identifying defects and problems to perform causal analysis techniques against, but can also be useful for predicting the impact and return on investment that results from the prevention activities. Hypothesis testing can also be used to confirm the improvement.

Once we have confirmed the improvement we also want to confirm the change of the capability of the process. Here is one example to illustrate this concept. An example of a change in the capability of the project's defined unit testing process is a change in the ability of the process to stay within its process specification boundaries. The change in capability may be statistically measured by calculating the number of major defects found during unit testing of the highest criticality before and after the improvement is made. This change in capability is represented by narrowing the control limits and/or showing a shift in the mean.

The bottom line: Has the process change corrected the problem and improved performance?

Organizational Innovation and Deployment: Enabling the Selection and Deployment of Improvements

The purpose of organizational innovation and deployment is to select and deploy incremental and innovative improvements that measurably improve the organization's processes and technology and in turn support the organization's quantitative quality and process performance objectives derived from its business objectives.

The focus of the OID process area is process improvement that is based on a quantitative knowledge of the organization's set of standard processes and technologies, and the expected quality and performance of those processes and technologies in predictable situations.

OID supports projects and process groups in selecting and deploying improvements that have the possibility to meet the quality and process performance objectives as derived from the organization's business objectives. These quality and process performance objectives include increased/decreased cycle time, reduced delivery time, and greater customer and end user satisfaction. Process performance is a measure of the actual process results achieved and is characterized by both process measures and product measures. Process performance models can be used to measure the costs and benefits that normally cannot be measured before the end of the life cycle is reached. OID can be used to support projects that need to be quantitatively managed by investigating incremental and innovative solutions that may help the project to meet its quality and process improvement objectives.

Process measures include:

- Effort;
- Cycle time;
- Defect removal efficiency.

Product measures include:

- Reliability;
- Defect density;
- Response time.

Collecting and Analyzing Improvement Proposals

Organizational innovation and deployment can be seen as an extension of the activities described in the Organizational Process Focus (OPF) process area. OPF provides candidate process improvement sources including measurement and analysis of the processes, lessons learned from process implementation, and peer review defect trends. In like manner organizational innovation and deployment directs project members and project group members to search for process and quality performance objective proposals that propose incremental and innovative improvements to specific processes and technologies.

Examples of sources for process and technology improvement proposals taken from the CMMI® description of OID include:

- Analysis of customer problems;
- Analysis of project performance compared to quality and productivity objectives;
- Analysis of data on defect causes;
- Examples of innovative improvements that were successful in projects outside of the organization;
- Process and technology improvement proposals.

It is quite possible for a project or organization to have the technical prowess to reach the product quality or process performance being demanded by the customer, industry, or competitors but at the same time, figuratively speaking, go out of business achieving those results. The OID process area clearly stresses that a cost–benefit analysis must be conducted before the decision is made to implement the improvements or innovations suggested. Certainly, the contribution of each candidate process improvement proposal against the organization's quality and process performance objectives must be estimated. Here is another opportunity to make use of process performance models. Not only can they provide insight into the effects of process changes, but they can also be used to predict performance of process change and support cost–benefit analyses. Process performance models can help determine whether conducting a pilot project is even worth it to the organization.

Deploying Improvements

Because innovations normally stand for "bleeding edge" technology or major changes in the normal way things are done, pilot projects should be identified to test these innovative or incremental improvements before they are broadly implemented. Each pilot project should be implemented in an environment that is characteristic of the environment in a broad-scale deployment.

The results of the pilot project must be evaluated to determine if the pilot should be terminated, whether the pilot needs to be replanned and reimplemented, or whether improvements should be deployed throughout more of the organization. Process performance models and hypothesis testing can be used to determine the extent of the improvement and its effectiveness. The final selection of process and technology improvement proposals for deployment across the organization should always be based on quantifiable criteria derived from the organization's quality and process performance objectives. It is important to always keep in mind that all incremental and innovative improvements that are selected for deployment must *measurably improve* the organization's processes and technologies.

A deployment plan for each selected process improvement should be developed. What does it take for widespread deployment throughout the organization? The deployment plan must take into consideration:

- How each process improvement needs to be adjusted for organization-wide deployment;
- What changes are needed to deploy each process and technology improvement including possible changes to:
  - Process descriptions, standards, and procedures;
  - Development environments;
  - Organizational culture and characteristics;
  - Existing commitments;
  - Knowledge and skills of workforce.

Remember, as the changed processes are deployed, it may be necessary to recalculate the process performance baselines and process performance models.

Measures and objectives must be established to determine the value of each improvement with respect to the organization's business objectives. Suggested measures found in the CMMI® include:

- Return on investment;
- Time to recover the cost of the process or technology improvement;
- Measured improvement in the project's or organization's process performance;
- Ability to respond quickly to changes in project requirements, market situations, and the business environment;
- Measured progress toward achieving the organization's quantitative quality and process performance objectives.

Summary

To summarize our discussion of CAR and OID, it is appropriate to also reiterate the purpose statements of OPP and QPM.

When higher degrees of quality and performance are demanded, the organization and projects must determine if they have the ability to improve the necessary processes to satisfy the increased demands. This is an important indicator of higher maturity organizations. They make decisions based on data and know their limitations due to the analysis of that data. Quantitative management *is tied* to the organization's strategic and quantitative goals for product quality, service quality, and process performance. Achieving the necessary quality and process performance objectives requires *stabilizing the processes or subprocesses that contribute most to the achievement of the objectives*.

Organizational process performance is used to establish and maintain a quantitative understanding of the performance of the organization's set of standard processes. It both defines and helps support the quantitative quality and process performance objectives for the organization and provide process performance data, baselines, and models to support the quantitative project management of the projects.

The organizational data for these measures is analyzed to establish a distribution and range of results. This provides an "expected performance" of the standard processes when used on any project throughout the organization.

Quantitative project management is responsible for determining what the quantitative quality and process performance objectives are for the project, developing a project's defined process that includes identified subprocesses that are statistically managed, and utilizing the support from CAR, OPP, and OID to achieve those quantitative objectives. QPM provides an understanding of both special and common causes of variation expected in process performance.

If the demands on the project cannot be met based on the organizational data from the process performance baselines, the project must determine what subprocesses it must statistically manage and try to stabilize them.

It is here that CAR comes into play, helping to identify the causes of defects and other problems, and taking action to prevent the occurrence of those defects or

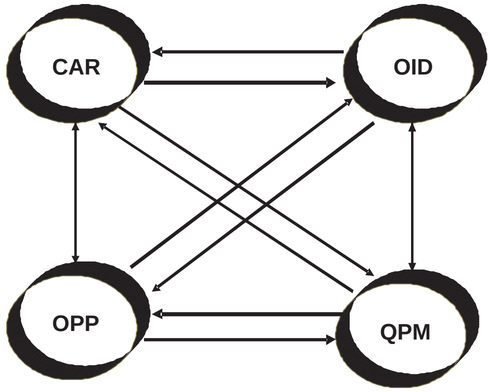


Figure 21.8 Interactions of OPP, QPM, CAR, and OID.

problems in the future. CAR may need to take advantage of OID if the action proposals are inadequate to solve the special or common cause of variation. OID enables the selection and deployment of incremental and innovative improvements that help the organization meet its quality and process performance objectives established in OPP, as well as help the project achieve its quantitative quality and process performance objectives. OID receives it improvement requests from OPP, QPM, CAR, and other project and organizational activities that require incremental or innovative support (Figure 21.8).

Both CAR and OID process areas are designed to assist an organization to move beyond stabilizing processes to improving them based on the quantitative understanding of those processes and not only meet the voice of the process but also the voice of the customer.

References

[1] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI Second Edition, Guidelines for Process Integration and Product Improvement,* Reading, MA: Addison-Wesley, 2007.

[2] Brasard, M., and D. Ritter, *The Memory Jogger II: A Pocket Guide of Tools for Continuous Improvement and Effective Planning*, Salem, NH: GOAL/QPC, 1994.

[3] Wikipedia, "5 Whys," February 2008, http://en.wikipedia.org/wiki/Five_whys.

CHAPTER 22

Reducing Variation

Understanding Variation

When the topic of variation is discussed, the context is normally control charts and statistical process control. But this chapter illustrates how the CMMI® can be used to reduce variation at each Maturity Level that eventually leads up to statistical process control.

One of the specialists in understanding variation who has influenced members of the SEI, as well as many other diverse fields such as transportation, manufacturing, utilities and aerospace, is Donald Wheeler. Wheeler is an internationally recognized speaker and trainer in the understanding and use of data in business and industry. Wheeler holds a doctorate in statistics from Southern Methodist University and was a student and associate of W. Edwards Deming for 19 years. Wheeler has written numerous technical books on statistical process control and related topics but the one book that continues to have a visible impact on those who strive to fully understand and use the CMMI® at higher maturity levels is *Understanding Variation—The Key to Managing Chaos* published in 1993 and republished in 2000 [1]. In Chapter 2 of his book, Wheeler discusses Walter Shewhart's approach to interpreting data [1]:

> We analyze numbers in order to know when a change has occurred in our process of system. . . . Some variation is routine, run-of-the-mill, and is to be expected even when the process has not changed. Other variation is exceptional, outside the bounds of routine, and therefore to be interpreted as a signal of process change. . . . The key to effectiveness of the process behavior chart or control chart is contained in the way in which the limits are computed from the data. . . . If over a reasonably long period of time, all of the points fall within the limits of a process behavior chart, and if the points are well behaved, then the process can be said to display nothing but routine variation. . . . However, when points fall outside the limits of a process behavior chart they are interpreted as signs of exceptional failure. Exceptional variation is attributed to assignable causes which, by definition, dominate the many common causes of routine variation. . . . Eliminating the effects of assignable causes of variation from your process can ensure that the process you are following will operate more consistently, predictably and more reliably.

Statistical process control and a more detailed explanation of common and assignable causes of variation were presented in Chapters 18 and 19. The ideas of variation are presented here to provide the backdrop for the main theme of this chapter. *The CMMI® model can be viewed not only as the basis for an organization's process improvement initiative but also as a continuous journey in reducing variation.*

This chapter presents an evolutionary path within the CMMI® model that illustrates how process improvement steps taken to move from an individual focus, to a project focus, to a measurement-oriented organizational focus, to a quantitative management focus can be thought of as successive steps in reducing variation in an organization's processes and business results. The process areas of Project Planning, Project Monitoring and Control, Measurement and Analysis, Organizational Process Definition, Integrated Project Management, Organizational Process Performance, and Quantitative Project Management will be used to support this chapter's concepts.

Variation Among Individuals

Figure 22.1 shows a "staged" view or representation of the CMMI® model. At the initial maturity level, the process is described as being unpredictable, poorly controlled, and reactive. We often described this state as "chaotic." One of the traits of CMMI® Maturity Level 1 is that the process belongs to the people. If others follow a process, it is normally due to the strong personality of someone on the project who has experience using processes in another environment. From a variation point of view, a ML 1 organization has great variation based on its individual employees following their own process paths. This is why ML 1 companies depends so heavily on the heroics of its people.

Level	Process Characteristics	Process Areas	
5 Optimizing	Focus is on quantitative continuous process improvement	Causal Analysis and Resolution Organizational Innovation and Deployment	
4 Quantitatively Managed	Process is measured and controlled	Quantitative Project Management Organizational Process Performance	
3 Defined	Process is characterized for the organization and is proactive	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organization Process Definition Organizational Training	Integrated Project Management Integrated Teaming Organizational Environment For Integration Integrated Supplier Management Risk Management Decision Analysis & Resolution
2 Managed	Process is characterized for projects and is often reactive	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management	Product and Process Quality Assurance Configuration Management Measurement & Analysis
1 Initial	Process is unpredictable, poorly controlled, and reactive		

Figure 22.1 CMMI® overview.

Project's Processes to Reduce Variation

At CMMI® Maturity Level 2, processes belong to the project and are enforced by the project manager. The processes, standards, guidelines, checklists, and templates are enforced for all of the project members through project planning and project monitoring and control to achieve more uniformity in development and product and service quality. Assuming that all projects followed some form of process, the amount of variation that was seen in organizations at Maturity Level 1 is reduced, even if all of the projects follow a different process.

In addition, the new process area of Measurement and Analysis helps the organization to develop and sustain a measurement capability that can be used to support management information needs. Measurement and analysis can also provide a foundation that can be built on as the organization evolves toward CMMI® Maturity Level 3 with its organizational set of standard processes. The process area of Measurement and Analysis guides an organization to define measures to be used along with the data collection process, the storage mechanisms, the analysis processes, the reporting processes, and the feedback processes. This is a critical step for an organization to become CMMI® Maturity Level 3 and to have the proper foundation in place for reaching CMMI® Maturity Level 4.

Organizational Processes to Reduce Variation

As described in Chapter 15, an organization that wishes to achieve CMMI® Maturity Level 3 needs to have its processes owned by the organization for economy of scale to be realized and process measurement to make practical sense. The organization's set of standard processes contains the definitions of the processes that guide all of the activities in an organization. A standard process enables consistent development and maintenance across the organization and is essential for long-term stability and improvement.

These organizational process definitions are tailored and incorporated into the project's defined processes throughout the organization and thus variation in project development and product and service quality is again reduced. This is described in the process areas of Organizational Process Definition and Integrated Project Management.

In addition to the standard processes, an organization at CMMI® Maturity Level 3 also establishes its organizational measurement repository. This measurement repository contains both product and process measures that are related to the organization's set of standard processes. It also contains the information needed to understand and interpret the measures and assess them for reasonableness and applicability. With this measurement repository, trends can be seen and predictability can be achieved. In addition, process performance baselines can now be developed to support quantitative project management.

Quantitative Project Management

Finally at CMMI® Maturity Level 4, the ownership of the processes reverts back to the projects. This is because individual projects must determine if the requirements and constraints placed on them demand the use of Quantitative Project Management techniques even though the tailored processes still come from the organization's set of standard processes. The data collected in the organization's measurement repository is used to develop a process performance database or, more accurately, a "set of process performance databases." This is described very well, in my opinion, in the process area of Organizational Process Performance (OPP). OPP also provides the quantitative measurement objectives for quality and process performance for the organization. These organizational quantitative quality and process performance measurement objectives may be inherited by the project or the project may need to develop quantitative objectives of its own.

