



Process Improvement Is Continuous Improvement

We can never reach perfection.

**The CMM does not provide all the answers;
it too is evolving and improving.**

**Process management means constructive
and continual improvement.**

The focus is on always doing better.

Our reach should always exceed our grasp.



Issues for CMM v2

Need to address:

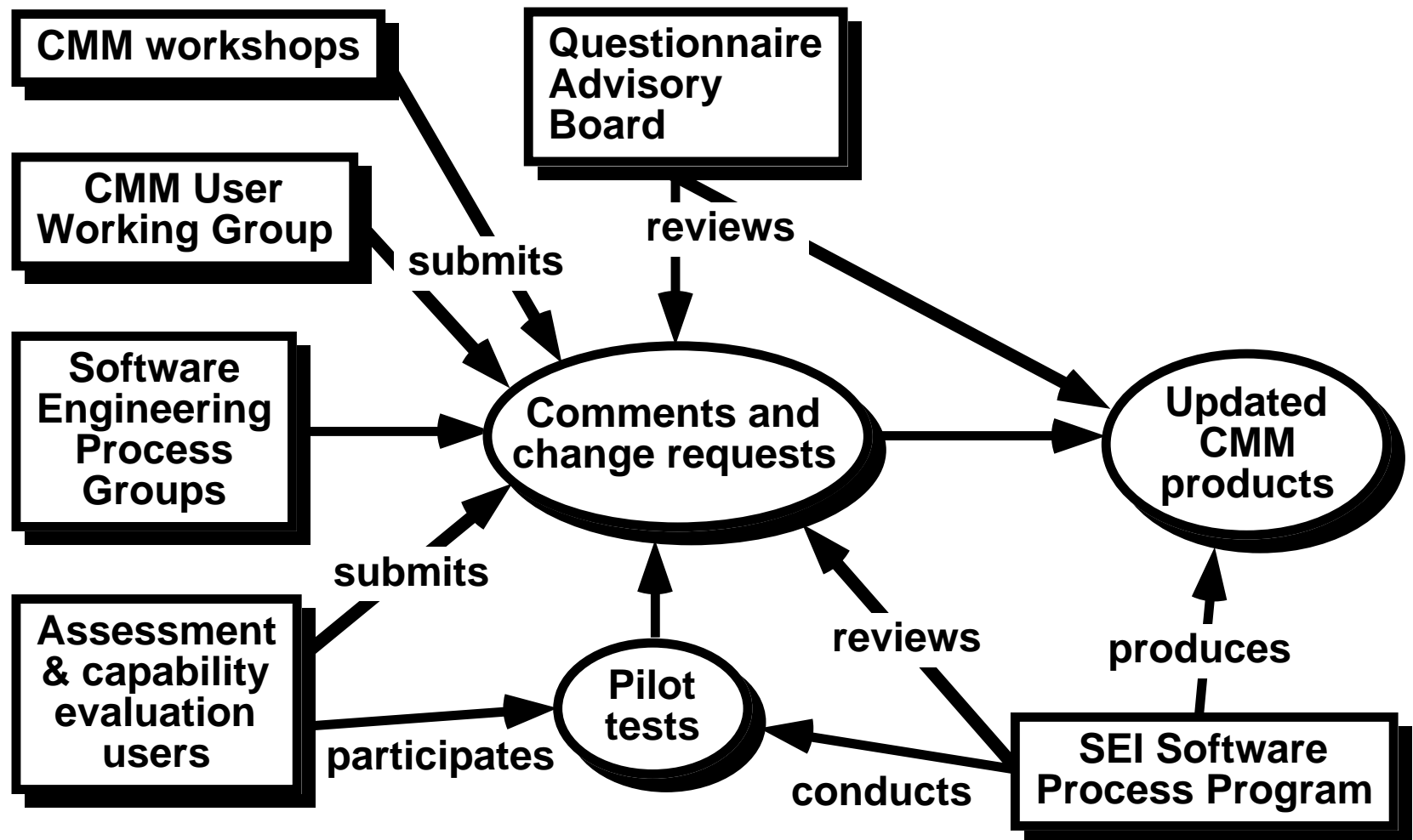
- **change requests and feedback from the CMM user community**
- **issues identified by ISO's SPICE project**

Consider restructuring key process areas to span maturity levels.

Expand the descriptions of Levels 4 and 5.



Public Review Process for CMM





Impact of the New CMM

"If my organization was assessed at Level X using the original model and preliminary maturity questionnaire, will it be assessed at Level X using the updated CMM and updated maturity questionnaire?"

- **If an organization based its improvement program on the *spirit* of the maturity model, there will probably be little impact on its assessed level.**
- **If an organization's improvement program is designed on the items in the preliminary maturity questionnaire, then the organization may see some impact.**

Maturity level scores have always been based on the findings of an assessment rather than the answers to the maturity questionnaire.



Releases of the CMM

Version 1.0 of the Capability Maturity Model for Software (CMM) released in August 1991

CMM v1.1 released in February 1993

CMM v2 planned for the 1996-1998 time frame



Process Improvement Using the CMM

Software process improvement occurs within the context of:

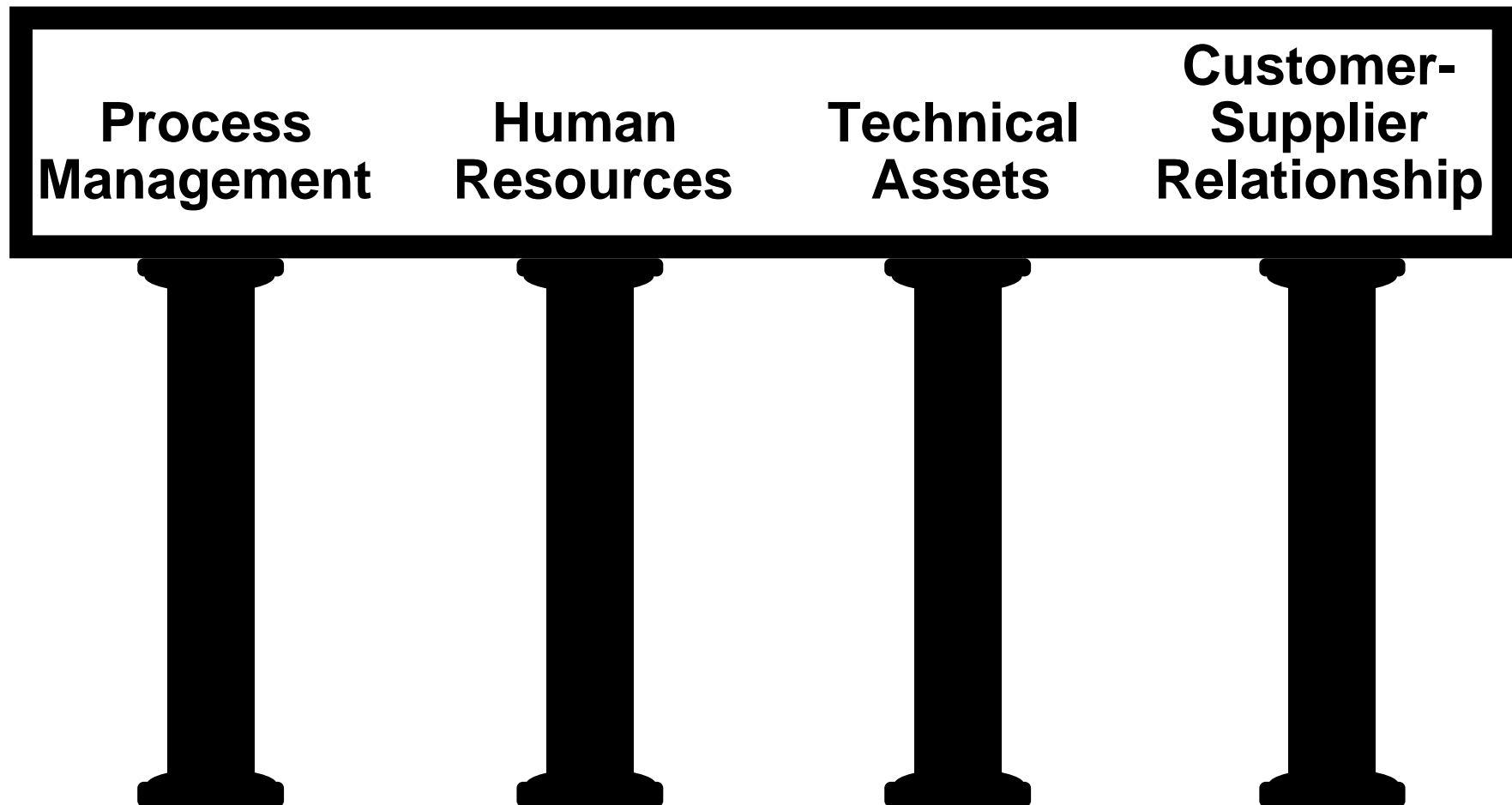
- **the organization's strategic plans**
- **its business objectives**
- **its organizational structure**
- **the technologies in use**
- **its social culture**
- **its management system**



Carnegie Mellon University

Software Engineering Institute

Process Management and the CMM in Context





CMM and Quality Management

The CMM does not address all the issues that need to be faced for software process and quality improvement.

Issues that are addressed only indirectly, or by implication, include:

- **specific tools, methods, and technologies**
- **concurrent engineering and teamwork**
- **system engineering, marketing, etc.**
- **human resources**
- **change management**

View the CMM, assessments, and evaluations in the larger context.



CMM and Business Context

The CMM is an application of Total Quality Management principles to software engineering.

Emphasis should be on customer satisfaction.

The result should be higher quality software products produced by more competitive companies.



Using the CMM in Context

The key practices in the CMM are expressed in terms of a large government contracting organization.

When the business environment differs from that template, an appropriate interpretation of the practices should be made.

The true CMM "requirements" are the goals for achieving the key process areas.



Scope of the CMM: Using "Key"

The CMM is not exhaustive.

There are software management and engineering processes and practices that are not described in the CMM.

KEY indicates a focus on the major leverage points.



Using Higher Level Practices

Processes at higher maturity levels may be performed, although perhaps ineffectively, even by Level 1 organizations.

Peer reviews can help even a Level 1 project.

Building an organizational capability means institutionalizing good practices on a firm foundation.



Topics

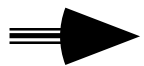
Introduction

The Capability Maturity Model

Understanding the Initial and Repeatable Levels

Understanding the Defined Level

Understanding the Managed and Optimizing Levels



Conclusion and Discussion



A Foundation, Not a Destination

The optimizing level is not the destination of process management.

The destination is better products for a better price: economic survival.

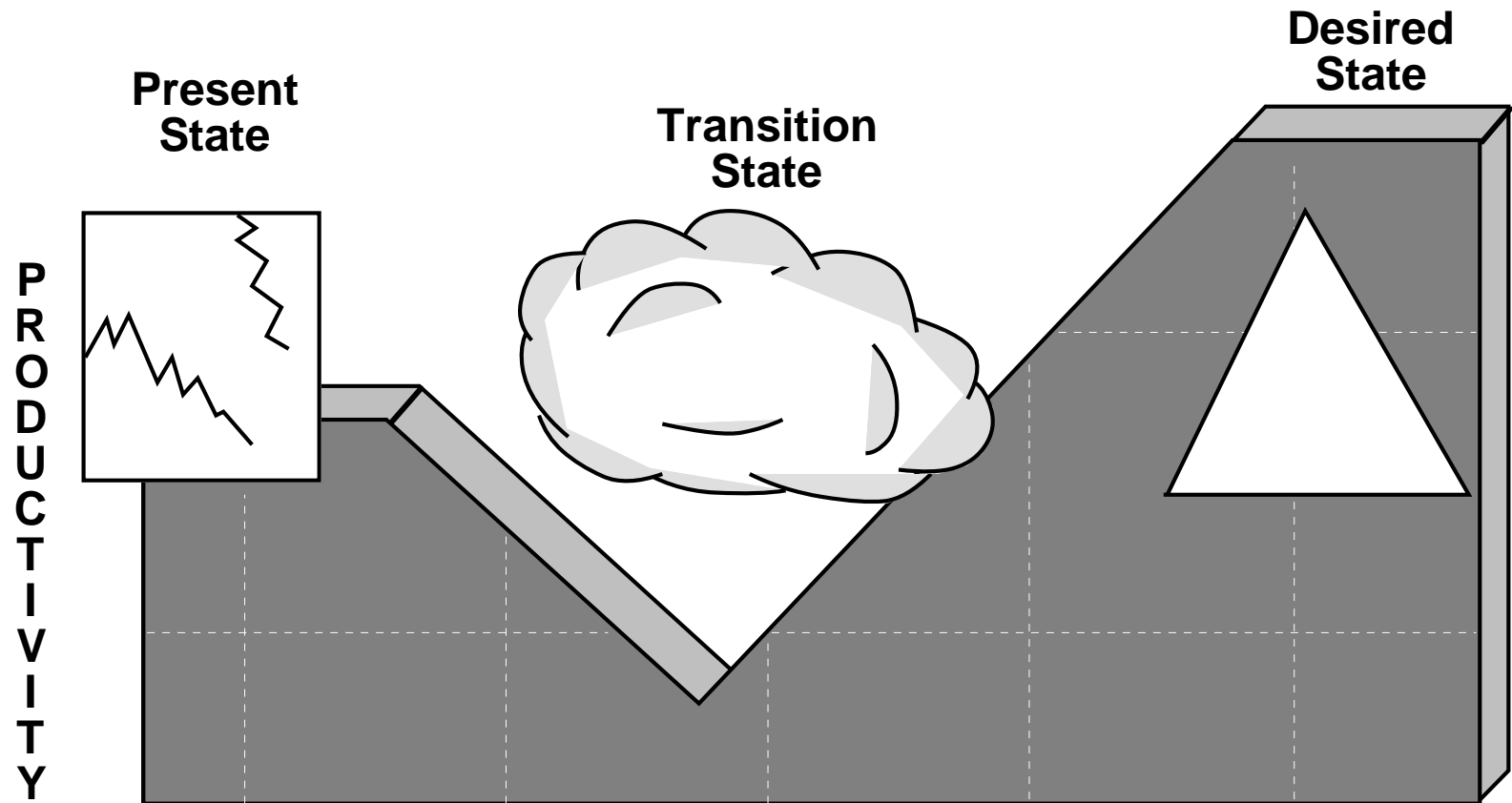
The optimizing level is a foundation for building an ever-improving capability.



Carnegie Mellon University

Software Engineering Institute

The Great Productivity Dip





Process Improvement Is a Lifestyle Change

Silver Bullet = Diet

95% of all dieters regain the weight they have lost ... and more ... within one year of a diet.

Process Improvement = Lifestyle Change

60% of those who change their lifestyle to eat less and exercise more maintain their weight loss.



Make Change a Normal Process

Recognize that there are no silver bullets.

Relate improvements to overall plans and goals.

Accept that improvement will come in small incremental steps.

Recognize that reactive changes generally make things worse.

Believe that crisis prevention is more important than crisis recovery.

Accept continuous improvement as a way of life.



Goals of PCM

- 1. Continuous process improvement is planned.**
- 2. Participation in the organization's software process improvement activities is organization-wide.**
- 3. The organization's standard software process and the projects' defined software processes are improved continuously.**



Process Change Management (PC, PCM)

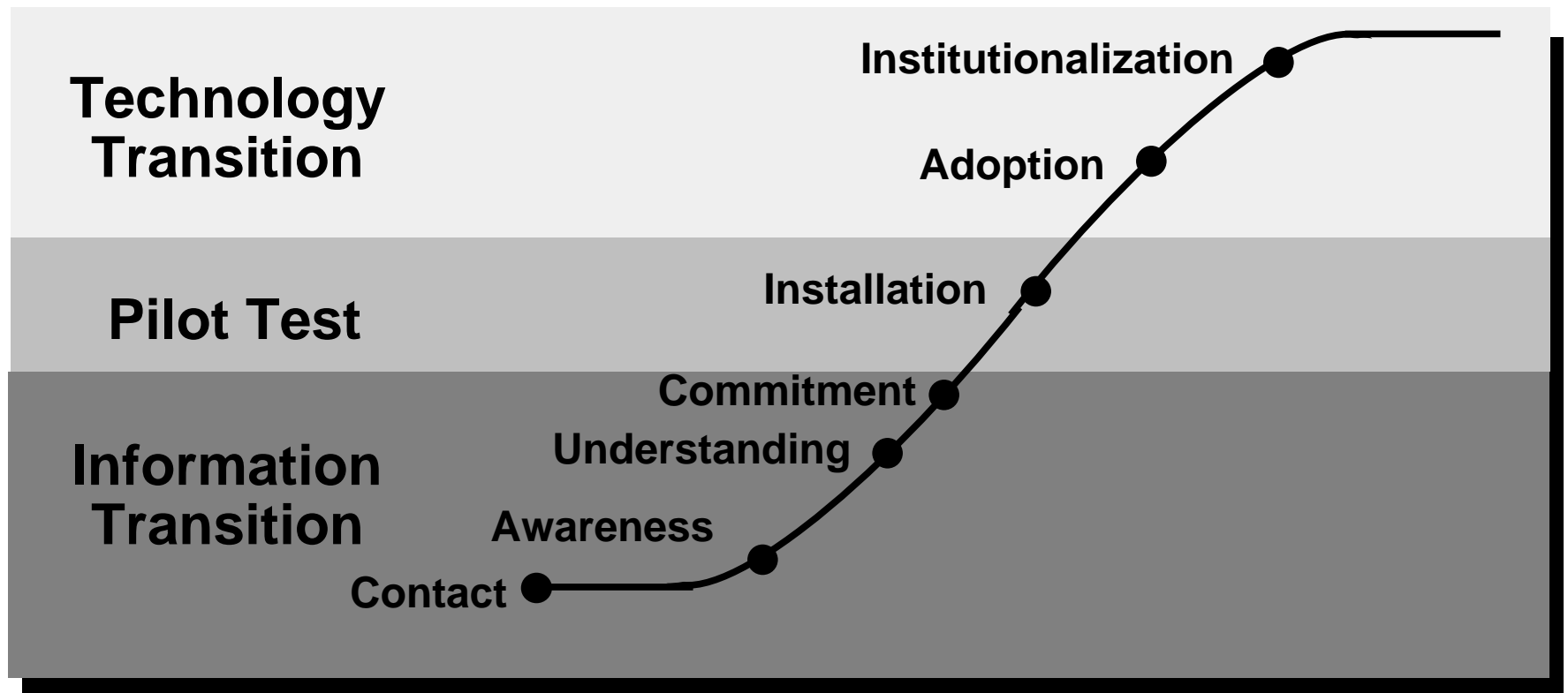
Purpose is to continually improve the software processes used in the organization with the intent of improving software quality, increasing productivity, and decreasing the cycle time for product development

Involves:

- **defining process improvement goals**
- **systematically identifying, evaluating, and implementing improvements to the organization's standard software process and the projects' defined software processes**



Technology Transfer Curve





Goals of TCM

- 1. Incorporation of technology changes is planned.**
- 2. New technologies are evaluated to determine their effect on quality and productivity.**
- 3. Appropriate new technologies are transferred into normal practice across the organization.**



Technology Change Management (TM, TCM)

Purpose is to identify new technologies (i.e., tools, methods, and processes) and transfer them into the organization in an orderly manner

Involves:

- **identifying, selecting, and evaluating new technologies**
- **incorporating effective technologies into the organization**



Important Concepts in DP

Kickoff meetings confirm a common understanding of the processes to be followed.

Causal analysis functions best in a stable process.