Quantitative management is tied to the organization's strategic goals for product quality, service quality, and process performance. When higher degrees of quality and performance are demanded, the organization and projects must determine if they have the ability to improve the necessary processes to satisfy the increased demands. Achieving the necessary quality and process performance objectives requires stabilizing the processes that contribute most to the achievement of the objectives and reducing process variation to support the quantitative management objectives for the project. It is these concepts that are captured by the guidance provided by the process area of Quantitative Project Management.

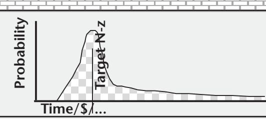
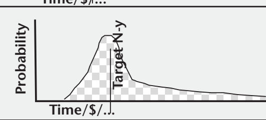
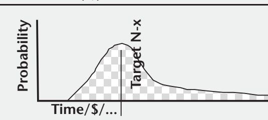
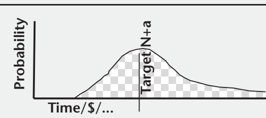
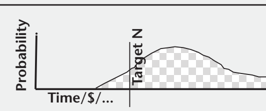
Level	Process Characteristics	Predicted Performance
5 Optimizing	Focus is on continuous quantitative improvement	
4 Quantitatively Managed	Process is measured and controlled	
3 Defined	Process is characterized for the organization and is proactive	
2 Managed	Process is characterized for projects and is often reactive	
1 Initial	Process is unpredictable, poorly controlled, and reactive	

Figure 22.2 Process capability prediction.

Summary

Figure 22.2 provides a process capability prediction view of the CMMI®. It illustrates the theme of reduction of variation that this chapter provided. At the initial level, target dates of cost, schedule, performance, and quality are often missed by wide variation.

At the managed level, the variability of the actual results around the target decreases. At the defined level, variability again decreases. Target hits increase and the target begins to move in toward the y-axis due to reduced rework. At the quantitatively managed level, variability continues to decrease. Target results improve, development times become shorter, and productivity and quality increase. Continuing to the optimizing level, defect prevention helps to reduce rework further and variation continues to be reduced.

There are many different views of the CMMI®. There are also many different ways in which the CMMI® can help an organization that are not always obvious on the surface. Reducing variation and improving process capability is a benefit of using the CMMI® that all organizations should understand, appreciate, and try to achieve.

Reference

[1] Wheeler, D. J., *Understanding Variation: The Key to Managing Chaos*, Knoxville, TN: SPC Press, 2000.

PART VII

Institutionalization

CHAPTER 23

Repeatable, Effective, and Long Lasting

Processes must be defined, implemented, and institutionalized if any organization expects them to be repeatable, effective, and long lasting. Questions we may ask ourselves include "Just what does it take for a process to be repeatable, effective, and long lasting? Why should organizations care?" Another question may be "What is institutionalization anyway?" and its related question is "Why does the CMMI® place such a great emphasis on this concept?"

Are Your Project Members Using Effective Processes?

Before answers to these and other related questions are provided, the reader ought to take out a blank piece of paper and list as many factors as possible on what an organization might put in place, or do, or direct to ensure that its developers and managers are using effective processes. It is probable that you listed training, because training is certainly an activity that helps us to be effective in utilizing the concepts that are offered. Naturally, all processes that an organization asks its people to follow should be documented, reviewed, and implemented. There should also be tailoring guidelines that each individual and project can use to obtain the maximum benefit from them. As lessons are learned, the documented processes ought to be updated to reflect the current understanding of business.

You have probably also listed resources because it takes money, time, people, and tools and technology to make processes repeatable and long lasting. Perhaps your list includes measures. And just possibly, your list includes policies that come from the senior management team to direct the everyday behavior and activities of the organizational workforce.

If your list included some of the items just mentioned, and perhaps even a few that were not listed, it means that you understand the technical, managerial, and cultural effort that it takes to truly develop processes for an organization's projects that can be repeated with measurable results, that are effective, and that will be long lasting.

One of the most important reasons that organizations should strive to ensure their processes are repeatable, effective, and long lasting is that process improvement is not free. Developing processes that are not intended to truly support measurable results is a waste of the organization's money and the time and energy of the people who engage in developing these critical processes.

Institutionalization

When we develop, review, and implement processes that are repeatable, effective, and long lasting we say that we are *institutionalizing* those processes across people, projects, and the organization. What then is Institutionalization? Institutionalization involves implementing practices that provide needed infrastructure support; ensures processes are defined, documented, and understood; and enables organizational learning to improve so those processes can take place. Effective institutionalization is evidenced by the fact that the processes are used and updated as a result of that usage.

Without institutionalization, these things happen:

- Processes are not executed or managed consistently.
- The processes will not survive staff changes.
- Process improvement may not relate to business goals.
- The organization will find itself continuously "reinventing the wheel."
- There will be no commitment to provide resources or infrastructure to support or improve the processes.
- There will be no historical basis for cost estimation.

The continuous representation of the CMMI® defines capability levels and generic model components to help an organization focus on its ability to pursue process improvements in multiple process areas and institutionalize their use along the way. Institutionalization, in this context, implies that the process is ingrained in the way the work is performed.

Although the continuous representation uses generic practices for Capability Levels 1 through 5, the staged representation uses only the generic practices from Capability Levels 2 and 3. This chapter concentrates on those generic practices pertaining to Capability Levels 2 and 3. First it is appropriate to define and illustrate the building blocks of Capability Level 0 and Capability Level 1.

Capability Level 0

Capability Level 0 does not mean that there is no measurable activity happening. It simply means that one or more of the specific goals are not being performed or are only being partially performed. This, then, implies that one or more specific practices are not satisfied and, hence, that the process area in question is not satisfied. The CMMI® labels this as incomplete implementation of processes. I use the titles of the specific goals (SGs) and specific practices (SPs) from the Configuration Management process area to illustrate Capability Levels 0 through 3 in this chapter.

Table 23.1 shows that if an organization and all of its projects were satisfying the specific practices within Specific Goal 1 and Specific Goal 2, but were not satisfying the specific practices within Specific Goal 3, Capability Level 0 would be assigned to Configuration Management for that organization. It should be clear from this example that this does not mean that nothing is being done regarding Configuration Management. If an organization had reached this level of satisfaction of

Table 23.1 Capability Level 0 for Configuration Management

Configuration Management		*Not Satisfied*
SG 1	Establish baselines	Satisfied
SP 1.1	Identify configuration items	Satisfied
SP 1.2	Establish a configuration management system	Satisfied
SP 1.3	Create or release baselines	Satisfied
SG 2	Track and control changes	Satisfied
SP 2.1	Track change requests	Satisfied
SP 2.2	Control configuration items	Satisfied
SG 3	Establish integrity	Not satisfied
SP 3.1	Establish configuration management records	Not satisfied
SP 3.2	Perform configuration audits	Not satisfied

the specific practices of Configuration Management, we could recommend that the organization focus next on Specific Goal 3 and its specific practices.

Capability Level 1

Before any organization can state that it has institutionalized practices for any given process area, it must be able to show that all of the specific goals for that specific process area can be achieved. Another way to express this is to state that when the project performs all of its processes for a process area, it can show that it can accomplish the work necessary to produce the required work products. It does not have to be pretty, and not all processes may be faithfully followed. The cost, schedule, and quality goals might not be met. It may require project members to be heroes. But it can be proven that every practice related to a specific goal of a process area is being implemented, and the work products necessary to lead to the delivery of the product or product component are being produced.

Note that this is a very significant statement. Maturity Level 1 could indicate that the organization is accomplishing almost nothing, or it could indicate that the organization is accomplishing almost all of the specific practices and corresponding specific goals.

Capability Level 1 indicates that all specific practices are being performed and the corresponding specific goals achieved.

Table 23.2 shows that if an organization and all of its projects were satisfying the specific practices within Specific Goal 1, Specific Goal 2, and Specific Goal 3, then it can be said that the organization satisfies the configuration management requirements found in the CMMI®.

Capability Level 2 Generic Practices

With the definition of Capability Level 0 and Capability Level 1 as the background, let us examine the generic practices that assist an organization to reach Capability

Table 23.2 Capability Level 1 for Configuration Management

Configuration Management		*Satisfied*
SG 1	Establish baselines	Satisfied
SP 1.1	Identify configuration items	Satisfied
SP 1.2	Establish a configuration management system	Satisfied
SP 1.3	Create or release baselines	Satisfied
SG 2	Track and control changes	Satisfied
SP 2.1	Track change requests	Satisfied
SP 2.2	Control configuration items	Satisfied
SG 3	Establish integrity	Satisfied
SP 3.1	Establish configuration management records	Satisfied
SP 3.2	Perform configuration audits	Satisfied

Level 2 in targeted process areas and eventually Maturity Level 2 and beyond if desired. The generic practices help an organization to institutionalize the processes supporting the specific practices across people, projects, and the organization.

Institutionalization for Capability Level 2 for any process area includes satisfying Generic Practices GP 2.1 through GP 2.10. These generic practices are discussed in the following subsections.

GP 2.1: Establish an Organizational Policy

Establish and maintain an organizational policy for planning and performing the process. The purpose of this practice is to define the organizational expectations for the process and make these expectations visible to those in the organization who are affected. Policies are expectation setting documents that are delivered from the senior management team. Policies then are "expectation setting documents." They should describe the behavior that the senior management team expects out of the workforce. For example, if a policy stated that every project must carry out formal life-cycle work product inspections on product components that have been identified as critical, an outside auditor could expect to interview a project leader and ask him or her, "What product components have been identified as critical for your project?" Show the place in the project plan or project quality plan where it is stated that formal life-cycle work product inspections will be conducted and the requirements for the reviewers who will participate in those inspections. How these inspections will be carried out, how much training is required, who will serve as moderators, how checking rates and logging rates will be determined, and how results will be analyzed are issues that should be covered in the organization's set of standard processes. Proving that the life-cycle work product inspections are planned and carried out on identified critical product components would satisfy this institutionalization requirement.

GP 2.2: Plan the Process

Establish and maintain the requirements, objectives, and plan for performing the process. The purpose of this practice is to determine what is needed to perform the process and achieve the established objectives of the project. It helps the project management and members prepare a plan for effectively performing the process on the project and getting agreement on this plan from all relevant stakeholders.

Planning the process means planning to use the documented process on a project so that its use will be effective. Even the planning process itself should be planned. We may ask ourselves, "What would it take to ensure that a process will be used effectively on a project?" Another way to phrase it would be "What do I need to do to successfully implement a process on my project?" Considerations for planning the process should include:

- The defined and documented process description in an accessible and usable form;
- The standards for work products and services;
- The product life-cycle models that are approved, documented, trained, and supported;
- The schedule for the process;
- The dependencies among the activities, work products, and services;
- The resources or work environment needed to perform the process including funding, people, and tools;
- Training needed;
- Work products to be placed under configuration management;
- Involvement of relevant stakeholders;
- Activities for monitoring and controlling the process;
- Objective evaluation activities for the process and work products;
- Management review activities for the process and work products;
- Measurement requirements to provide insight into the performance of the process, its work products, and its services.

These activities should be a normal part of project management planning, but GP 2.2 reminds us that using processes on our project does not just mean getting a template and filling it in to keep the process and quality police off of our backs.

GP 2.3: Provide Resources

Provide adequate resources for performing the planned process, developing the work products, and providing the services of the process. The purpose of this practice is to ensure that the resources needed are available when they are needed. The resources include:

- Adequate funding;
- Appropriate physical facilities;
- Skilled people or training, mentoring, and coaching to help the existing workforce gain the necessary knowledge and skills;
- Appropriate tools.

Remember that in Chapter 4 we stated that *adequate* or *appropriate* must be interpreted in light of the organization's business objectives. Of course, business constraints must always be considered. Organization's are encouraged to take a hard look at their business objectives and company capabilities in the resources category when attempting to institutionalize the suggested practices found in the CMMI®.

It is better to reduce the scope of the process improvement initiative for a period of time rather than attempt to accomplish all of the expected specific and generic practices without adequate resources. This can only result in frustration on the part of the process improvement champions and also for those who would like to utilize those improved processes, but can only get half-developed processes and occasional support.

GP 2.4: Assign Responsibility

Assign responsibility and authority for performing the process, developing the work products, and providing the services of the process. The purpose of this practice is to ensure that there is accountability over the life of the process for performing the planned process and achieving the specified results.

The real emphasis here must be on appropriate authority. Most management teams that I have talked to, assessed, or offered consulting support to have no trouble with assigning responsibility. When it comes down to actually giving individuals (e.g., project mangers) true authority to carry out the necessary processes identified for their project, however, difficulties begin to arise. If a schedule, budget, and resources have been assigned to the project manager, having the necessary responsibility and authority indicates the project manager can make decisions within the framework of the project without any overriding interference from higher level management. The assignment and authority must be ensured over the life of the process. However, one should not overlook the point here that although authority and responsibility should be assigned, there must still be a mechanism in place to effectively coordinate the actions and activities that have been developed. In other words, there must be a managing mechanism to ensure that progress is made in the right direction consistent with the efforts of others.

GP 2.5: Train People

Train the people performing or supporting the planned process as needed. Training must support the successful performance of the process by establishing a common understanding of the process and imparting the knowledge and skills needed to perform the process or support the performance of the process. The training should be developed for the appropriate levels of management and practitioners. The training should also be for other departments or groups or relevant stakeholders who do not require the details, but do need intermediate-depth training to ensure that they have an appropriate vocabulary to understand what to expect and what is expected of them. Overview training should be provided to those who interact with those performing the work to set proper behavioral expectations.

This generic practice focuses on training in the defined processes of a given process area needed at the project level and possibly at the organizational level. An example taken from the engineering process area of Requirements Development gives examples of training that might be offered:

- Requirements analysis techniques;
- Requirements elicitation tools and techniques;
- Requirements specification;
- Requirements modeling.

In my opinion, training should be thought of as "establishing the necessary knowledge and skills." Establishing the necessary knowledge and skills includes training, mentoring, coaching, and on-the-job experience. Training should also include updating previous experience.