Goals of DP

- 1. Defect prevention activities are planned.**
- 2. Common causes of defects are sought out and identified.**
- 3. Common causes of defects are prioritized and systematically eliminated.**



Defect Prevention (DP)

Purpose is to identify the cause of defects and prevent them from recurring

Involves:

- **analyzing defects that were encountered in the past**
- **taking specific actions to prevent the occurrence of these types of defects in the future**



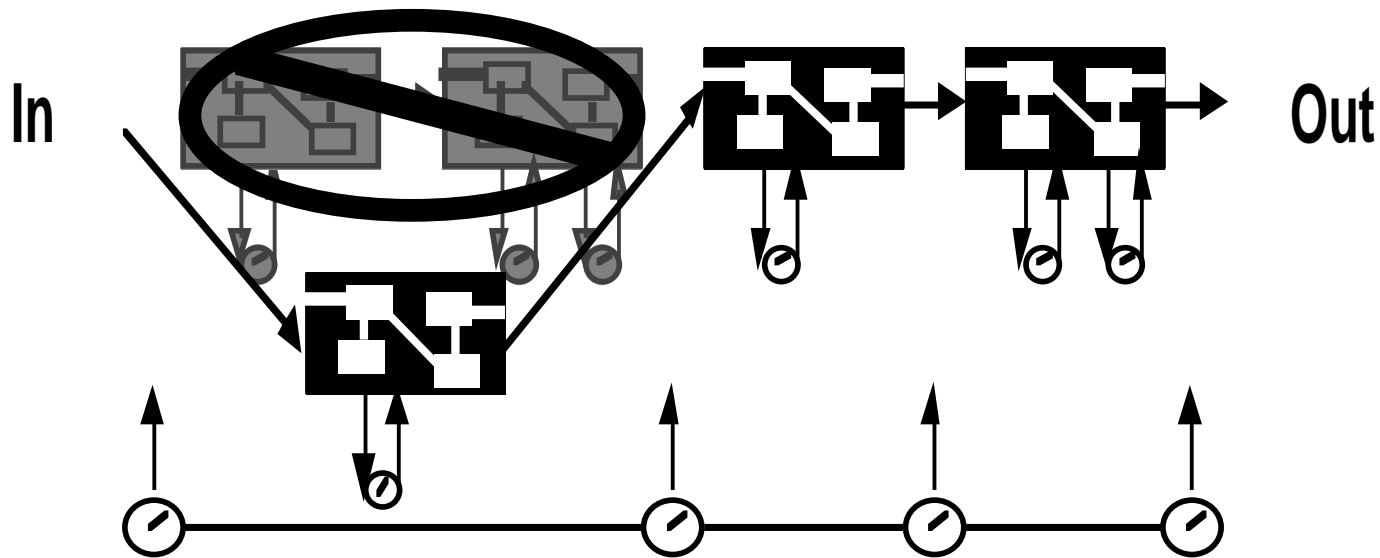
The Key Process Areas for Level 5

Optimizing (5)

**Process change management
Technology change management
Defect prevention**



The Management View of the Software Process at Level 5



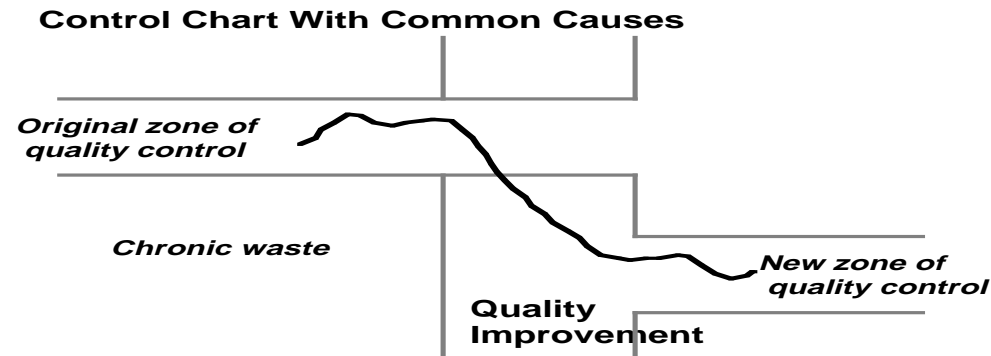
The software process is continuously improved in a controlled manner.



Understanding the Optimizing Maturity Level

Automate, pilot new technologies, do technology transition.

Identify and eliminate chronic causes of poor performance.



Continually improve the software process.



Goals of SQM

- 1. The project's software quality management activities are planned.**
- 2. Measurable goals for software product quality and their priorities are defined.**
- 3. Actual progress toward achieving the quality goals for the software products is quantified and managed.**



Software Quality Management (QM, SQM)

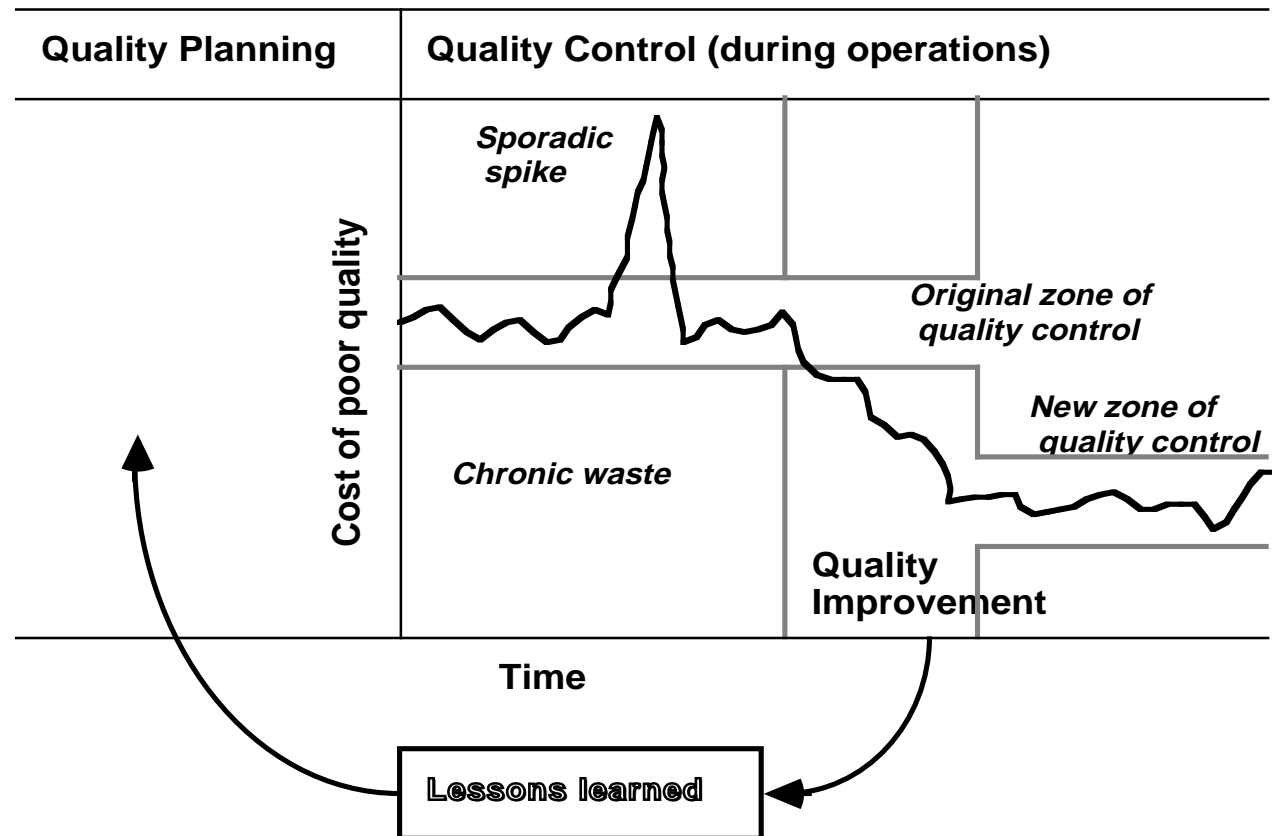
Purpose is to develop a quantitative understanding of the quality of the project's software products and achieve specific quality goals

Involves:

- **defining quality goals for the software products**
- **establishing plans to achieve these goals**
- **monitoring and adjusting software plans, software work products, activities, and quality goals to satisfy the needs and desires of customer and end-user**



The Juran Trilogy Diagram



J.M. Juran Wilton, CT

Used with the express permission of the Juran Institute, August 1990.



Some Quantitative Tools

Basic statistical process control tools include:

- **control charts**
- **cause-and-effect (fishbone) diagrams**
- **Pareto charts**
- **scatter diagrams**

Advanced tools include:

- **robust design (Taguchi)**
- **quality function deployment (QFD)**



Measurement Across Maturity Levels

Myth that measurement occurs only at Level 4

Level 5—improvement and cost/benefit data

Level 4—process and product quality data

Level 3—process data

Level 2—planning and tracking data

Level 1—haphazard data



Goals of QPM

- 1. The quantitative process management activities are planned.**
- 2. The process performance of the project's defined software process is controlled quantitatively.**
- 3. The process capability of the organization's standard software process is known in quantitative terms.**



Quantitative Process Management (QP, QPM)

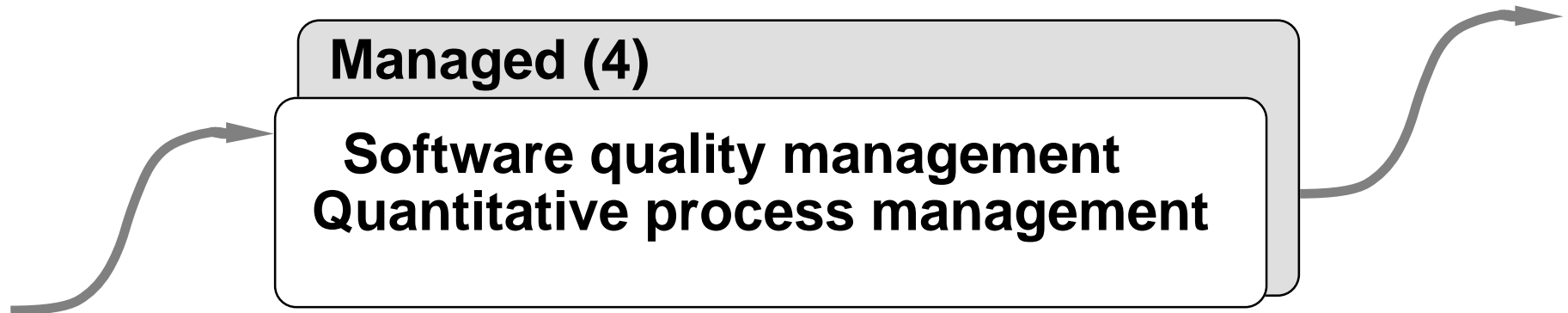
**Purpose is to control the process
performance of the software project
quantitatively**

Involves:

- **establishing goals for process performance**
- **measuring the performance of the project**
- **analyzing these measurements**
- **making adjustments to maintain process performance within acceptable limits**

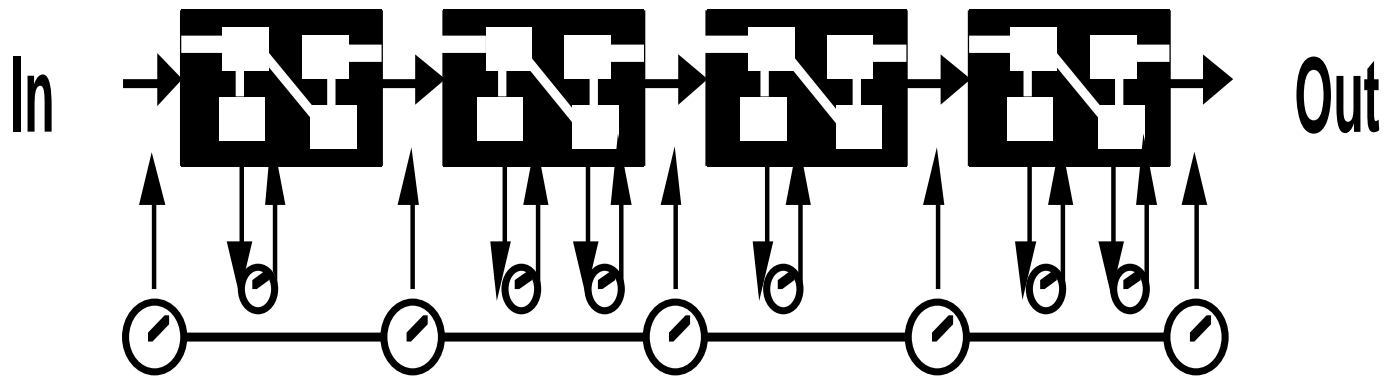


The Key Process Areas for Level 4





The Management View of the Software Process at Level 4

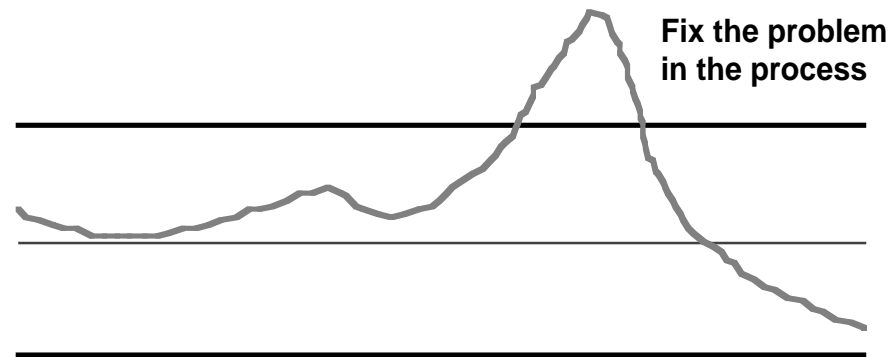


The production of the software product is quantitatively understood throughout the software process.



Understanding the Managed Maturity Level

Applying the principles of statistical process control, address special causes of process variation.



Control Chart With Special Causes

Explicitly address the customer's needs as part of a philosophy of quality management.



Topics

Introduction

The Capability Maturity Model

Understanding the Initial and Repeatable Levels

Understanding the Defined Level



Understanding the Managed and Optimizing Levels

Conclusion and Discussion



Implementations of PR

Possible alternative ways of implementing peer reviews include:

- **Fagan-style inspections**
- **structured walkthroughs**
- **active reviews**



Goals of PR

- 1. Peer review activities are planned.**
- 2. Defects in the software work products are identified and removed.**



Peer Reviews (PR)

Purpose is to remove defects from the software work products early and efficiently

Important corollary is to develop a better understanding of the software work products and of defects that might be prevented

Involve a methodical examination of work products by the producer's peers to identify defects and areas where changes are needed



Important Concepts in IC

Intergroup Coordination deals with the interface to groups beyond the software engineering group and possibly beyond the control of the organization doing software process improvement.

Examples of these groups, the interface to which must be managed, include:

- **systems engineering**
- **marketing**
- **training**

Intergroup Coordination is a first step on the road to concurrent engineering.



Goals of IC

- 1. The customer's requirements are agreed to by all affected groups.**
- 2. The commitments between the engineering groups are agreed to by the affected groups.**
- 3. The engineering groups identify, track, and resolve intergroup issues.**



Intergroup Coordination (IC)

Purpose is to establish a means for the software engineering group to participate actively with the other engineering groups so the project is better able to satisfy the customer's needs effectively and efficiently

Involves the disciplined interaction and coordination of the project engineering groups with each other to address system-level requirements, objectives, and plans



Important Concepts in SPE

Software Product Engineering includes:

- **software requirements analysis**
- **software design**
- **coding**
- **integration**
- **testing**

Software Product Engineering addresses the total software engineering environment.



Goals of SPE

- 1. The software engineering tasks are defined, integrated, and consistently performed to produce the software.**
- 2. Software products are kept consistent with each other.**



Software Product Engineering (PE, SPE)

Purpose is to consistently perform a well-defined engineering process that integrates all the software engineering activities to produce correct, consistent software products effectively and efficiently

Involves performing the engineering tasks to build and maintain the software using appropriate tools and methods



Goals of ISM

- 1. The project's defined software process is a tailored version of the organization's standard software process.**
- 2. The project is planned and managed according to the project's defined software process.**



Integrated Software Management (IM, ISM)

Purpose is to integrate the project's software engineering and management activities into a coherent, defined software process tailored from the organization's software process assets

Involves:

- **developing the project's defined software process by tailoring the organization's standard software process**
- **managing the software project according to this defined software process**



Important Concepts in TP

Training may include informal as well as formal vehicles for transferring skills and knowledge.

At Level 2, the phrase "receive training" is used. Training at Level 2 is not likely to have been institutionalized across the organization.

At Levels 3 and above, the phrase "receive required training" is used. Institutionalization of training is expected.



Goals of TP

- 1. Training activities are planned.**
- 2. Training for developing the skills and knowledge needed to perform software management and technical roles is provided.**
- 3. Individuals in the software engineering group and software-related groups receive the training necessary to perform their roles.**



Training Program (TP)

Purpose is to develop the skills and knowledge of individuals so they can perform their roles effectively and efficiently

Involves:

- **identifying the training needs of the organization, the projects, and individuals**
- **developing and/or procuring training to address these needs**



Library of Software Process-Related Documentation

Established to:

- **store process documents that are potentially useful to other current and future projects**
- **make them available for sharing across the organization**

Contains example documents and document fragments



Organization's Software Process Database

Established to collect and make available data on the software processes and resulting software work products

Contains or references both the actual measurement data and the related information needed to understand the measurement data and assess it for reasonableness and applicability



Tailoring Guidelines

Established to guide the software projects in:

- **selecting a software life cycle from those approved for use**
- **tailoring and elaborating the organization's standard software process and the selected software life cycle to fit the specific characteristics of the project**



Software Life Cycles

A software life cycle is the period of time that begins when a software product is conceived and ends when the software is no longer available for use.

One software life cycle may not be appropriate for all situations, given a variety of contractual and customer relationships.

An organization may identify more than one software life cycle for use by the projects.



Organization's Standard Software Process

The operational definition of the basic process that guides the establishment of a common software process across the software projects in the organization

Describes the fundamental software process elements that each software project is expected to incorporate into its defined software process

Describes the relationships (e.g., ordering and interfaces) between these software process elements (software process architecture)



Software Process Assets

Software process assets include:

- **the organization's standard software process (including the software process architecture and software process elements)**
- **the descriptions of software life cycles approved for use**
- **the guidelines and criteria for tailoring the organization's standard software process**
- **the organization's software process database**
- **the library of software process-related documentation**



Goals of OPD

- 1. A standard software process for the organization is developed and maintained.**
- 2. Information related to the use of the organization's standard software process by the software projects is collected, reviewed, and made available.**



Organization Process Definition (PD, OPD)

Purpose is to develop and maintain a usable set of software process assets that improve process performance and provide a basis for cumulative, long-term benefits

Involves developing and maintaining the organization's standard software process and related process assets



Important Concepts in OPF

The typical mechanism for providing a process focus for the organization is the Software Engineering Process Group (SEPG).

Other mechanisms are possible:

- **process review boards**
- **quality circles**
- **process steering committees**

These mechanisms may work in conjunction with, or in place of, an SEPG, depending on an organization's implementation of Organization Process Focus.



Goals of OPF

- 1. Software process development and improvement activities are coordinated across the organization.**
- 2. The strengths and weaknesses of the software processes used are identified relative to a process standard.**
- 3. Organization-level process development and improvement activities are planned.**



Organization Process Focus (PF, OPF)

Purpose is to establish the organizational responsibility for software process activities that improve the organization's overall software process capability

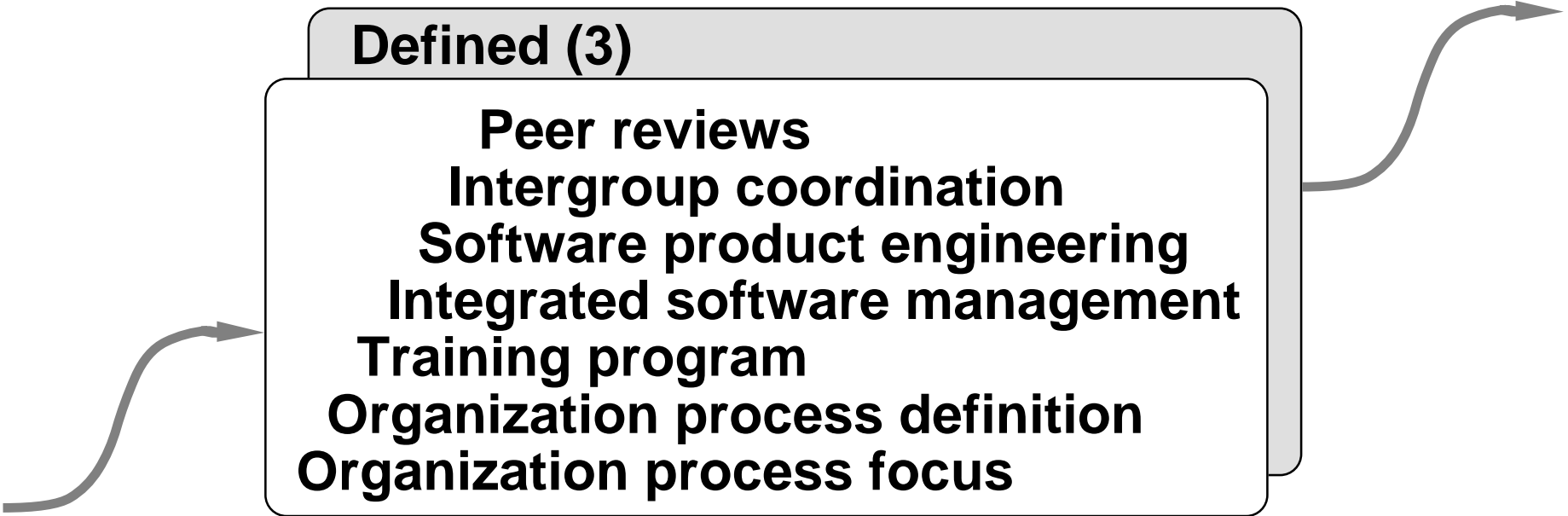
Involves:

- **developing and maintaining an understanding of organization and project software processes**
- **coordinating the activities to assess, develop, maintain, and improve these processes**



The Key Process Areas for Level 3

Defined (3)

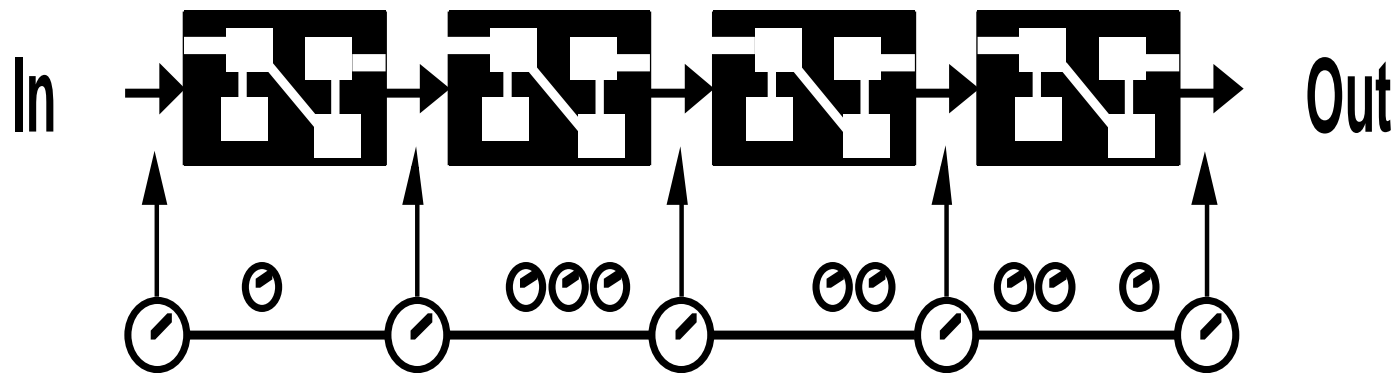


The diagram consists of a large, light-gray rounded rectangle with a darker gray header. The header contains the text 'Defined (3)'. Inside the rectangle, a list of seven key process areas is displayed in bold black text. A curved arrow enters from the left, pointing to the list, and another curved arrow exits from the right, pointing towards the top right corner of the slide.

- Peer reviews**
- Intergroup coordination**
- Software product engineering**
- Integrated software management**
- Training program**
- Organization process definition**
- Organization process focus**



The Management View of the Software Process at Level 3



Roles and responsibilities in the process are understood.

The production of the software product is visible throughout the software process.



Understanding the Defined Maturity Level

To control a process, it must be well understood.

Identify the inputs, how they will affect the process, and their readiness criteria.

Identify the outputs and the completion criteria for the outputs.

Establish a shared understanding of how the process works and the role of each participant.

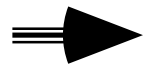


Topics

Introduction

The Capability Maturity Model

Understanding the Initial and Repeatable Levels



Understanding the Defined Level

Understanding the Managed and Optimizing Levels

Conclusion and Discussion



Managed and Controlled

Some software work products do not need the formality of configuration management but do need to be placed under some form of:

- **version control**
- **change control**

This is referred to as "managed and controlled" in the key practices.



Baseline Versus Developmental Configuration Management

BASELINE CONFIGURATION MANAGEMENT – establish baselines for identified software work products at predetermined points

DEVELOPMENTAL CONFIGURATION MANAGEMENT – control of the configuration exercised by the developers as they perform their work

The SCM key process area can be satisfied with baseline configuration management.



Goals of SCM

- 1. Software configuration management activities are planned.**
- 2. Selected software work products are identified, controlled, and available.**
- 3. Changes to identified software work products are controlled.**
- 4. Affected groups and individuals are informed of the status and content of software baselines.**



Software Configuration Management (CM, SCM)

Purpose is to establish and maintain the integrity of the products of the software project throughout the software life cycle

Involves:

- **identifying configuration items/units**
- **systematically controlling changes**
- **maintaining integrity and traceability of the configuration throughout the software life cycle**



Important Concepts in SQA

INDEPENDENCE – SQA should be independent of the software producers and project management

FITNESS FOR USE – SQA should provide feedback on the usability of the standards and procedures, as well as process fidelity



Goals of SQA

- 1. Software quality assurance activities are planned.**
- 2. Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.**
- 3. Affected groups and individuals are informed of software quality assurance activities and results.**
- 4. Noncompliance issues that cannot be resolved within the software project are addressed by senior management.**



Software Quality Assurance (QA, SQA)

Purpose is to provide management with appropriate visibility into the process being used and the products being built

Involves:

- **reviewing and auditing the software products and activities to ensure that they comply with the applicable procedures and standards**
- **providing the software project and other appropriate managers with the results of those reviews and audits**



Important Concepts in SM

PRIME CONTRACTOR – the organization responsible for building a system, that contracts out part of the work to another contractor, the **SUBCONTRACTOR**

Qualified does not mean "best technically qualified" or "highest process capability"

Factors other than process capability and technical ability influence the qualifications of subcontractors

- **strategic business alliances**



Goals of SM

- 1. The prime contractor selects qualified software subcontractors.**
- 2. The prime contractor and the software subcontractor agree to their commitments to each other.**
- 3. The prime contractor and the software subcontractor maintain ongoing communications.**
- 4. The prime contractor tracks the software subcontractor's actual results and performance against its commitments.**



Software Subcontract Management (SM)

Purpose is to select qualified software subcontractors and manage them effectively

Involves:

- **selecting a software subcontractor**
- **establishing commitments with the subcontractor**
- **tracking and reviewing the subcontractor's performance and results**



Goals of PTO

- 1. Actual results and performances are tracked against the software plans.**
- 2. Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans.**
- 3. Changes to software commitments are agreed to by the affected groups and individuals.**



Software Project Tracking and Oversight (PT, PTO)

Purpose is to provide adequate visibility into actual progress so that management can take effective actions when performance deviates significantly from the plan

Involves:

- **tracking and reviewing software accomplishments and results against documented estimates, commitments, and plans**
- **adjusting plans based on actual accomplishments and results**



Software Plans

SOFTWARE PLANS – the collection of plans, both formal and informal, used to express how software development and/or maintenance activities will be performed.

SOFTWARE DEVELOPMENT PLAN (SDP) – the collection of plans that describe the activities to be performed for the software project:

- governs the management of the activities performed by the software engineering group for a software project
- is not limited to the scope of any particular planning standard, such as DOD-STD-2167A and IEEE-STD-1058, which may use similar terminology



Making Commitments

COMMITMENT – a pact that is freely assumed, visible, and expected to be kept by all parties

PROJECT MANAGER – the role with total business responsibility for an entire project

SENIOR MANAGEMENT – management with a primary focus on the long-term vitality of the organization, rather than short-term project and contractual concerns and pressures



Process in Planning

PROCESS – a sequence of steps performed for a given purpose; a set of activities that achieve a desired result

METHOD – an approach to be followed in executing a process

PROCEDURE – a written description of a course of action to be taken to perform a given task

SOFTWARE TOOL – software that provides automated support for a method



Goals of SPP

- 1. Software estimates are documented for use in planning and tracking the software project.**
- 2. Software project activities and commitments are planned and documented.**
- 3. Affected groups and individuals agree to their commitments related to the software project.**



Software Project Planning (PP, SPP)

Purpose is to establish reasonable plans for performing the software engineering and for managing the software project

Involves:

- **developing estimates for the work to be performed**
- **establishing the necessary commitments**
- **defining the plan to perform the work**

Plan provides the basis for initiating the software effort and managing the work



Important Concepts in RM

CUSTOMER – may be external or internal

SYSTEM REQUIREMENTS – the customer's statement of the requirements

SYSTEM REQUIREMENTS ALLOCATED TO SOFTWARE (referred to in the CMM usually as *allocated requirements*) – the subset of the system requirements allocated to the software part of the system

SOFTWARE REQUIREMENTS – derived from software requirements analysis of the allocated requirements



Goals of RM

- 1. System requirements allocated to software are controlled to establish a baseline for software engineering and management use.**
- 2. Software plans, products, and activities are kept consistent with the system requirements allocated to software.**



Requirements Management (RM)

Purpose is to establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project

Involves establishing and maintaining an agreement with the customer on the requirements for the software project

Agreement is the basis for estimating, planning, performing, and tracking the project's software activities



The Key Process Areas for Level 2

Repeatable (2)

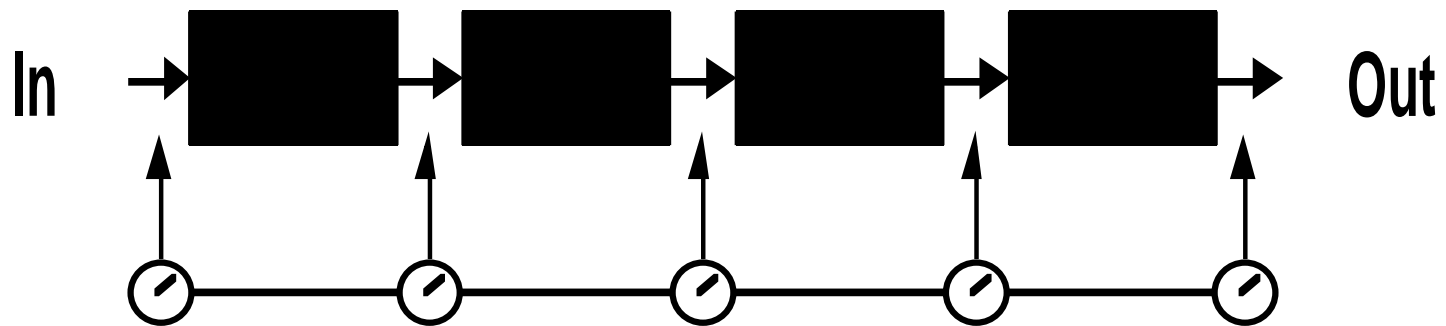


The diagram consists of a light gray rounded rectangle with a darker gray header. The header contains the text 'Repeatable (2)'. The main body of the rectangle is white and contains a list of six process areas. A curved arrow enters from the left, pointing to the white box. Another curved arrow exits from the right side of the gray header, pointing towards the top right corner of the slide.