GP 2.6: Manage Configurations

Place designated work products of the process under appropriate levels of control. The purpose of this practice is to establish and maintain the integrity of the work products throughout their useful lives. Remember from Chapter 8 on quality management that if a system exhibits integrity we can expect to have things happen such as these:

- Changes to any configuration item within the system are only made according to an established and maintained process and procedure.
- Life-cycle work products are kept consistent when requirements change requests are approved, and the requirements specification is then updated. All related life-cycle work products are reviewed to determine if accompanying changes to them are necessary as well.
- Periodic audits are made on the contents of the system to ensure that changes made to product components are both complete and correct.

The word *appropriate* in Generic Practice 2.6 must not be interpreted as merely requiring version control or developmental control. For example, change requests to requirements should require an organizational configuration control board to be involved because the contract or agreement with the customer may be changed.

I would like to point out here that the examples or generic practice elaborations indicate that many different artifacts as well as life-cycle work products should be candidates for being placed under configuration control. These artifacts include items such as:

- Work breakdown structure;
- Rationale for decision making;
- Technical data package;
- Criteria for product reuse;
- Acceptance documents for the received product components;

- Earned value reports;
- Estimating algorithms.

Performing GP 2.6 for all PAs will lead to a more controlled and effective project.

GP 2.7: Identify and Involve Relevant Stakeholders

Identify and involve the relevant stakeholders as planned. The purpose of this practice is to establish and maintain the expected involvement of stakeholders during the execution of the process. Identifying the set of stakeholders that needed to be involved during the project life cycle was discussed in Chapter 7 on project planning and monitoring and control. Generic Practice 2.7 reminds us for each process to think about which subset of stakeholders or relevant stakeholders should be involved and to what level to ensure that processes are adequately accomplished.

GP 2.8: Monitor and Control the Process

Monitor and control the process against the plan and take appropriate corrective action. The purpose of this practice is to perform the direct day-to-day monitoring and controlling of the process implementation, which includes these steps:

- Collect and analyze measures of actual performance against the plan.
- Review accomplishments and results of the implemented process against the planned process.
- Identify and evaluate the effects of significant deviations from the planned process.
- Identify problems in the planned and implemented process.
- Take corrective action when requirements and objectives are not being satisfied, when issues are identified, or when progress differs significantly from the plan.
- Track corrective action to closure.

This practice is also used to determine how effective the project's processes are and whether or not they are helping the project members to achieve the necessary product quality.

GP 2.9: Objectively Evaluate Adherence

Objectively evaluate adherence of the process and the work products and services of the process to the applicable requirements, objectives, and standards, and address noncompliance. The purpose of this practice is to provide credible assurance that:

- The process has been implemented as planned.
- The planned process satisfies the relevant policies, requirements, standards, and objectives.

- The implemented process satisfies the planned process.
- The results of following the process satisfy their requirements and standards.

Objective evaluation should provide all levels of management with confidence in the results that are being provided. It should address what processes are being followed on the projects, whether they are efficient and effective, and whether they are helping the developers to produce the required product quality.

Evaluation of adherence is typically done by people who are not directly responsible for managing or performing the activities of the process. It may be done by people within the organization but external to the process or project, or it may be done by people external to the organization.

GP 2.10: Review Status with Higher Level Management

Review the activities, status, and results of the process with higher level management and resolve issues. The purpose of this practice is to provide higher level management with the appropriate visibility into the process as described earlier in Generic Practice 2.9. These reviews may be part of the monthly senior management review meetings or may take place as a separate meeting. One Motorola business unit senior manager conducted monthly meetings on process improvement progress. He would ask the "process owners" to report on what progress had been made on the process improvement areas they were sponsoring. The "process owners" were middle managers in this Motorola business unit. This monthly process improvement progress check forced them to talk with their SEPG facilitator and even get involved to show their interest in the process improvement effort.

Also in attendance at these meetings were representatives from quality assurance and process, in addition to the project manager(s) to give the senior management team a panorama of what was really taking place process-wise on the projects. These meetings with the senior management team to discuss process and product quality improvement should be conducted such that they ensure that data-oriented decisions on the planning and performing of the process can be made.

Capability Level 3 Generic Practices

Institutionalization for Capability Level 3 of a process area includes GP 3.1 and GP 3.2, both of which are discussed next.

GP 3.1: Establish Defined Process

Establish and maintain the description of the defined process. The purpose of this practice is to establish a description of the project's process that is tailored from the organization's set of standard processes to address the needs of a specific instantiation on a project. Chapter 10 on integrated project management provided a detailed description of how this was to be accomplished for each integrated project. The descriptions of the project's defined processes provide the basis for planning, performing, and managing the activities, work products, and services associated with the process.

GP 3.2: Collect Improvement Information

Collect work products, measures, measurement results, and improvement information derived from planning and performing the process to support the future use and improvement of the organization's processes and process assets. This generic practice provides a reminder to all organizations and projects to constantly and forever keep improving the process.

A More In-Depth Look at GP 3.2

There is something more to be said about GP 3.2 to ensure that the reader realizes the depth of this generic practice:

- Collecting measures and results and work products to give back to the organizational process database does not mean that a project merely sticks its project information onto an organizational database and goes onto the next project.
- GP 3.2 implies that the lessons learned and measurements taken must be appropriate to be placed on the organizational process database and be deemed useful for future project examination and possible use.
- GP 3.2 also implies that there is a board or group that analyzes all data submitted at the end of a project. This analysis is done to determine if the data should be added to the existing organizational process database or replace existing data.

Relationships Between Generic Practices and Process Areas

There are links between some of the generic practices and the process areas that provide assistance with the implementation and institutionalization of the CMMI® practices. This was done by the CMMI® authors on purpose to assist in the institutionalization of the practices for each process area. One might look at this relationship in this way: A process area can be seen as enabling a generic practice to be accomplished. For example, Configuration Management enables or supports the implementation of GP 2.6 (Manage Configurations). If the Configuration Management process area can be shown to be fully implemented across the organization, the organization can also say that GP 2.6 has been fully implemented.

Likewise, a generic practice may be thought of as a subset of a complete process area. If an organization implements GP 2.6 (Manage Configurations) for a set of process areas, one can say that a subset of Configuration Management is being satisfied. This synergy is important for an organization to understand as it progresses on its process improvement journey, and it is important for lead appraisers and appraisal teams alike to understand when they are conducting SCAMPI Class A appraisals.

Other relationships between generic practices and process areas include:

- GP 2.9—Process and Product Quality Assurance;
- GP 2.7—SP 2.6 of Project Planning;
- GP 2.8—Project Monitoring and Control and Measurement and Analysis;
- GP 3.1—Integrated Project Management;
- GP 2.2—Project Planning.

Table 6.2 in the Reference Model and Table 7.2 in [1] provide a detailed mapping of many of the generic practices to their process area counterparts.

Appraisal Considerations

If an organization has previously received a CMMI® Maturity Level 2 rating and now wishes to be appraised at CMMI® Maturity Level 3, it must not only institutionalize the process areas defined at CMMI® Maturity Level 3, but must also ensure that the process areas previously institutionalized at CMMI® Maturity Level 2 are also institutionalized at CMMI® Maturity Level 3.

Summary

Processes must be defined, implemented, and institutionalized if any organization expects them to be repeatable, effective, and long lasting. Institutionalization involves implementing practices that provide needed infrastructure support; ensures processes are defined, documented, and understood; and enables organizational learning to improve so those processes can take place. These practices are defined as generic practices. They are expected model components and are applicable for all process areas. Generic Practices 2.1 through 2.10 define the practices that help organizations institutionalize the process areas as Capability Level 2. Generic Practices 3.1 and 3.2 define the practices for institutionalizing the process areas as Capability Level 3. Understanding the relationships between the process areas and the generic practices provides assistance with the implementation and institutionalization of the CMMI® practices.

Reference

[1] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI® Second Edition, Guidelines for Process Integration and Product Improvement*, Reading, MA: Addison-Wesley, 2007.

PART VIII

Advanced Concepts

CHAPTER 24

The Constagedeous Approach to Process Improvement

The management, quality, engineering, and process management principles behind CMMI® and its use in process improvement are the same regardless of the model representation—staged or continuous—that was briefly defined in Chapter 3. The motivation behind launching and sustaining process improvement programs should always be backed by sound business goals regardless of the CMMI® representation that is chosen.

Choosing Between the Staged and Continuous CMMI® Representations

Given the two representations in the CMMI®, an organization may feel it has to choose a process improvement path from either the process area capability approach or the organizational maturity approach. However, after using the CMMI® to conduct workshops, conducting CMMI® assessments, and helping organizations to get their CMMI®-based process improvement program going, it is my belief that both the staged representation and the continuous representation not only can, but must be used together to provide a realistic approach that results in effective process improvement to support an organization's business objectives.

In this chapter we look at the similarities and differences between the two representations and suggest ways that the combined continuous *and* staged, or *constagedeous* approach can provide guidance that will result in an effective process improvement initiative.

To further support the view of the CMMI® as a constagedeous model, the SEI released CMMI®-DEV v1.2 in August 2006 and presented the process areas in a constagedeous manner. CMMI® v1.1 was available in either the staged representation format or the continuous representation format. CMMI®-DEV v1.2 is presented only in the constagedeous format.

CMMI® Structure: Staged Versus Continuous

The difference between the staged and continuous representations is in the way the CMMI®'s process areas are presented and implemented. As you are aware, a process area is a collection of related specific practices that are performed collectively to achieve objectives pertaining to specific goals combined with generic practices that are performed collectively to achieve institutionalization objectives specified by generic goals. In the staged representation, process areas are grouped into Maturity Levels. Organizations choosing the staged framework implement those process areas at the specified Maturity Level. These sets of process areas are predetermined. As each Maturity Level is achieved, the set of related processes is stabilized, and the organization's ability to predict the future performance of its processes increases. Figure 24.1 presents the classic view of the staged representation from the CMMI®-DEV v.1.2.

In the continuous representation, process areas are grouped into four process categories (process management, project management, engineering, and support; see Figure 24.2). These and other alternative categories were discussed in Chapter 3. These process categories are implemented as determined by the organization. Process areas achieve capability levels, which reflect the "manner" in which the content of the process areas is performed. The "manner" simply indicates *how* the content of the process area is performed.

For both staged and continuous representations, the content of a process area is reflected in the process area's specific practices and specific goals. In other words, specific practices are indicators of *what is needed* to perform the process area, whereas specific goals are indicators of *the expected behavior*. Generic goals and generic practices, on the other hand, are indicators of process institutionalization as described in Chapter 23. They indicate whether the practices behind the process areas have been mastered by the organization and have become the normal way people perform these activities. Institutionalization implies that there is an infrastructure in place for ensuring the adoption and permanence of best practices. In both representations, institutionalization occurs via the generic practices.

Level	Process Characteristics	Process Areas	
5 Optimizing	Focus is on quantitative continuous process improvement	Causal Analysis and Resolution Organizational Innovation and Deployment	
4 Quantitatively Managed	Process is measured and controlled	Quantitative Project Management Organizational Process Performance	
3 Defined	Process is characterized for the organization and is proactive	Requirements Development Technical Solution Product Integration Verification Validation Decision Analysis & Resolution	Organizational Process Focus Organization Process Definition Organizational Training Integrated Project Management Risk Management
2 Managed	Process is characterized for projects and is often reactive	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management	Product and Process Quality Assurance Configuration Management Measurement & Analysis
1 Initial	Process is unpredictable, poorly controlled, and reactive		

Figure 24.1 CMMI®-DEV v1.2 staged representation.

Module	Process Areas
Process Management Concepts	Organizational Process Focus Organizational Process Definition Organizational Training Organizational Process Performance Organizational Innovation and Deployment
Project Management Concepts	Project Planning Project Monitoring and Control Supplier Agreement Management Integrated Project Management Risk Management Quantitative Project Management
Engineering Concepts	Requirements Development Requirements Management Technical Solution Product Integration Product Verification Validation
Support Concepts	Configuration Management Process and Product Quality Assurance Measurement and Analysis Decision Analysis and Resolution Causal Analysis and Resolution

Figure 24.2 CMMI®-DEV v1.2 continuous representation.

Process Improvement Is the Driving Force

Organizations that choose CMMI®-DEV v1.2 as a framework for process improvement do so because they want mature, stable, and predictable processes. Representation is not the driving force. Process improvement is the driving force. Process improvement does not happen just for its own sake. Every organization determines what its business goals are and what its measurement objectives are, and based on this they must discern the path to process improvement. When you look at your own business and you know what the model content is, you start thinking what you need to do to address your business issues. You don't think continuous or staged. You do process improvement to support your business.

The question is then: "Why has there been such a controversy over the use of the staged or continuous representation of the CMMI®?" Certainly the legacy of the use of the CMM® for Software is a reason to lean toward the staged representation of the CMMI®. Certainly if you were an engineering-oriented company, you might be tempted to choose the continuous representation of the CMMI® because it has a strong focus on the engineering process areas.

But is it true that, if an organization chooses the staged representation, the processes can only be improved if they are associated with the process areas defined at Maturity Level 2? In other words, is an organization allowed to try to improve its requirements development processes if the organization has not achieved CMMI® Maturity Level 2 yet? Requirements development is a CMMI® Maturity Level 3 process area. But how can an organization successfully institutionalize the practices found in Requirements Management at CMMI® Maturity Level 2 if it does not have the proper mechanism to gather and analyze the requirements in the first place? If an organization chooses to focus on the engineering process areas brought over from the systems engineering CMM®, and it chooses the continuous representation, does it have to worry about project management at all, or can that come later after the engineering processes are in place and being implemented?

If you have been involved with process improvement to support your business objectives with measurable results, you perhaps used an incremental approach to collect or gather the requirements in an ad hoc manner at first and then applied the Requirements Management principles to those requirements so that you could start improving your requirements gathering techniques as well as your Requirements Management processes. Now you undoubtedly realize that this sounds a lot like the continuous representation of the CMMI®. You may therefore reach the following conclusion that "My company should use the continuous representation, right?" Well, the answer to the last question is that, in fact, you will want to use both approaches concurrently, and we discuss how that can be accomplished in this chapter.