- Software configuration management**
- Software quality assurance**
- Software subcontract management**
- Software project tracking and oversight**
- Software project planning**
- Requirements management**



The Management View of the Software Process at Level 2



Requirements and resources flow in.

The production of the software product is visible at defined points.

Artifacts of the process are controlled.



Understanding the Repeatable Maturity Level

Management must "walk the talk" to initiate an improvement effort.

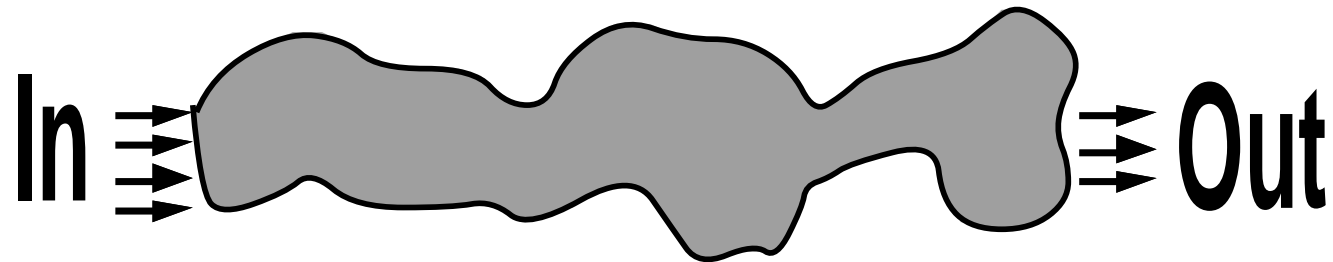
Only with management discipline will good software engineering practices be retained in the crunch.

Management processes establish role models for process improvement.

Management – and process – discipline empowers the engineering processes and the technical staff.



The Management View of the Software Process at Level 1



Requirements flow in.

A software product is (usually) produced by some amorphous process.

The product flows out and (hopefully) works.



A Myth: The Problems Are All Technical

Examined real cases

- **red teams**
- **assessment and evaluations**

Projects generally fail for management reasons

- **Defense Science Board Task Force on Military Software report, 1987**
- **"Bugs in the Program" report, 1989**

The major problems in software development are managerial - not technical.



Typical Level 1 Environments

"I'd rather have it wrong than have it late."

A senior software manager (industry)

"The bottom line is schedule. My promotions and raises are based on meeting schedule first and foremost."

A program manager (government)

"By regularly putting the development process under extreme time pressure and then accepting poor-quality products, the software user community has shown its true quality standard."

DeMarco and Lister (*Peopleware*, 1987)



Understanding the Initial Maturity Level

Performance driven by the competence and heroics of the people doing the work

Consistency and compliance to standards driven by management priorities - usually schedule is the top priority

High quality and exceptional performance possible so long as the best people can be hired

Unpredictability – for good or ill – characterizes the initial level organization



Topics

Introduction

The Capability Maturity Model

➡ Understanding the Initial and Repeatable Levels

Understanding the Defined Level

Understanding the Managed and Optimizing Levels

Conclusion and Discussion



A Well-Defined Process

A well-defined process can be characterized in terms of:

- **readiness criteria**
- **inputs**
- **standards and procedures for performing the work**
- **verification mechanisms (e.g., peer reviews)**
- **outputs**
- **completion criteria**



A Reasonable Process

A reasonable software process is:

- **practiced**
- **documented**
- **enforced**
- **trained**
- **measured**
- **able to improve**



An Example of Decomposing the CMM Structure

Maturity Level	Level 2 – Repeatable
Key Process Area	Software Project Planning
Goal	1. Software estimates are documented ...
Common Feature	Activities Performed
Key Practice	9. Estimates for the size of the software work products ...



An Example Key Practice: Size Estimating

Software Project Planning

Activity 9 **Estimates for the size of the software work products (or changes to the size of software work products) are derived according to a documented procedure:**

This procedure typically specifies that ...



Key Practices

The infrastructures and activities that contribute most to the effective implementation and institutionalization of a key process area



Verifying Implementation

Describes the steps to ensure that the activities are performed in compliance with the process that has been established

Typical subfeatures include reviews and audits by:

- **senior management**
- **project management**
- **software quality assurance**



Measurement and Analysis

Describes the need to measure the process and analyze the measurements

Typically includes examples of the measurements that could be taken to determine the status and effectiveness of the Activities Performed



Activities Performed

Describes the roles and procedures necessary to implement a key process area

Typical subfeatures include:

- **establishing plans and procedures**
- **performing the work**
- **tracking it**
- **taking corrective actions as necessary**



Ability to Perform

Describes the preconditions that must exist in the project or organization to implement the software process competently

Typical subfeatures include:

- **resources**
- **organization structure**
- **delegation**
- **training**
- **orientation**



Commitment to Perform

Describes the actions the organization must take to ensure that the process is established and will endure

Typical subfeatures include:

- **policies**
- **senior management sponsorship**
- **responsibility**



Institutionalize and Implement

The organization outlives those who leave it.

The organizational culture must convey the process.

Management must nurture the culture.

Culture is conveyed with role models and rewards.



Common Features

Attributes that indicate whether the implementation and institutionalization of a key process area is effective, repeatable, and lasting

Used to organize the key practices in each key process area

Common features are:

- **commitment to perform**
- **ability to perform**
- **activities performed**
- **measurement and analysis**
- **verifying implementation**



An Example of Goals: Software Project Planning

- 1. Software estimates are documented for use in planning and tracking the software project.**
- 2. Software project activities and commitments are planned and documented.**
- 3. Affected groups and individuals agree to their commitments related to the software project.**



Organization Responsibilities

The organization will have primary responsibility for acting on:

- **Software Quality Assurance**
- **Organization Process Focus**
- **Organization Process Definition**
- **Training Program**
- **Technology Change Management**
- **Process Change Management**



Project Responsibilities

The project will have primary responsibility for acting on:

- **Requirements Management**
- **Software Project Planning**
- **Software Project Tracking and Oversight**
- **Software Subcontractor Management**
- **Software Configuration Management**
- **Integrated Software Management**
- **Software Product Engineering**
- **Intergroup Coordination**
- **Peer Reviews**
- **Quantitative Process Management**
- **Software Quality Management**
- **Defect Prevention**



Responsibility for Implementing Key Process Areas

The project is primarily responsible for addressing many key process areas.

The organization is primarily responsible for addressing other key process areas.

There are both project and organizational responsibilities in all key process areas.



Key Process Areas to Achieve Level 5

Product and process quality

Continuous process improvement

Optimizing (5)

**Process change management
Technology change management
Defect prevention**



Key Process Areas to Achieve Level 4

**Product
and
process
quality**

**Integrated
engineering
process**

Managed (4)

**Software quality management
Quantitative process management**



Key Process Areas to Achieve Level 3

Integrated engineering process

Project management

Defined (3)

**Peer reviews
Intergroup coordination
Software product engineering
Integrated software management
Training program
Organization process definition
Organization process focus**



Key Process Areas to Achieve Level 2

**Project
management**

Ad hoc

Repeatable (2)

**Software configuration management
Software quality assurance
Software subcontract management
Software project tracking and oversight
Software project planning
Requirements management**



Key Process Areas

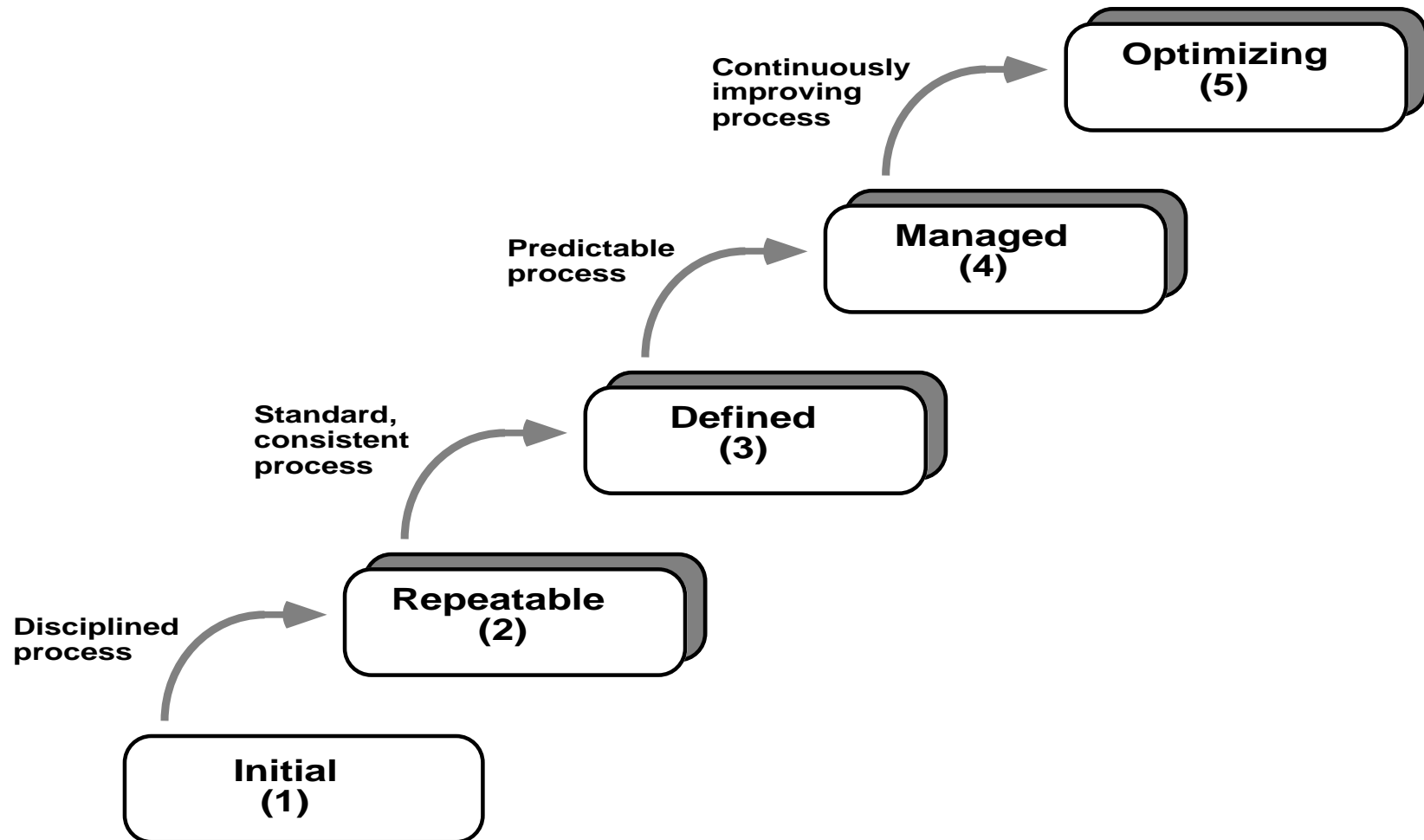
Identify a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability

Defined to reside at a single maturity level

Identify the issues that must be addressed to achieve a maturity level



The Five Maturity Levels



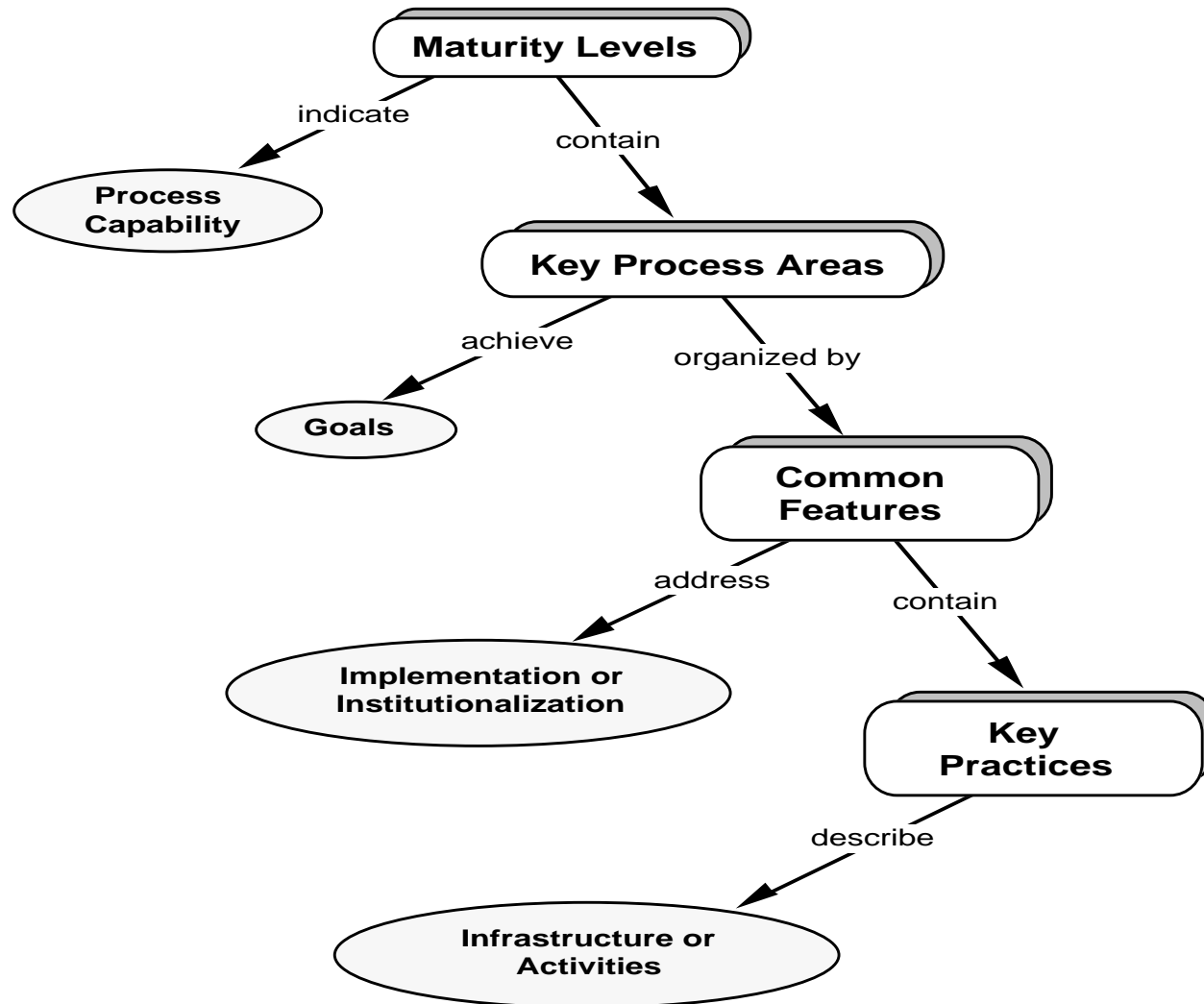


Maturity Levels

- MATURITY LEVEL – a well-defined evolutionary plateau on the path toward becoming a mature software organization**
- **each level is a layer in the foundation for continuous process improvement**
 - **there are five maturity levels in the CMM**



The CMM Structure





Components of the CMM

Maturity Levels —→ Process Capability

Key Process Areas → Goals

Common Features —→ Implementation or Institutionalization

Key Practices —→ Infrastructure or Activities



Critical Process Maturity Concepts

PROCESS CAPABILITY — the range of expected results that can be achieved by following a process, a predictor of future project outcomes

PROCESS PERFORMANCE — a measure of the actual results achieved from following a process

PROCESS MATURITY — the extent to which a specific process is explicitly defined, managed, measured, controlled, and effective

- implies a potential for growth in capability
- indicates both the richness of an organization's software process and the consistency with which it is applied



CMM Supporting Role

The CMM should support:

- **setting goals for senior management**
- **identifying priorities for process improvement**
- **identifying process capability of organizations**
- **predicting future process performance of projects**
- **industry-wide comparisons of the state of the practice**



CMM Definition

A description of the stages through which software organizations evolve as they define, implement, measure, control, and improve their software processes

A guide for selecting process improvement strategies by facilitating:

- **determination of current process capabilities**
- **identification of the issues most critical to software quality and process improvement**



What Is the CMM?

The application of process management and quality improvement concepts to software development and maintenance

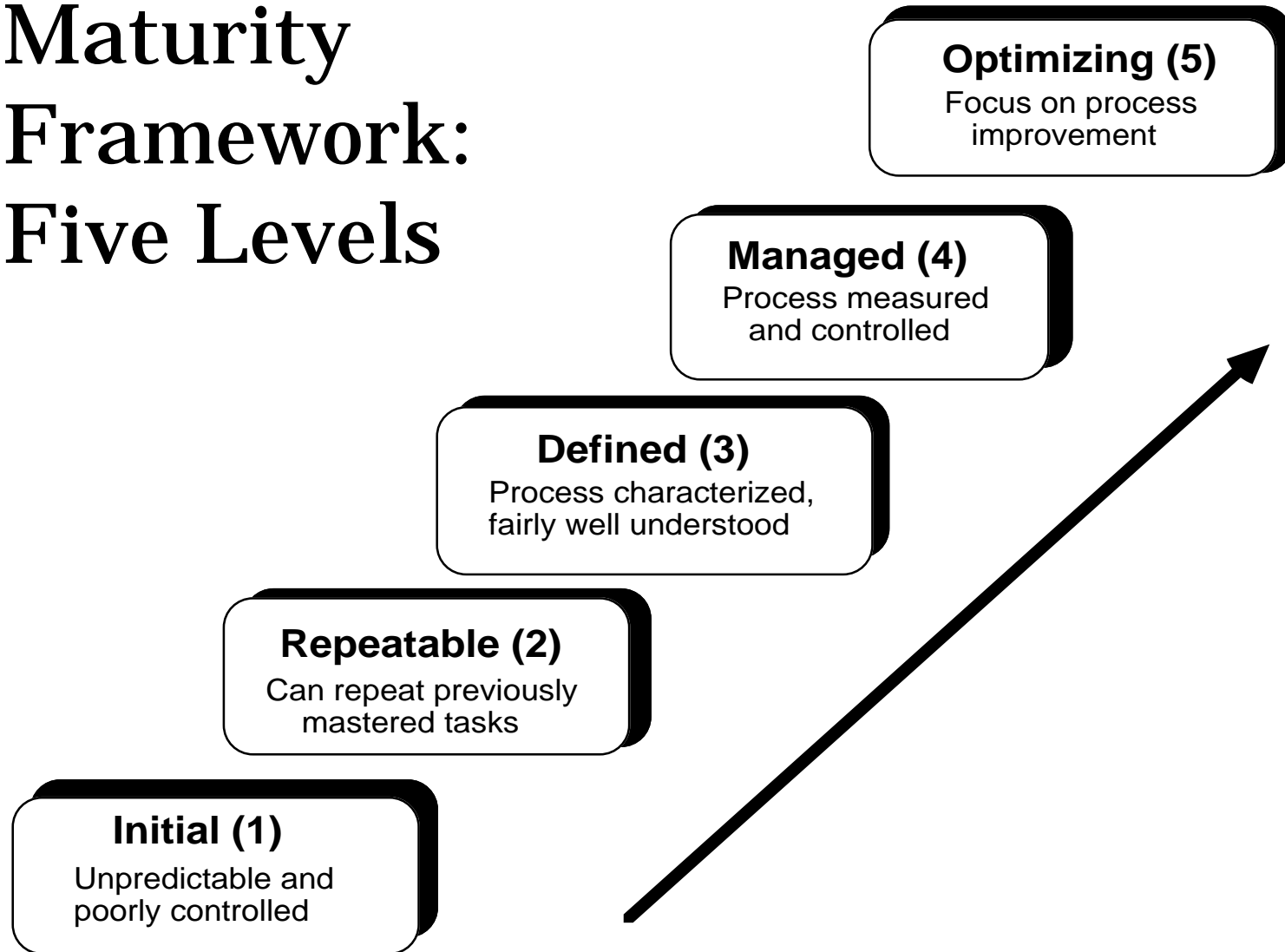
A guide for evolving toward a culture of engineering excellence

A model for organizational improvement

The underlying structure for reliable and consistent software process assessments and software capability evaluations



Maturity Framework: Five Levels





Maturity Model Inspirations

Process management concepts – Crosby, Deming, Juran, ...