Myths and Misconceptions

We would like to offer a set of myths and misconceptions, negative influences on process improvement, and positive influences on process improvement from the point of view of the staged representation and the continuous representation to make a stronger case for an organization to use "both" as tools to support process improvement.

Staged Representation

The staged representation of the CMMI® focuses on organizational maturity:

- Represented by the degree of process improvement across a predefined set of process areas.
- Chosen to meet the process improvement needs of an organization.
- Processes are ordered and then grouped based on predefined organizational maturity relationships that address the business needs of many organizations.
- Provides an indicator of the maturity of an organization's processes in order to answer the question "What will be the most likely outcomes of the next project we undertake?"

Myths and Misconceptions (Staged)

- Requirements Management must be implemented fully before Requirements Development can take place.

- Quality Assurance and Configuration Management only apply to projects at CMMI® Maturity Level 2.
- An organization's measurement program must be fully implemented by the time it reaches CMMI® Maturity Level 2.
- Peer reviews should not be implemented until the organization is ready for CMMI® Maturity Level 3 activities.
- Risk Management should not be implemented until the organization is working on CMMI® Maturity Level 3.
- Organizations will not have trouble with suppliers after reaching CMMI® Maturity Level 2.
- Engineering activities are not necessary to achieve CMMI® Maturity Level 2.
- Organizational training is not necessary until an organization is ready to start on CMMI® Maturity Level 3 processes.

Negative Influences on Process Improvement (Staged)

- Gives organizations the idea that they can only implement process improvement initiatives that focus on process areas that are included in a particular maturity level.
- Does not provide guidance on how to incrementally implement process areas such as the Technical Solution PA.
- Does not provide any guidance for what it would look and feel like to implement a ML 3 process area from a lower maturity level perspective.
- Focuses the organization more on the achievement of a maturity level than on measurable improvement that supports the organization's business objectives.
- Can influence the organization to focus too much on management practices and neglect technical practices.

Positive Influences on Process Improvement (Staged)

- Helps organizations prioritize their process improvement efforts, especially when:
  - The organization's process improvement initiative is just getting started;
  - The organization is at a low maturity level;
  - The organization has little experience in process deployment.
- Focuses the organization on putting in place project management functions that are needed to support all of the organization's engineering and management activities throughout its process improvement journey.

Continuous Representation

The continuous representation of the CMMI® focuses on organizational process area capabilities:

- To the extent to which a process is explicitly documented, managed, measured, controlled, and continually improved;
- That represent improvements in the implementation and effectiveness of an individual process area;
- That support the continuous improvement of individual process areas that are critical to the organization's business needs;
- That provide an indicator of improvement within a single process area in order to answer the question "What is a good strategy for implementing improvement of this process area?"

Myths and Misconceptions (Continuous)

- An organization can focus its process improvement initiative on any group of process areas it wants without worrying about possible dependencies.
- Capability Level 1 for a given process area is easily obtained (e.g., Quantitative Project Management).
- An organization can practically evolve any process area to Capability Level 5.
- Focusing on the continuous representation and the capability levels will remove the focus on the number that is associated with the staged representation.
- There is a clear evolutionary path that all process areas can simply follow to help an organization evolve to capability level 5 for all or most of its process areas.
- Using the continuous representation will not require prioritization of the implementation of its process areas.
- The engineering process areas are not as useful for software engineering applications because they are based primarily on systems engineering needs.

Negative Influences on Process Improvement (Continuous)

- Seems to indicate that an organization is given extreme flexibility of choosing the process areas for implementation in any order desired without regard to dependencies.
- Allows an emphasis on implementing the engineering activities and diminishing the importance of the management activities.

Positive Influences on Process Improvement (Continuous)

- Through the use of the engineering process areas and the generic practices, the continuous representation can provide an individual capability improvement path for each process area.
- If the business need demands it, the path to evolve a particular process area or category of process areas to higher levels of capability can be easily identified and measured.

- Assists an organization in constructing a target profile of process areas that collectively will help the organization to solve known business objectives.

The Constagedeous Approach to Process Improvement

If your company is truly focused on process improvement to support its business objectives and to deliver high-quality products and services, then it should use the "constagedeous" approach to process improvement. In reality, many organizations do "constagedeous" improvements—whether they realize it or not—by choosing to implement the continuous representation in a staged way. They address a couple of process areas at a time before they move to another set of improvements. They may look at their progress by assessing the process area(s) of interest (continuous assessment), or they may put a number of improvements in place and decide to examine the process areas belonging to a certain maturity level (staged assessment).

Less experienced organizations may want to avoid being overwhelmed by the demand of the staged representation to assess the progress of all process areas that belong to a maturity level. They may thus choose an evolutionary approach to realizing and validating improvements, a couple of process areas at a time, using the "constagedeous" approach. It's all a matter of how one looks at these possibilities. Process improvement is an evolutionary activity—organizations take small steps at a time in an incremental way. No one can address all of the issues all at one time. You do interim progress checkups regardless of the representation. You could employ the continuous approach to assess the capability of your process areas or you could put improvements in place and wait and assess the maturity level of a number of process areas as prescribed by that level. If you want to focus on project management process areas, your organization will need to gather requirements; design and build components and peer review and unit test them; perform integration; conduct systems testing; and eventually produce a product that is delivered. In other words, you still need to perform basic engineering activities. Claiming CMMI® Maturity Level 2 without doing engineering makes absolutely no sense.

If you want to focus your organization on the engineering processes areas, you will still need enough Project Planning, Project Monitoring and Control, and Risk Management to manage the project. You still need to perform sufficient Configuration Management to control the life-cycle work products that your project produces, and you must ensure that the more critical processes are followed to guarantee the minimal product quality defined in your requirements. You may even have to manage suppliers who will build subsystems that you will integrate to build the final product. In other words, you still need to perform the basic project management functions and eventually produce a product that is delivered. Claiming CMMI® Maturity Level 2 without doing project management makes absolutely no sense.

Equivalent Staging

The CMMI® has defined equivalent staging as shown in Figure 24.3. Equivalent staging provides an organization with a mechanism to use the continuous representation to guide the process improvement initiative and still be aware of the progress toward any maturity level defined by the staged representation.

As Chapter 23 pointed out, if an organization wanted to institutionalize Requirements Management at Capability Level 2, it would have to prove it could satisfy all of the specific practices or approved alternative practices and, in turn, satisfy the corresponding specific goals. Then it would have to show it had institutionalized the practices for Requirements Management by satisfying Generic Practices 2.1 through 2.10, which in turn would satisfy Generic Goal 2. Figure 24.3 points out a relationship that is informative and even exciting. If an organization sets up a process improvement plan that would result in the achievement of Capability Level 2 for the process areas of Requirements Management, Project Planning, Project Monitoring and Control, Process and Product Quality Assurance, Configuration Management, Supplier Agreement Management, and Measurement and Analysis, it would automatically achieve Maturity Level 2. Achieving Capability Level 2 for those predefined process areas is the equivalent of achieving Maturity Level 2.

In addition, there are no mental restrictions to working on multiple target capability levels for other process areas at the same time. An organization could work for a target capability level (CL) of CL 2 for Requirements Development and Technical Solution, a target capability level of CL 1 for Verification and Validation, and still be working on achieving CL 2 for the seven process areas that make up the predefined set for Maturity Level 2.

In my opinion, this flexibility offered by taking a constagedeous approach to process improvement makes the CMMI® the most powerful model in the world.

Case Studies

Using the constagedeous approach and focusing on the continuous representation for process improvement increments that are not related to maturity levels offers multiple alternative process improvement paths that are not obviously available with any other model or standard.

Understanding the relationships between process areas and generic practices together with the link between the capability levels and the generic practices allows the alternative process improvement paths to be defined. Let's look at a couple of case studies.

Case Study 1

A European organization produced software for back-office banking applications. It had experience with ISO 9001:2000 but not the CMMI®. The organization had a parent who was also its major customer. The parent was not happy with the way requirements were being elicited or managed. The software organization itself would be classified as a hard CMMI® ML 1. Some consultants might be tempted to tell this organization that it needed to achieve Maturity Level 2 and then start working on refining its Requirements Development process. However, the software organization did not have this luxury. Using the continuous representation and the definitions of the generic practices, I suggested the following path:

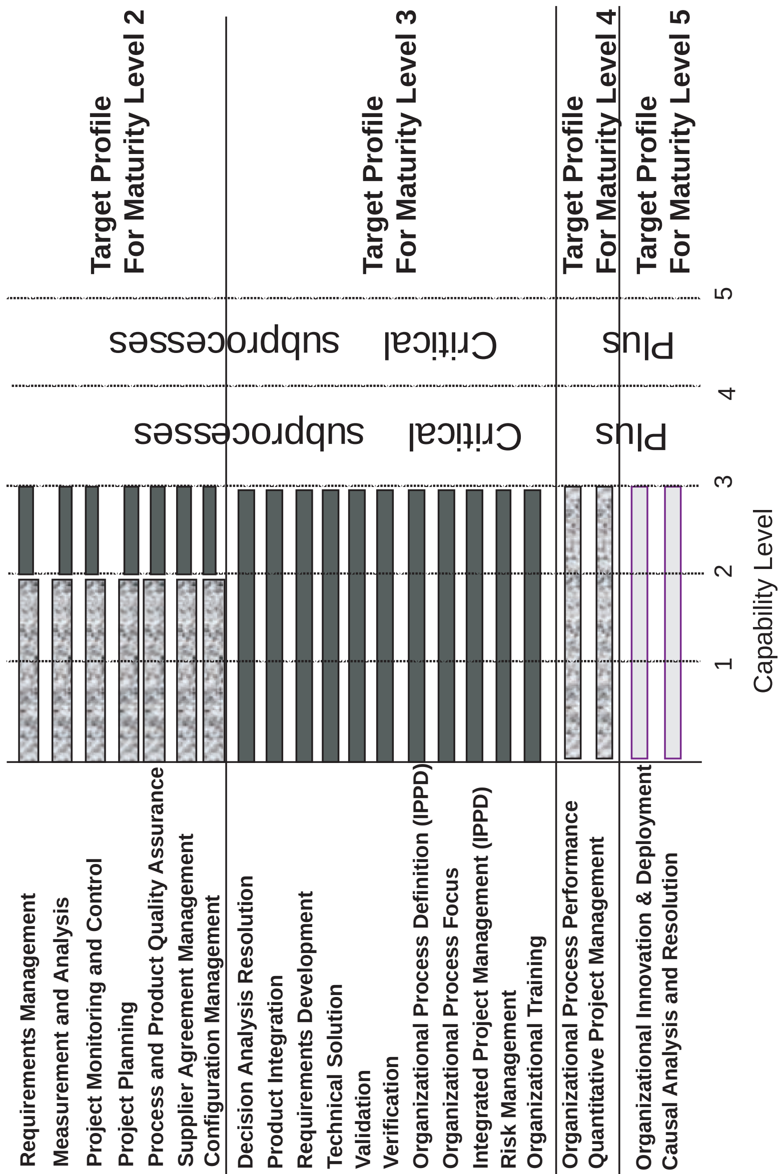


Figure 24.3 Equivalent staging. (*After:* [1].)

- Get the required training, mentoring, and coaching on Requirements Development and the part of the Technical Solution PA for Maturity Level 3 that overlaps with requirements elicitation and refinement.
- Focus on Capability Level 2 institutionalization. An organizational level or even a Capability Level 3 was not necessary to satisfy the business objectives at this time.
- Implement Requirements Management from Maturity Level 2.
- Implement enough Configuration Management to support the Requirements Development, Technical Solution, and Requirements Management PAs. This could be accomplished by satisfying GP 2.6 (Manage Configurations) for those process areas.
- Implement enough quality assurance to support the requirements engineering activities. This could be accomplished through satisfying GP 2.9 (Objectively Evaluate Adherence).
- Implement enough project planning and project monitoring and control to guide the requirements engineering efforts. This could be accomplished through the implementation of GP 2.2 (Plan the Process) and GP 2.8 (Monitor and Control the Process).

Although the software organization was against the full implementation of Configuration Management and Quality Assurance, the use of the related generic practices to show strong requirements engineering practices was acceptable.

And although the software organization did not understand why it had to worry about planning and tracking in general, it did accept the need to support the improvement they needed for requirements engineering.

This initial process improvement effort was successful. In time, the organization asked for support in Risk Management and Supplier Agreement Management. After some visible successes in those process areas, the organization decided to conduct an appraisal to determine if the process improvement progress they had made brought them close to being awarded CMMI® Maturity Level 2. They were also interested in how much progress they had made in the individual ML 3 process areas they had had to implement to succeed in the requirements engineering practices previously. The incremental process improvement approach using the continuous representation had not only helped the organization to succeed in its business and keep its major client, but it had helped the software organization to realize the value of the entire CMMI® and gave them encouragement to conduct a staged representation appraisal and continue with their process improvement initiative.

Case Study 2

An organization decided to use the CMMI® for its process improvement initiative and immediately wanted an appraisal that would rate them against Maturity Level 2 requirements. Because the organization had little previous experience with the CMMI®, I suggested we conduct a SCAMPI B appraisal instead of immediately conducting the more formal and demanding SCAMPI A appraisal. I also suggested that if we found any weak process areas in the organization that the appraisal would become more tutorial in those areas rather than only come across as harsh if process areas were not being implemented in an institutionalized way. An agreement with management was made and the SCAMPI B appraisal was conducted.

The SCAMPI B appraisal results showed that many processes were defined, but had not been taught or piloted on even one project. Other processes were defined and understood, but the implementation had just started and institutionalization was a long way off.

The organization developed an action plan and told me it had started working against it. A year later, the organization requested a second SCAMPI B. Shortly into this appraisal, it became apparent that little heed had been paid to the recommendations that were made during the first SCAMPI B appraisal. Management was perplexed, but obviously had not been visibly involved with the process improvement activities during the past year. Because most of the recommendations were going to be exactly the same, I tried to think of what message could be delivered that would support the organization's business objectives but not make the Maturity Level 2 score the only focus of success.

Using the continuous representation and the links between the process areas and the generic practices, I persuaded the organization to concentrate on the process areas it needed most to succeed in and use the generic practices to get it started in the admittedly weak process areas. The result was the following:

- Requirements Management, Project Planning, and Project Monitoring and Control would be targeted for Capability Level 2.
- Process and Product Quality Assurance, Supplier Agreement Management, and Measurement and Analysis would be targeted for Capability Level 1.
- Configuration Management would be targeted for Capability Level 0, but goal satisfaction for SG 1 and SG 2 would be achieved.

Although is it possible for anyone to come up with such a scenario using the tried and true staged representation, the task was made so much easier with the tools offered by the continuous representation. Once these initial target capability levels were achieved, the focus would be on raising the capability levels of the other process areas that were targeted for CL 1 or CL 0. Eventually the equivalent staging of CL 2 for all process areas defined in ML 2 would be targeted and the organization could reach its CMMI® Maturity Level 2 goal while simultaneously satisfying its business objectives.

No one doubts that achievement of a maturity level is an important benchmark and an important marketing tool for any organization. But any process improvement initiative must have as its first focus the accomplishment of its business objectives. Using the combination of tools afforded by the staged and continuous representations of the CMMI® has proven to be the most effective approach to accomplishing both.

Summary

The message is clear. When you choose the continuous approach, the capability of your various process areas has implications on organizational maturity, and when you choose the staged approach the organizational maturity has implications on the maturity of your process areas. Process improvement is the driving force—and process improvement does not happen for its own sake. Every organization must focus on its own business objectives and vision. It must determine what its problems or process weaknesses are, and based on these, must determine the path to process improvement.

When you look at your own business objectives and you know what the model content is, you can start thinking about what you need to do to address your business issues. You do not think continuous or staged. You simply do process improvement to satisfy your business issues and you do this using a "constagedeous" approach.

Reference

[1] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI® Second Edition, Guidelines for Process Integration and Product Improvement*, Reading, MA: Addison-Wesley, 2007.

CHAPTER 25

Applying CMMI® to Manufacturing*

Sarit Assaraf and Itzhak Lavi, Israel Aerospace Industry in Israel, Guest Authors

Background and Motivation

Israel Aerospace Industries (IAI) is involved in development, manufacturing, maintenance, and service of aerospace products. Some of IAI's divisions and groups focus on development projects; others focus mainly on manufacturing processes. In some cases, such as the IAI/LAHAV division, the division's business is based on both development and manufacturing projects.

When IAI decided to implement the CMMI®, this decision referred mainly to development projects. But when we started to consider the implementation of CMMI® in the IAI/LAHAV division, we decided to examine the possibility of implementing CMMI® practices in manufacturing processes as well.

Initial Analysis

The CMMI® was designed and constructed mainly for development processes. In fact, version 1.2 of the model is explicitly called "CMMI® for Development."

As a result, the examples provided by the model are directed mainly to development processes, and in version 1.1 they were confined only to software development and system engineering processes. Yet, we found that it is possible and helpful to apply most of the best practices contained in the model to manufacturing processes as well.

In an organization where a significant portion of its business is based on manufacturing in addition to development, the CMMI® can be used to help create and maintain uniform processes as much as possible by aligning all processes, both in development and in manufacturing, to comply with the same process model. In fact, we found that in some aspects, applying CMMI® to manufacturing was sometimes easier than in development:

- The level of repeatability in manufacturing processes is higher, due to the number of identical products that are produced.
- The culture of measurement and analysis is much better institutionalized in manufacturing than in development.

- The manufacturing processes rely heavily on standard components and interfaces, most of them are COTS.
- The level of interchangeability is high, indicating repeatable processes—the outcome of institutionalized engineering procedures.

In IAI/LAHAV's case, the organizational business is composed of a mixture of development and manufacturing. Both contribute significantly to the organization's business, and we decided to include both development and manufacturing processes in the CMMI® implementation effort.

First we describe two types of manufacturing projects: build to specifications or "build to spec" and build to print projects.

BTS and BTP Projects' Context

- *Build to spec (BTS):* The product specification is provided by the customer and the project has to design the product, design and build the production line, and manufacture products according to customer order
- *Build to print (BTP):* A complete product design is already available. The design is supplied either by the customer or comes from a previous BTS project, and the project sometimes has to design, build, or change the production line infrastructure and manufacture units according to a customer order.

Figure 25.1 describes the basic process used in BTP and BTS projects. This process helped us map the CMMI® engineering process areas to the activities embedded in the manufacturing process.

In BTS projects, customer requirements are defined mainly in terms of product specifications. Based on product specifications, the product is designed, integrated, and tested. The production line requirements are developed, and the production

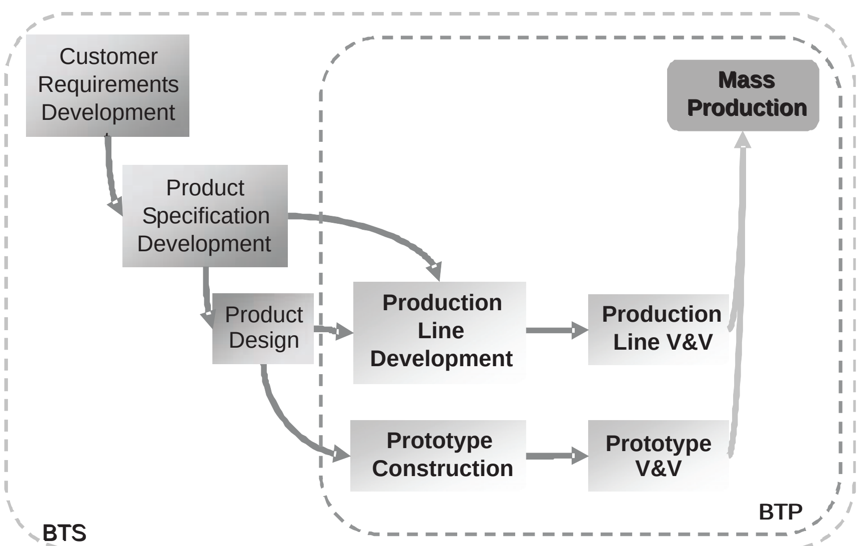


Figure 25.1 Basic BTS and BTP processes.

infrastructure is designed, integrated, and tested. The developed product includes not only the product to be manufactured, but also the production line infrastructure, involving both product design and production engineering processes. The production line infrastructure may be used in the manufacture of several products in several production projects.

BTP usually involves mass production according to a predefined specification or drawing, therefore, no development of the product is needed. Customer requirements are defined mainly in terms of the product technical package (drawings, part lists, and so forth) and are managed by manufacturing engineering. Based on the product's technical package, the manufacturing line requirements are developed (when necessary), and the manufacturing infrastructure is designed, integrated, and tested. This infrastructure may be used in the manufacture of several products in several manufacturing projects by applying interchangeability and flexibility of the production rate, allowing reductions in maintainability costs:

- Different versions of the product may be manufactured using the same infrastructure.
- Different products may be manufactured using different versions of the manufacturing infrastructure.

Applying CMMI® in BTS and BTP Projects

This section describes the implementation of CMMI® in BTS and BTP manufacturing projects. The set of processes in the organization consists of processes unique to manufacturing and other processes that are unique to development. These processes are documented in different procedures, some of which are specific for development and others for manufacturing, allowing for the coexistence of two different business lines implementing the same framework. Selecting the appropriate procedures is part of the tailoring process performed at the beginning of the project.

When applying CMMI® to BTS and BTP projects, the entire set of relevant processes should be considered, including development of the product (in BTS projects), development of manufacturing infrastructure, management of the production line, quality management, and so forth.

We next describe how CMMI® engineering process areas can be applied in BTS and BTP manufacturing projects, and then we discuss how other process areas, namely, project management process areas, support process areas, and organizational process areas, are applied.

Applying Engineering Process Areas in Manufacturing BTS Projects

Myth: CMMI®, especially the engineering process areas, is only for development, therefore it cannot be applied in BTS projects.

False! BTS projects, as explained earlier, include a significant development effort, and the application of CMMI® practices to this development effort can be very helpful. Here are some examples:

- *RD SG3: Analyze and Validate Requirements.* Simulation of product design and the production line: Using commercial SW simulation tools, validation of production line requirements can be performed before building the production line.
- *RD SP3.1: Establish Operational Concepts and Scenarios.* Operational concepts and scenarios in this case consist mainly of production line scenarios: workflow, workstations, assembly process, inventory buffers, timeline, and so forth.
- *VAL: Validate the Product.* Validation of the product prototype and production line using predefined scenarios. This process is tightly connected with RD SP1.2 and SP3.1.

Applying CMMI® in Manufacturing BTP Projects

Myth: CMMI®, especially the engineering process areas, is only for development or BTS projects, therefore it cannot be applied in BTP projects.

False! BTP projects, as explained earlier, may include development of manufacturing infrastructure, and the application of CMMI® practices to this development effort can be very helpful. Here are some examples:

- *RD SP1.1: Elicit Needs.* Analysis of product design and stakeholders' needs and expectations, including constraints, problems, and feedback from manufacturing and manufacturing engineering departments.
- *RD SG3: Analyze and Validate Requirements.* Simulation of the production line and manufacturing infrastructure prior to mass production.
- *TS SP3.2: Develop Product Support Documentation.* Includes production line support documentation.
- *PI SP3.4: Package and Deliver.* Same as in BTS projects, packaging and delivering the product.
- *VAL: Validate the Product.* Validation of the production line using predefined scenarios.
- *TS SP1.2 + RD SP3.1: Operational Concepts and Scenarios.* Manufacturing infrastructure operational scenarios are established and maintained.
- *PP and PMC: Plan and Monitor the Project.* Product engineering management and mass production management in the same mode as in the BTS projects, thus more emphasis is put on management of the latter.

Applying Project Management Process Areas in Manufacturing Processes

Manufacturing project aspects, like production engineering management, mass production management, and manufacturing infrastructure (assembly line, tools, testing equipment, jigs, and so on), are managed similar to developmental projects. Basic elements of project management such as WBS, cost estimation, stakeholders' involvement management, milestones, and progress reviews are usually similar. However, special attention should be given to some practices:

- *PP SP1.1: Estimate the Scope of the Project.* The project scope will always include manufacturing activities, but there may be some developmental activities.
- *PP SP1.3: Define the Project Life Cycle.* A manufacturing project life cycle is usually very different from a developmental project life cycle. It also usually includes phases such as "production line setup," manufacturing cycle, and production line maintenance.
- *IPM SP1.1: Establish the Project's Defined Process.* A manufacturing project may or may not be similar to other manufacturing projects in terms of the manufacturing processes involved. There may be families of manufacturing projects, with similar products and manufacturing processes within each such family, but big differences from one family to the other. Unlike developmental projects, where the standard processes are basically similar, manufacturing processes may be very different, depending on the manufactured product. This will, of course, impact the implementation of OPD as well.
- *QPM SP 1.2 Compose the Defined Process.* This practice is similar to IPM SP1.1, except that here the defined process definition, as composed by selection of processes and subprocesses, is based on performance. Although this concept is relatively new in developmental projects, it is well known and used in manufacturing. Therefore, we believe that applying this practice and the other specific practices of QPM will seem more natural in manufacturing projects than in developmental projects.

Applying Organizational Process Management Process Areas in Manufacturing Processes

Organizational process management process areas are not implemented at the project level, but at the organizational level. In our case, organizational process management process areas have some special flavor when they address manufacturing processes:

- *OPD SP1.1: Establish Standard Processes.* As mentioned earlier, manufacturing projects have their own standard processes, and they may or may not be similar to other manufacturing projects in terms of the manufacturing processes involved. There may be families of manufacturing projects, with similar products and manufacturing process within each such family, but with big differences from one family to the other. Unlike developmental projects, where the standard processes are basically similar, manufacturing processes may be very different, depending on the manufactured product. Therefore, the organization has to identify all types of manufacturing processes, according to its products, and define its standard manufacturing processes accordingly.
- *OPP SP1.1 Select Processes.* This practice requires the organization to select processes and subprocesses that will be included in the process performance

analysis. In a production-oriented organization, this selection should include mainly manufacturing processes, such as production line setup and quality management processes for manufacturing.

- *OPP SP 1.4 Establish Process Performance Baselines*. Again, process performance baselines in a production-oriented organization should provide quantitative data on the performance of manufacturing processes. Typical information may include production rate, manufactured products' quality, and yield.
- *OPP SP 1.5 Establish Process Performance Models*. Process performance models in a production-oriented organization should enable prediction of manufacturing processes' performance and product quality from early and in-process indicators, such as effort invested in quality control activities and inventory level at each production stage.
- *OID SP 1.1 Collect and Analyze Improvement Proposals*. Improvement proposals will naturally focus on improving aspects of the manufacturing process, such as improving the production rate or yield.
- *OID SP 1.3 Pilot Improvements*. Any improvement suggested must first be tested in a carefully designed pilot, before it is applied to the entire production line of a product or family of products.
- *OID SP 2.2 Manage the Deployment*. Managing the deployment should deal with questions such as these: To which products and production lines should the selected improvement be applied? When should it be applied? Who will make the required changes and how?
- The implementation of this practice in manufacturing processes is very similar to its implementation in developmental projects, but here the results usually can be observed and measured much more rapidly, thus the entire OID cycle is much faster. Of course, as usual, the implementation of this practice is closely related to measuring, analyzing, and recording process performance, as required in OPP.

Applying Support Process Areas in Manufacturing Processes

Support process areas consist of the MA, PPQA, CM, DAR, and CAR process areas. Implementation of PPQA, CM, and DAR in manufacturing processes is very similar, if not identical, to their implementation in development. However, implementation of MA and CAR deserves special attention:

- *MA SP1.1: Establish Measurement Objectives*. Measurement objectives must be aligned with the production line and the manufacturing process needs. Usually measurement objectives provide information on the production status, on how to make manufacturing most efficient, and how to obtain the level of quality and uniformity needed.
- *CAR SP1.1: Select Defect Data for Analysis*. Defect data to be analyzed will be such that finding the cause of the defects and eliminating it will be most beneficial to reaching the project's quality goals as well as other commitments.

Because most operations on the production line are repeatable, finding and correcting the cause of defects may influence the project performance very much, and thus affect the organization performance much more than doing this in developmental projects. As a matter of fact, this type of activity is very common in manufacturing-oriented organizations—much more so than in development-oriented ones.