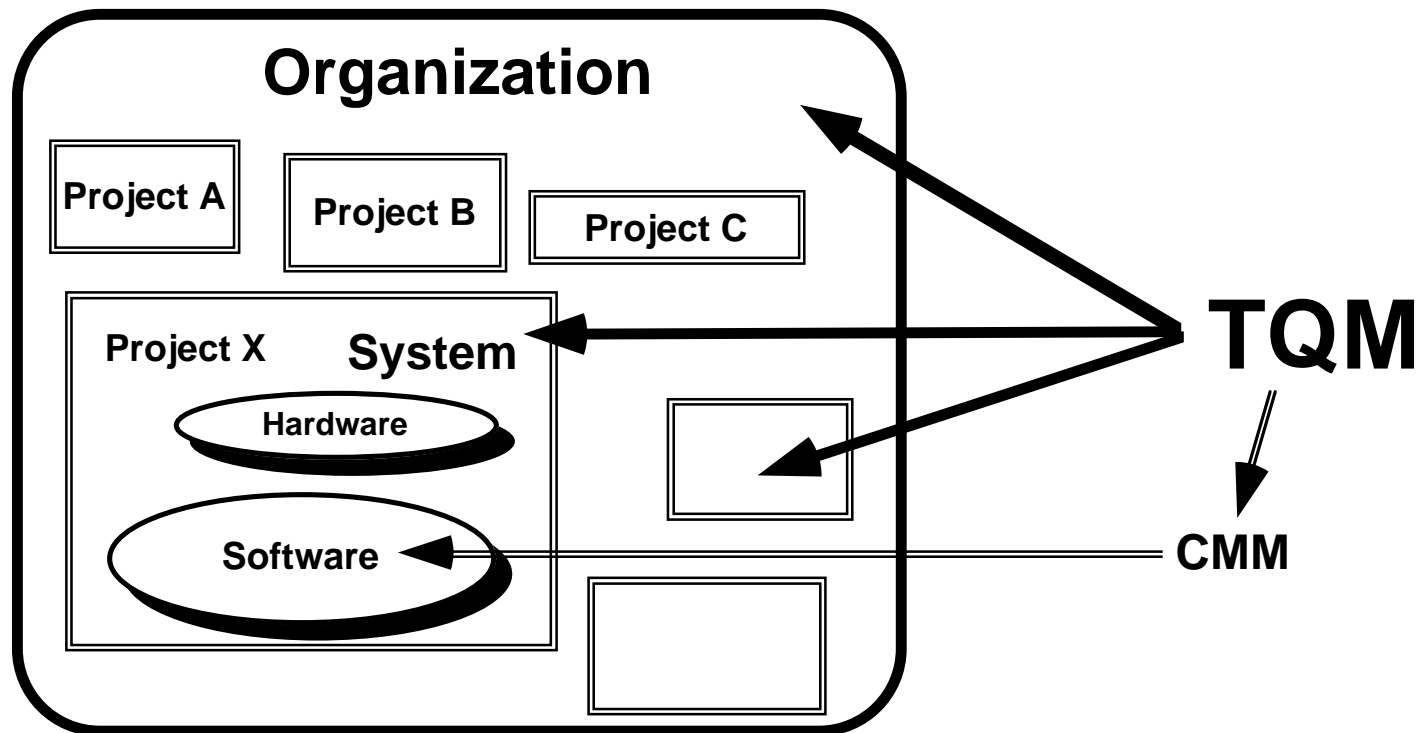
Experience

- **30 years of similar software problems**
- **commonly known software problems**
- **solutions exist**

Application of common-sense engineering



Applying TQM to Software



Process improvement fits in an overall business context – CMM applies to software.



Common Points in the Quality Movement

Enabling quality improvement is a management responsibility.

Quality improvement focuses on fixing the process, not the people.

Quality improvement must be measured.

Rewards and incentives are necessary to establish and maintain an improvement effort.

Quality improvement is a continuous process.



Total Quality Management

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

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

Department of Defense, *Total Quality Management Master Plan*, August 1988.



The Process Management Premise

The quality of a (software) system is largely governed by the quality of the process used to develop and maintain it.

This premise implies focus on process as well as product.

The value of this premise is visible world-wide in the Total Quality Management movements in the manufacturing and service industries.



Process Provides a Framework

A focus on people causes resistance to change – people naturally desire to do good work.

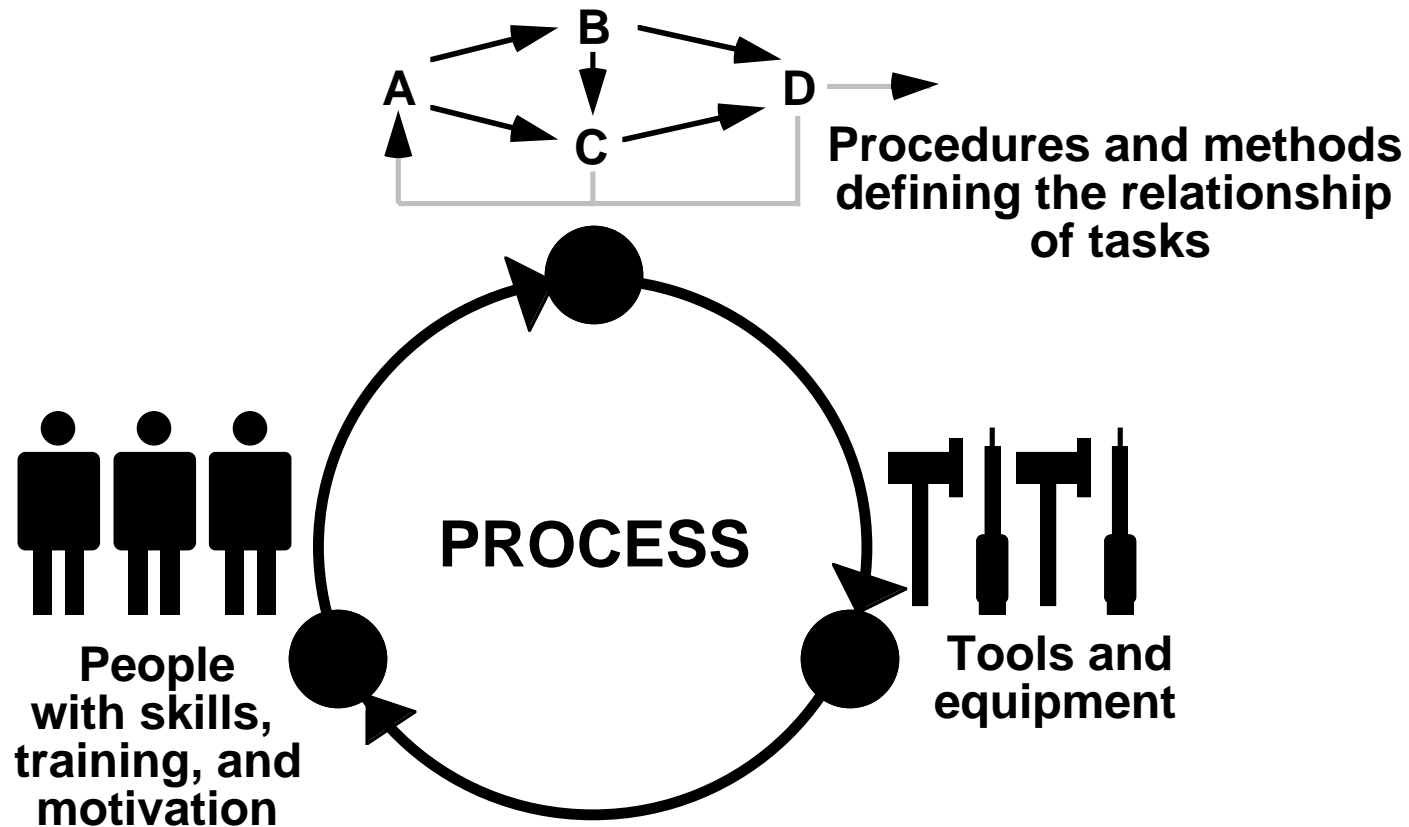
A focus on tools that do not fit into the process leads to ineffective automation – and shelfware.

A focus on procedures that do not match the process leads to unusable procedures – and shelfware.



A Definition of Process

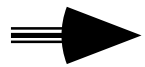
The means by which people, procedures, methods, equipment, and tools are integrated to produce a desired end result.





Topics

Introduction



The Capability Maturity Model

Understanding the Initial and Repeatable Levels

Understanding the Defined Level

Understanding the Managed and Optimizing Levels

Conclusion and Discussion



Contacts for General SEI Information

SEI Customer Relations (412) 268-5800

SEI FAX number (412) 268-5758

**Internet Address
 customer-relations@sei.cmu.edu**

**Mailing Address
 Customer Relations
 Software Engineering Institute
 Carnegie Mellon University
 Pittsburgh, PA 15213-3890**



SEI Mission and Vision

Our mission is to provide leadership in advancing the state of the practice of software engineering to improve the quality of systems that depend on software.

Our vision is to bring engineering discipline to the development and maintenance of software.



Carnegie Mellon University

Software Engineering Institute

The Software Engineering Institute

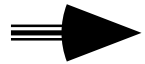
**Federally funded research and development
center (FFRDC)**

Affiliated with Carnegie Mellon University

Established in 1984



Topics



Introduction

The Capability Maturity Model

Understanding the Initial and Repeatable Levels

Understanding the Defined Level

Understanding the Managed and Optimizing Levels

Conclusion and Discussion



The Agenda - 2

Understanding the Defined Level	90 mins
--	----------------

Break

Understanding the Managed and Optimizing Levels	60 mins
--	----------------

Conclusion and Discussion	30 mins
----------------------------------	----------------



The Agenda - 1

Introduction

15 mins

The Capability Maturity Model

75 mins

Break

**Understanding the Initial and
Repeatable Levels**

90 mins

Lunch



Setting Expectations

This tutorial provides

- **an overview of the Capability Maturity Model (CMM) for managers and technical staff**
- **an awareness of the concepts of the CMM**

This tutorial is not a substitute for training or experience in applying the CMM.

The audience is expected to be knowledgeable about software engineering and management.



Carnegie Mellon University

Software Engineering Institute

The Capability Maturity Model: A Tutorial

April 1993

**Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213**

Sponsored by the U.S. Department of Defense