An Example of OID Implementation

Kaizen events are performed in the manufacturing process. Here is a summary of an event that was performed on the subprocess of packaging:

- *Purpose:* Reduce work hours required for this subprocess from 53 to 45 per part.
- *Method:*
    - Observation, mapping, and measurement of the actual work process and the consumed amount of work hours;
    - Analysis of the work flow using brainstorming, the Five Whys, and so forth;
    - Calculation of the work hours after the process improvement;
    - Implementation of the improvements in the work process;
    - Measurement of the new amount of consumed work hours after the improvement at the end of 1 month of the implementation, to verify stability.
- *Result:* The process was improved and the target of reducing the amount of work hours—from 53 to 46—was achieved.

Using CAR and OPP principles the process of packaging was improved. Of course, all of this was possible due to the available standard processes (OPD). OID was implemented to measure and monitor the improvement.

An Example of CAR Implementation

In the BTP projects the manufacturing processes are constantly monitored, and failures or bad performance are recorded and analyzed using CAR principles.

In the manufacturing of a titanium beam for the F15 aircraft, the process of drilling was not performed according to the defined route card instructions; therefore, the result was nonorthogonal holes.

- *Purpose:* Correction of the drilling process to prevent rejection of parts and repair (rework).
- *Motivation:* Loss of money due to rework, employee dissatisfaction and frustration due to inefficiency, and low customer survey grades.

- Method (Kaizen event):
  - Observation, mapping, and measurement of the actual drilling process;
  - Examination of the used tools (drilling tools and drilling aids, blankets, and rulers) and procedures;
  - Analysis of the findings and solutions consolidation;
  - Correction of the drilling tools to prevent repetition of failures;
  - Observation, mapping, and measurement of improved drilling process and the consumed amount of work hours;
  - Repeated observation and process measurement at the end of 1 month to verify implementation, stability, and tuning of the corrected drilling process (measurement of the number of rejects, number of consumed work hours, employee satisfaction).
- *Result:* The drilling process was improved, the number of rejects was reduced to zero, the amount of work hours went according to the plan, and no rework was reported. Employees were very satisfied with the improvements and even recommended further observations in other workstations to detect other improvement opportunities.

Summary

CMMI® can be applied to manufacturing projects:

- Usually all process areas are applicable and can appropriately be interpreted for manufacturing processes. However, when some process areas seem to be nonapplicable for manufacturing, the continuous representation can be used.
- Although manufacturing processes are very different from development processes, they are still part of the organizational set of standard processes and, therefore, can be included in the scope of the appraisal.
- *When manufacturing activities represent a significant portion of the activities in the organization, they should be included in the CMMI® implementation and appraisal scope, due to their significant contribution to the business.*
- Some process areas are easier and more natural to implement in manufacturing, such as:
  - *Measurement and analysis.* Quality, number of defects, and production rate, for instance, are measured. Due to the large population, the sample rate is higher than in development, therefore allowing better analysis over a shorter range of time.
  - *Causal analysis and resolution.* Events and defects are studied and analyzed using various methods such as kaizen events, Top 10, fishbone diagrams, and the Five Whys.
  - *Quantitative project management.* SPC is used to monitor and control key characteristics, allowing adjustment and tuning of critical manufacturing subprocesses.

  - *Organization process performance*. Production capability and quality of manufacturing products.

- When an organization starts to implement CMMI® practices and plans CMMI® process appraisals, it should not refrain from including "nondevelopmental" processes such as manufacturing in the scope, especially when these processes are responsible for a significant amount of the organization's revenue.

The new CMMI® architecture now allows for different CMMI® models for different areas of application. At the moment, three models are defined: CMMI® for Development (CMMI®-DEV), CMMI® for Acquisition (CMMI®-ACQ), and CMMI® for Services (CMMI®-SVC). An additional CMMI® for Manufacturing model would fit very well in this architecture. The core process areas defined for CMMI®-DEV, CMMI®-ACQ, and CMMI®-SVC are also relevant and meaningful for manufacturing processes. In addition, some processes that are very important in manufacturing can be covered by some additional special process areas. These processes would include areas such as logistic support, production line management, inventory management and just-in-time concepts, repeatability, reducibility, and production capability.

PART IX

Summary

CHAPTER 26

Summary and Concluding Remarks

The Intent of *Practical Insight into CMMI®, Second Edition*

This intent of this book is to provide the reader with an insight into what activities an organization would be engaged in and what the role of each level of management and the practitioners would be if their systems, hardware, and software engineering processes were based on the CMMI®-DEV v1.2. I have tried to capture the essence of each of the process areas by presenting them in a practical context without the technical structure of the CMMI® masking the valuable nuggets of information. I have tried to provide some of the engineering and management principles behind the CMMI® goals and practices. The application of the CMMI® must be pragmatic. It must support an organization's business objectives with measurable results. It should not be used as a guide to passing the "test."

The emphasis of the CMMI® is to support organizational development processes and changing of cultures to show a measurable benefit for the organization's business objectives and vision. The CMMI® provides a framework from which to organize and prioritize engineering, people, and business activities. It supports the coordination of multidisciplined activities that are or may be required to successfully build a project and it returns the concept of "engineering systems think" to project development. Process improvement is about supporting an organization's business with measurable results. The CMMI® is about providing the necessary guidance to support successful organizational process improvement.

Essence of the CMMI®

The essence of the CMMI®-DEV v1.2 remains simple. It is project management, quality management, and engineering, glued together by process management. Although the concept of the CMMI® is simple, the authors have provided details and guidance that are not found in other related models and standards. It is the details of the CMMI® in its front-end material, its description of institutionalization factors, its introductory notes, and its goals, practices, subpractices, and examples that make the CMMI® a one-of-a-kind process improvement model in the world today.

Choosing Between the Staged and Continuous CMMI® Representations

Given the two representations in the CMMI®, an organization may feel it has to choose a process improvement path from either the process area capability approach or the organizational maturity approach. However, after using the CMMI® to conduct workshops, conducting CMMI® assessments, and helping organizations to get their CMMI®-based process improvement program going, it is my belief that both the staged representation and the continuous representation not only can but must be used together to provide a realistic approach that results in effective process improvement to support an organization's business objectives.

High Maturity

I have tried to present a view into the upgraded understanding of high maturity that the SEI is presenting to the world. What I have presented was only meant to be a "toe in the water." To be a high-maturity organization, process improvement leaders must understand the organization's vision, business objectives, and measurement objectives. They must study many more books and articles. They must try processes out and determine their value. They must get advice from experts in statistical control and related areas such as Six Sigma. They must train, mentor, and coach their managers, project members, and collaborators to get the most out of their organization's high-maturity efforts. To be awarded a CMMI® ML 4 or ML5, I think an organization should be measurably world class. The effort to achieve CMMI® ML 4 or ML 5 is lost if it is approached as a minimum effort to satisfy the practices of the CMMI®.

Support for a Process Improvement Initiative

An organization should be engaged in process improvement efforts if it wishes to be competitive, stay in business, and be regarded as an industry leader. To do this, an organization's process improvement initiative must be supported at many levels throughout the organization. Figure 26.1 shows the CMMI® as part of a series of cascading umbrellas that can guide an organization in determining what it needs to enable a successful process improvement journey.

- The CMMI® provides the overarching framework to provide the guidance for the process improvement initiative. The concepts found in the CMMI® can and should be complemented with other models and standards such as ISO 9001:2000, Six Sigma, ISO 15504 (SPICE), PMI, ITIL, Lean, Agile, and SCRUM.
- The classes of SCAMPI appraisals can be used to help an organization determine its current process capability.
- An action plan must be created to help the organization prioritize its process improvement efforts, provide adequate resources, and support the business objectives.

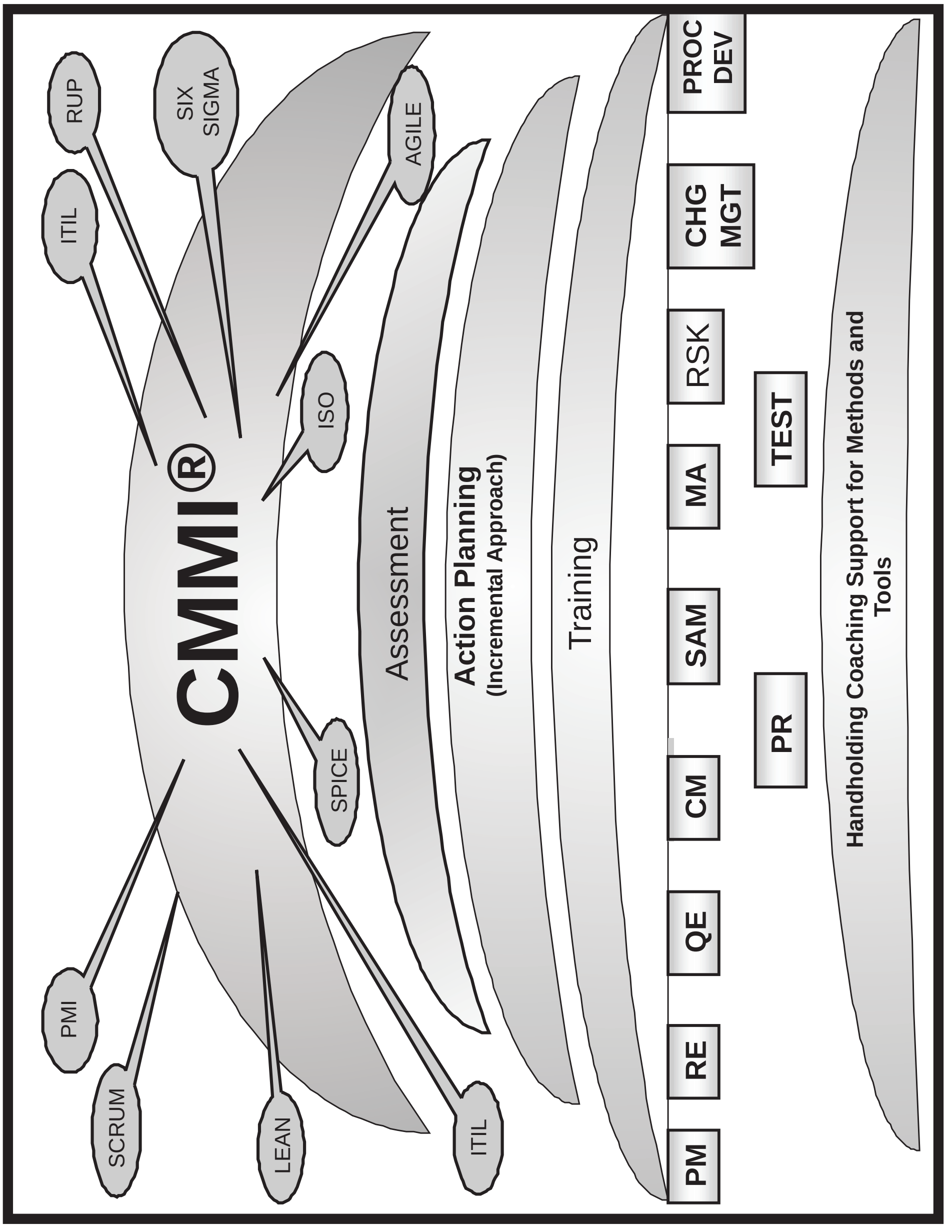


Figure 26.1 Support for a process improvement initiative.

- Training, mentoring, and coaching in the engineering and management principles behind the practices of the CMMI® are essential. The CMMI® is not an engineering handbook. It tells us we need to analyze our requirements; it does not say how. It tells us we must have a configuration management system; it does not tell us which one.
- "Hand-holding" coaching and support must be available to the organization's critical projects' members to ensure that the methods and techniques are implemented in an effective manner.

Process and Business

Process defines how a business does business. Combinations of process may be needed for an organization to be successful in its business:

- Software engineering processes;
- Hardware engineering processes;
- Systems engineering processes;
- Manufacturing processes;
- Financial processes;
- Human resources processes;
- Legal processes.

Process helps to establish the business culture and then sets guidelines and expectations. Process can be viewed as a methodology that is applied from elicitation of requirements to design through delivery. There are no shortcuts—there are no other alternative methods that a business can adopt that embraces a "cradle to grave" philosophy to ensure quality and profitability with *control* every step of the way. Because there are inherent costs to implementing process, members of senior management must demonstrate their belief in it through their communications, daily decision making, and financial commitment. Senior management's resolve must not waiver when deadlines beg for shortcuts to get the product out the door.

Process is the fastest, yet lowest cost path to get there and know if you are there!

Process improvement based on the framework provided by the CMMI® is the best approach an organization can take!

APPENDIX A
Sticky Areas

Kasse Initiatives has extensive assessment and appraisal experience worldwide. After conducting assessments based on the CMM® and CMMI® for the past 18 years and in 15 countries, it is our experience that there are classic "sticky areas" that have caused many organization's problems and appraisal teams' late nights, grief, and frustration. These "sticky areas" are discussed in this appendix.

Project Management

- Size and complexity estimation (lines of code, function points, gates on a gate array, number of interfaces, and so forth);
- Having "historical data" to support the estimation;
- Making estimations based on hours is *not* size estimation;
- Understanding the importance of the WBS for project management and for supplier management;
- Commitment process (see SG3 on commitment and reconciliation of differences between the estimates and the available resources);
- Stakeholders are not "just the project team members." They include both internal and external groups such as quality assurance, configuration management, regulatory agencies, customers, end users, independent test groups, and so forth.

Configuration Management

- Status accounting (supporting project managers, verification, and evaluation of system completeness);
- Configuration auditing:
  - Baseline auditing;
  - Functional configuration audits;
  - Physical configuration audits.
- Release notes or version description document.

Data Management

- Data management could actually include all technical data, all configuration management data, and data such as status reports for a project.
    - Data are the various forms of documentation required to support a program in all of its areas including administration, engineering, configuration management, financial, logistics, quality, safety, manufacturing, and procurement.
    - Data can take any form including reports, manuals, notebooks, charts, drawings, specifications, files, e-mail, or other forms of correspondence.

Figure A1.1 shows some example data stores that support projects.

Quality Assurance

- Ensures that objective evaluations can be made and escalation to the organization's highest level is possible.
- Project quality plan describes how the "quality functions" are going to be handled on the project throughout the project life cycle and by whom. The project quality plan does not contain the details of quality audits, configuration management, testing, peer reviews, and so forth.
- The quality assurance plan is a separate plan from the project quality plan that describes the quality function support the quality assurance representatives or group is going to provide to the project.
- Quality assurance and quality control are different focuses on quality. Quality control is exemplified by peer reviews and testing on life-cycle products. Quality assurance focuses on process and product compliance.

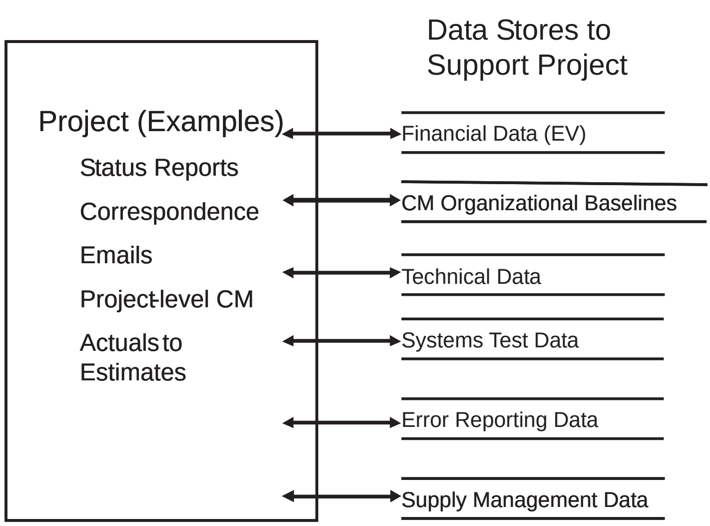


Figure A1.1 Data management stores.

Peer Reviews (Inspections and Structured Walkthroughs)

- *Design review.* A formal, documented, comprehensive, and systematic examination of a design to evaluate the design requirements and the capability of the design to meet those requirements and to identify problems and propose solutions. Design reviews would match up to Technical Reviews in the IEEE Standard 1028 on reviews and audits.
- *Peer review.* The review of work products performed by peers during development of the work products to identify defects for removal.
  - The level of formality (inspections, structured walkthroughs, walkthroughs, buddy checks, and so on) should be linked to criticality of the components or modules or subsystems being developed. Status accounting reports from the Configuration Management PA indicating the change in baselined elements is also a major input to the determination of the formality required from a peer review.
  - Maturity Level 3 companies normally conduct formal inspections and structured walkthroughs.

Unit Testing

- Should be based on a unit test plan that is connected to the overall master test plan.
- Expected results should be a part of the unit test plan.
- Actual results should be compared to the expected unit test results and these comparisons spot-checked by QA.
- Should check for unintended inputs, not just ones that are guaranteed to work with the program logic.
- *Point of contention.* A typical response from developers and project managers is that unit testing is up to the developers. They test to show the component or module works. Yes, but is that adequate if the component is critical red or critical yellow? If the component is critical red, a formal unit test plan, test scenarios, test data identification, expected results, and wrong input checks may all be required. A requirement of an independent unit test team may be required for some critical components. If the component or module is critical yellow, having a unit test "checklist plan" and an approval signature from the team leader may be adequate. Business objectives come into play again and strongly.

Interface Testing

- Integration is not just about conducting systems testing but testing the interfaces to ensure that the system skeleton works so that rigorous systems testing including functional testing, stress and boundary testing, and quality factors testing can be accomplished.

Measurement and Analysis

- Business objectives that demand measures outside of the classic project management size, cost, schedule, staffing, and so forth;
- Goal–question–metric paradigm;
- Storing measures along with the context for consistent interpretation and use later;
- ML2 and ML3 measurement demands that support ML4:
  - Process measures such as:
    - Effort;
    - Earned value;
    - Peer review effectiveness;
    - Testing effectiveness;
    - Test coverage;
    - Cycle time;
    - Defect removal efficiency;
    - Defect escape rates;
    - Number and density of defects (by severity) found during the first year following product delivery;
    - Rework time as a percentage of total project life-cycle time.
  - Product quality measures such as:
    - Critical resource utilization;
    - Throughput;
    - Reliability;
    - Defect density;
    - Response time;
    - Reliability;
    - Maintainability;
    - Interoperability;
    - Functionality;
    - Accuracy;
    - Problems per user month (PUM);
- Segmenting or stratifying data (creating categories of data instead of just counts).

Risk Management

- Risk management culture (making it part of project management);
- Developing risk management strategy and documenting why that strategy was used;
- Understanding what risk mitigation is and when to apply risk mitigation procedures—risk mitigation does not normally start right away, but is used if the risk becomes too close to the problem threshold;

- Understanding contingency plans and understanding when they should be applied and if they are cost effective.

Supplier Management

- SCOPE and applicability of supplier management (internal and external to the project's boundaries;
- Supplier selection criteria;
- Supplier agreement;
- Managing the progress and performance of the supplier;
- Transferring the product component from the suppler to the buyer;
- Auditing an organization's supplier before, during, and after contract award;
- Auditing selected supplier processes for those suppliers who have a critical part of the buyer's system;
- Auditing selected supplier work products as an indicator of the functionality and quality that will be delivered by the supplier.

Engineering Process Areas

- The CMMI® engineering process areas came from the systems engineering CMM® or EIA-731 based on EIA-632, Processes for Engineering a System.
- Requirements Development:
  - Elicitation of requirements using combinations of multiple techniques;
  - Capturing requirements along with:
    - Verification criteria;
    - Relationships between requirements;
    - Understanding the source of the requirements;
    - Determining the "why" of the requirement.
  - Analysis techniques including viewpoint analysis, data flow analysis, and objective-oriented analysis. These techniques should focus on improved understanding of the operational concept and the building of operational scenarios to refine existing requirements.
  - Early-phase validation of requirements using comprehensive methods that are equivalent to what might have been used for elicitation of the requirements to gain confidence that what is being developed will work in the operational environment. This loop back to the customer or stakeholders must happen at least once if the task is to build the product from the beginning, add a function to an existing product line, build to specification, or build to print.
- Technical Solution:
  - Determining alternative solutions even if the project is predominantly software.

- Requirements Management:
  - Bidirectional traceability;
  - Requirements traceability from requirements to related requirements to work products and so forth;
  - Keeping the life-cycle work products consistent through possible updates when a requirements specification change is approved (REQM–CM).
- Product Integration:
  - Ensuring that the product components match the current configuration *and* match the latest version controlled interface description before integration and testing of interfaces;
  - Testing of interfaces.
- Verification:
  - This is also done throughout the entire product life cycle. Verification does include testing and in some circumstances a test could both serve for verification and validation;
  - Capturing "as-run" process descriptions to use for possible process improvement.
- Validation:
  - End product validation and understanding the end user profiles;
  - Capturing "as-run" process descriptions to use for possible process improvement.
- Decision Analysis and Resolution (related closely to TS):
  - DAR is not necessary for "given" situations. Example: A Level 1 manager can sign for up to $25,000. A level 2 manager can sign for up to $50,000.
  - DAR is applicable to Technical Solution, Supplier Management, choosing an alternative architecture, moving into a new business market, and so forth.

APPENDIX B

SAM-ISM Starterkit for Supplier Evaluation Criteria

This Starterkit can be used as input to defining your organization's or project's checklist for supplier evaluation. This Starterkit is not complete and not exhaustive. Other categories and criteria may be necessary in your environment.

Business Orientation and Background

- Business objectives and vision;
- Company background;
- Organizational structure;
- Company's developmental plans:
    - Business targets,
    - Changes in ownership;
- Single or multisite;
- Size and stability of the company;
- Supplier's investment in research and development/track record in past;
- Supplier's customer profiles;
- Global presence to be able to support buyer in different parts of the world;
- Legal issues;
- Other customers (big ones, competing ones, supplier's competitors);
- How suppliers are managed;
- Corporate structure (compatible management structure);
- Company networks/cooperation around the world;
- Ease of access (communication and location);
- Ethical principles;
- Willingness to share entrepreneurial risks;
- Taxes and local laws;
- Motivation to work with buyer;
- Availability of resources (currently and ability to hire in future, turnover rate);
- References to check for actual supplier performance;
- Supplier's network worldwide;

- Production capability;
- Research and development experience;
- Integrity—confidentiality in handling information;
- Environmental policy;
- Strategy ambitions;
- Tendering practices.

Financial

- Financial background;
- Financial soundness of company (currently and future estimation);
- Shareholders;
- Stock market pressure;
- Ability to deal with financial milestones;
- Growth curves for past 1 to 2 years and projected growth curves for the next 1 to 2 years;
- Immune to buyouts by competitors.

Management

- Management structure—how are decisions made;
- Communication mechanisms within the supplier's company and their effectiveness;
- Overall spirit—personnel turnover;
- Attitude toward quality and process improvement.

Core Competencies

- Technical competency, process competency;
- Competence development program;
- Learning ability:
    - Attitude toward learning,
    - Training, mentoring, and coaching opportunities;
- Skills of the proposed project members and of the company as a whole;
- Language skills (ability to speak English or specific language of project);
- Documentation language (ability to write in English);
- Established culture that supports personal growth as well as technical growth;
- Training program (organizational and project level, opportunities and capabilities).

Process and Quality Management

- Quality culture;
- Process in use and evidence;
- Processes compatible with buyer's processes;
- Configuration management functions;
- Quality management functions;
- Testing methodology capability (unit, integration, systems);
- Senior management support for supplier delivery of high-quality products and services;
- Security;
- Safety;
- Scalability;
- Documentation level and quality;
- Peer review culture.

Experience Level

- Experience in software development;
- Similar project experience;
- Previous cooperation with buyer or like companies, history;
- Currently being used elsewhere in the company;
- Handling of intellectual property;
- Guarantees.

Project Management

- Transparent estimations—historical data, models, and/or formulas for estimation calculation are easy to see and understand;
- Project tracking—evidence that supplier routinely tracks actuals against the estimates found in the project plan;
- Risk management and contingency plans;
- Project or product focused;
- Working environment/tools;
- Tools compatibility with buyer;
- Expectations of buyer—for example, tools given by buyer or the use of incompatible tools between the buyer and supplier;
- Status reporting.

Certifications and Awards

- CMM®/CMMI® maturity level/ISO and other certifications;
- Brand/reputation;
- Awards and success stories;
- Readiness for audits or capability evaluations.

Orientation to Buyer

- Names and roles of people explicitly named and CVs;
- Ability to handle reasonable amounts of volume;
- Ability to commit—what other commitments does the organization have at the same time?
- Application domain experience—software domain availability;
- Certifications or readiness for audits;
- Willingness to share risks and rewards with buyer;
- Priority of buyer for supplier;
- Key personnel profiles and availability;
- Service and maintenance support (help desk, service-level agreement for maintenance);
- Cost/price versus quality of products and services;
- Flexibility;
- Relocation of foreign people in new cultural environment;
- Permission, visas, work permits needed;
- Travel ability.

APPENDIX C

Quantitative Management Vocabulary

Common cause of process variation The variation of a process that exists because of normal and expected interactions among the components of a process.

Defined process A managed process that is tailored from the organization's set of standard processes according to the organization's tailoring guidelines; has a maintained process description and contributes work products, measures, and other process improvement information to the organizational process assets.

Predictability A process is said to be predictable when, through the use of past experience, we can describe, at least within limits, how the process will behave in the future. An unpredictable process will display exceptional variation that is the result of special causes. A predictable process will display routine variation that is characteristic of common causes.

Process measurement The set of definitions, methods, and activities used to take measurements of a process and its resulting products for the purpose of characterizing and understanding the process

Process performance A measure of actual results achieved by following a process. It is characterized by both process measures (effort, cycle time, defect removal efficiency) and product measures (reliability, defect density, response time).

Process performance baseline A documented characterization of the actual results achieved by following a process. A measurement of performance for the organization's set of standard processes. Several process performance baselines can be used to characterize performance for subgroups of the organization.

Process performance model A description of the relationships among attributes of a process and its work products. A process performance model is used to estimate or predict the value of a process performance measure from the values of other process, product, and service measurements. Process performance models typically use process and product measurements collected throughout the life of the project to estimate progress toward achieving objectives that cannot be measured until later in the project's life.

Quality and process performance objectives Objectives and requirements for product quality, service quality, and process performance.

Quantitative management Quantitative management uses data from statistical management to help the project predict whether it will be able to achieve its quality and process-performance objectives and identify what corrective action should be taken.

Quantitative objective Desired target value expressed as quantitative measure.

Quantitatively managed process A defined process that is controlled using statistical and other quantitative techniques. The product quality, service quality, and process performance attributes are measurable and controlled throughout the project.

Special cause of process variation A cause of a defect that is specific to some transient circumstance and not an inherent part of a process.

Stable process The state in which all special causes of process variation have been removed and prevented from recurring so that only the common causes of process variation of the process remain.

Standard process An operational definition of the basic process that guides the establishment of a common process in an organization. A standard process describes the fundamental process elements that are expected to be incorporated into any defined process. It also describes the relationships (e.g., ordering and interfaces) among these process elements.

Statistical process control Statistically based analysis of a process and measurements of process performance, which will identify common and special causes of variation in the process performance, and maintain process performance within limits.

Statistically managed process A process that is managed by a statistically based technique in which processes are analyzed, special causes of process variation are identified, and performance is contained within well-defined limits.

Statistical control Statistical control is based on collecting and analyzing process and product measures so that special or assignable causes of variation can be identified and addressed to achieve predictable performance. Statistical control supports statistical management.

Statistical management Statistical management involves statistical thinking and the correct use of a variety of statistical techniques, such as control charts, confidence intervals, prediction intervals, and tests of hypotheses.

Selected Bibliography

Ahern, D., A. Clouse, and R. Turner, *CMMI Distilled: A Practical Introduction to Integrated Process Improvement*, Reading, MA: Addison-Wesley, 2001.

Basili, V. R., *Software Modeling and Measurement: The Goal Question Metric Paradigm*, Computer Science Technical Report Series CS-TR-2956 (UMIACS-TR-9296), College Park, MD: University of Maryland, September 1992.

Basili, V. R., G. Caldiera, and H. D. Rombach, "Goal Question Metric Paradigm," in *Encyclopedia of Software Engineering,* Vol. 1, New York: John Wiley & Sons, 1994, pp. 528–532.

Bate, R., et al., *Systems Engineering Capability Maturity Model, Version 1.1 (SE-CMM)* (CMU/SEI-95-MM-003), Pittsburgh, PA: Software Engineering Institute, 1995.

Bouldin, B. M., *Agents of Change*, Englewood Cliffs, NJ: Yourdon Press, 1989.

Brassard, M. and D. Ritter, *The Memory Jogger II*, Salem, NH: GOAL/QPC, 1994.

Burr, A. and M. Owen, *Statistical Methods for Software Quality*, Boston, MA: International Thomson Computer Press, 1996.

Bush, M., "Modern Software Assurance and a Five-Level Model of Software Assurance Maturity," *High Integrity Systems,* Vol. 1, No. 2, 1994, pp. 157–169.

Charette, R. N., *Software Engineering Risk Analysis and Management*, New York: McGraw-Hill, 1989.

Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI®, Guidelines for Process Integration and Product Improvement,* 2nd ed., Reading, MA: Addison-Wesley, 2007.

CMMI® Product Development Team, *Capability Maturity Model Integration (CMMI), CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing Version 1.1* (CMMI-SE/SW/IPPD/SS, V1.1), *Continuous Representation* (CMU/SEI-2002-TR-011, ESC-TR-2002-011), Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, March 2002.

CMMI® Product Development Team, *CMMI® for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1 Staged Representation* (CMU/SEI-2002-TR-012, ESC-TR-2002-012), Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, March 2002.

Crosby, P. B., *Quality Is Free*, New York: McGraw-Hill, 1979.

Curtis, B., W. Hefley, and S. Miller, *People Capability Maturity Model,* Reading, MA: Addison-Wesley, 2002.

Dalziel, M. M., and S. C. Schoonover, *Changing Ways,* New York: AMACOM, 1988.

Daskalantonakis, M., "A Practical View of Software Measurement and Implementation Experience with Motorola," *IEEE Software,* Vol. 18, No. 11, 1992, pp. 998–1010.

Daughtrey, T., *Fundamental Concepts for the Software Quality Engineer*, Milwaukee, WI: American Society for Quality Press, 2002.

Deming, W. Edwards, *Out of the Crisis*, Cambridge, MA: MIT Center for Advanced Engineering Study, 1986.

Deutsch, M., and R. Willis, *Software Quality Engineering: A Total Technical and Management Approach*, Upper Saddle River, NJ: Prentice-Hall, 1988.

Dowson, M., *Proceedings of the First International Conference on the Software Process Manufacturing Complex Systems*, IEEE Computer Society Press, October 21–22, 1991.

Ebenau, R. G., and S. H. Strauss, *Software Inspection Process,* New York: McGraw-Hill, 1993.

Electronic Industries Alliance*, EIA Interim Standard: National Consensus Standard for Configuration Management* (EIA/IS-649), Washington, D.C.: EIA, 1995.

Electronic Industries Alliance*, EIA Standard: EIA 632 Standard, Processes for Engineering a System* (ANSI /EIA/IS-632-1998), Washington, D.C.: EIA, 1998.

Electronic Industries Alliance, *Systems Engineering Capability Model* (EIA/IS-731), Washington, D.C.: EIA, 1998.

Evans, M., and J. Marciniak, *Software Quality Assurance & Management*, New York: John Wiley & Sons, 1987.

Federal Aviation Administration ICMM Project Team, *Federal Aviation Administration Integrated Capability Maturity Model (FAA-ICMM) for the Acquisition of Software Intensive Systems*, Washington, D.C.: FAA, 1997.

Ferguson, J., et al., *Software Acquisition Capability Maturity Model® (SA-CMM®) Version 1.01* (CMU/SIE-96-TR-020), Pittsburgh, PA: Software Engineering Institute, December 1996.

Florac, W. A., and A. D. Carleton, *Measuring the Software Process: Statistical Process Control for Software Process Improvement,* Reading, MA: Addison-Wesley, 1999.

Fowler, Priscilla, and Stan Rifkin, *Software Engineering Process Group Guide* (CMU/SEI-90-TR-24), Pittsburgh, PA: Software Engineering Institute, September 1990.

Freedman, D. P., and G. H. Weinberg, *Handbook of Walkthroughs, Inspections, and Technical Reviews,* Boston, MA: Little, Brown and Company, 1977.

Gilb, T., *Software Metrics*, Bromley, England: Winthrop, 1977.

Gilb, T., and D. Graham, *Software Inspections,* Reading, MA: Addison-Wesley, 1993.

Hall, E. M., *Managing Risk—Methods for Software Systems Development*, Reading, MA: Addison-Wesley, 1998.

Hetzel, W., *The Complete Guide to Software Testing*, Wellesley, MA: QED Information Sciences, 1988.

House, R. S., *The Human Side of Project Management*, Reading, MA: Addison-Wesley, 1988.

Humphrey, W. S.*, Managing the Software Process*, Reading, MA: Addison-Wesley, 1989.

IEEE Standard for Industry implementation of International Standards, *IEEE Guide to Classification of Software Anomalies* (IEEE STD 1044.1-1995), New York: IEEE, 1995.

IEEE Standard for Industry Implementation of International Standards, *IEEE Standard for Software Reviews* (IEEE Std. 1028-1997), New York: IEEE, 1997.

IEEE Standard for Industry implementation of International Standards, *Standard for Information Technology* (ISO/IEC 12207: 1995, includes IEEE/EIA Stds 12207.0-1996, 12207.1-1997, and 12207.2-1994), New York, IEEE, 1995.

International Organization for Standardization, *ISO 9000: Quality Management ISO Standards Compendium,* Geneva: ISO, 1994.

International Organization for Standardization, *ISO 9001: Quality Management Systems—Requirements,* Geneva: ISO, 2000.

International Organization for Standardization and International Electrotechnical Commission, *Information Technology—Software Life Cycle Processes* (ISO/IEC TR 12207), Geneva: ISO, 1995.

International Organization for Standardization and International Electrotechnical Commission, *Information Technology—Software Process Assessment* (ISO/IEC TR 15504), Geneva: ISO, 1998.

International Organization for Standardization and International Electrotechnical Commission, *Software Engineering—Software Measurement Process* (ISO/IEC TR 15939), Geneva: ISO, 2002.

Juran, J. M., *Juran on Planning for Quality*, New York: Macmillan, 1988.

Kan, S. H., *Metrics and Models in Quality Software Engineering,* 2nd ed., Reading, MA: Addison-Wesley, 2003.

Kaplan, R. S., and D. P. Norton, *The Balanced Score Card,* Boston, MA: Harvard Business School Press, 1996.

Kasse, T., *Action Focused Assessment for Software Process Improvement*, Norwood, MA: Artech House, 2002.

Kasse, T., *CMMI® Workshop*, Phoenix, AZ: Kasse Initiatives LLC, 2000.

Kasse, T., *Configuration Management Workshop*, Plano, TX: Kasse Initiatives LLC, 2007.

Kasse, T., *Project Management for Systems Engineers Workshop*, Plano, TX: Kasse Initiatives LLC, 2007.

Kasse, T., *Requirements Engineering Workshop*, Phoenix, AZ: Kasse Initiatives LLC, 2007.

Kasse, T., *Risk Management Workshop*, Phoenix, AZ: Kasse Initiatives LLC, 2007.

Kasse, T., *Software Quality Engineering Workshop*, Phoenix, AZ: Kasse Initiatives LLC, 2007.

Kasse, T., *Supplier Management Workshop*, Plano, TX: Kasse Initiatives LLC, 2007.

Kasse, T., *Systems Engineering Workshop*, Phoenix, AZ: Kasse Initiatives LLC, 2007.

Kezsborn, D. S., and K. A. Edward, *The New Dynamic Project Management, Winning Through the Competitive Advantage*, New York: John Wiley & Sons, 2001.

Kotonya, G., and I. Sommerville, *Requirements Engineering: Processes and Techniques*, Chichester, England: John Wiley & Sons, 1998.

Kulpa, M. K., and Kent A. Johnson, *Interpreting the CMMI—A Process Improvement Approach*, 2nd ed., Boca Raton, FL: Auerbach Publication, a CRC Press Company, 2008.

Military Standard: Defense System Software Development (DOD-STD2167), Washington, D.C.: U.S. Department of Defense, 1985.

Military Standard: Defense System Software Development (DOD-STD2167A), Washington, D.C.: U.S. Department of Defense, 1988.

Military Standard: Software Development and Documentation (DOD-STD498), Washington, DC: U.S. Department of Defense, 1994.

Moti, F., "Engineering Systems Thinking and Systems Thinking," *Systems Engineering,* Vol. 3, No. 3, New York: John Wiley & Sons, 2000, pp. 163–168.

Mutafelija, B., and H. Stromberg, *Systematic Process Improvements Using ISO 9001:2000 and CMMI®*, Norwood, MA: Artech House, 2003.

Ogden Software Development Activity, *Peer Reviews Training Course,* (MSG/SO) Hill Air Force Base, Ogden, UT, 1996.

Oosterlking, H., *Software Process Improvement with ING Group Technical Report*, London: European Software Engineering Process Group, June 1998.

Paulk, M., C. *Basic Statistics for Software Engineers,* Pittsburgh, PA: Software Engineering Institute, June 2001.

Paulk, M. C., et al., *Capability Maturity Model for Software (Version 1.1)* (CMU/SEI-91-TR-24), Pittsburgh PA: Software Engineering Institute, August 1991.

Paulk, M. C., et al., *The Capability Maturity Model®: Guidelines for Improving the Software Process*, Reading, MA: Addison-Wesley Longman, 1995.

Paulk, M. C., et al., *Key Practices of the Capability Maturity Model®, Version 1.1* (CMU/SEI-93-TR-25), Pittsburgh PA: Software Engineering Institute, February 1993.

Paulk, Mark C., Mary Beth Chrissis, and Charles V. Weber, *Capability Maturity Model for Software, Version 1.1* (CMU/SEI-93-TR-24), Pittsburgh PA: Software Engineering Institute, February 1993.

Pressman, R. S., *Software Engineering,* New York: McGraw-Hill, 1987.

Radice, R., *High Quality Software Inspections,* Andover, MA: PARADOXICON Publishing, 1995.

Radice, R., et al., "A Programming Process Architecture," *IBM Systems Journal,* Vol. 24, No. 2, 1985.

Rakitin, S. R., *Software Verification and Validation for Practitioners and Managers*, Norwood, MA: Artech House, 2001.

Sematech, Software Inspections Training Course, Technology Transfer 92061180B-TRG, 1992.

Shaw, M., "Prospects for an Engineering Discipline of Software," *IEEE Software,* Vol. 7, No. 6, November 1990, pp. 15–24.

Shaw, M., et al., *What a Software Engineer Needs to Know* (CMU/SEI-89-TR-30, DTIC Number ADA219064), Pittsburgh, PA: Software Engineering Institute, 1989.

Sheard, S., *The Frameworks Quagmire, A Brief Look,* Quag 14d SPC, 1998.

Shewhart, W. A., *Economic Control of Quality of Manufactured Product*, New York: Van Nostrand, 1931.

Software Engineering Institute, *The Capability Maturity Model®: Guidelines for Improving the Software Process*, Reading, MA: Addison-Wesley, 1995.

Software Engineering Institute, *Integrated Product Development Capability Maturity Model®, Draft Version 0.98*, Pittsburgh, PA: Enterprise Process Improvement Collaboration and Software Engineering Institute, Carnegie Mellon University, July 1997.

Software Engineering Institute, *Software CMM®, Version 2.0 (Draft C)*, October 1997.

Software Technology Support Center, *Peer Reviews,* 1996.

Stagliano, A. A., *Rath and Strong's Six Sigma Advanced Tools Pocket Guide*, New York: McGraw-Hill, 2004.

Trillium: Telecom Software Product Development Process Capability Assessment Model, Draft 2.2, Bell Canada, July 1992.

U.S. Air Force, *Software Development Capability/Capacity Review*, ASD Pamphlet 800-5, September 10, 1987.

Weber, C. V., et al., *Key Practices of the Capability Maturity Model®* (CMU/SEI-91-TR-25), Pittsburgh, PA: Software Engineering Institute, August 1991.

Wheeler, D. J., *Understanding Variation: The Key to Managing Chaos,* Knoxville, TN: SPC Press, 2000.

Young, R., *The Requirements Engineering Handbook,* Norwood, MA: Artech House, 2004.

About the Author

Tim Kasse serves as the CEO and principal consultant of Kasse Initiatives LLC. He also serves as a principal consultant at Delta Axiom, an international process improvement alliance company headquartered in Denmark.

Mr. Kasse spent 4 years at the Software Engineering Institute (SEI). He was a major contributor to the development of the Capability Maturity Model® for Software and led the evolution of the SEI's assessment method that was commercialized in October 1990. From 1999 to 2000 Mr. Kasse led the development of the SEI's Intermediate CMMI® Workshop for Lead Assessors. Mr. Kasse has been authorized by the SEI to conduct SCAMPI℠ Class A, B, and C appraisals. He has participated in more than 100 process assessments and consulting engagements in 25 countries throughout North America, South America, Europe, Asia, and the Middle East.

Mr. Kasse is the architect of the Action Focused Assessment that has been applied in major organizations throughout the world. He is the author of the books *Action Focused Assessment for Software Process Improvement* (Artech House, 2002) and *Practical Insight into CMMI®* (Artech House, 2004). Mr. Kasse is the primary developer of many Kasse Initiatives workshops including workshops on supplier management, quality engineering, configuration management, risk management, systems engineering, and change management. He is a recognized speaker at major process improvement and quality management conferences around the world.

Mr. Kasse serves the SEI as a visiting scientist supporting the CMMI® through training and presentations worldwide. He holds the position of Visiting Fellow at the Institute of Systems Science, National University of Singapore. He holds a master's degree in computer science and a bachelor's degree in systems engineering and has more than 38 years of systems/software-related experience. His focus is on helping companies balance the achievement of business objectives with planned process improvement.

Index

M

N

O

P

Q

S