

# Capability Maturity Model (CMM)

Unit IV



# What is CMM?

- CMM: Capability Maturity Model
- Also called as SEI-CMM
- Developed by the Software Engineering Institute (SEI) of the Carnegie Mellon University
- Framework that describes the key elements of an effective software process.



# What is CMM?

- Describes an evolutionary improvement path for software organizations from an ad hoc, immature process to a mature, disciplined one.
- Provides guidance on how to gain control of processes for developing and maintaining software and how to evolve toward a culture of software engineering and management excellence.



# Process Maturity Concepts

## ■ Software Process

- set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products (e.g., project plans, design documents, code, test cases, user manuals)

## ■ Software Process Capability

- describes the range of expected results that can be achieved by following a software process
- means of predicting the most likely outcomes to be expected from the next software project the organization undertakes



# Process Maturity Concepts

- Software Process Performance
  - actual results achieved by following a software process
- Software Process Maturity
  - extent to which a specific process is explicitly defined, managed, measured, controlled and effective
  - implies potential growth in capability
  - indicates richness of process and consistency with which it is applied in projects throughout the organization

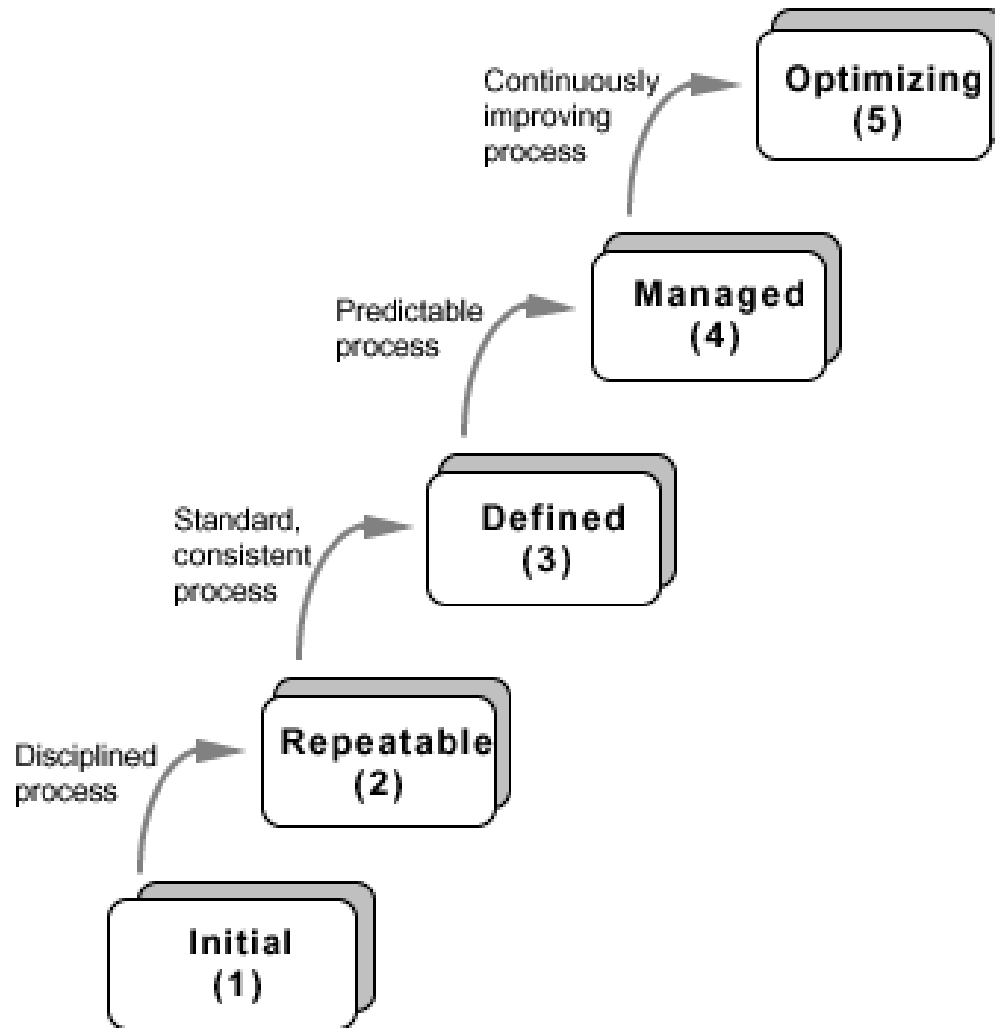


# What are the CMM Levels?

(The five levels of software process maturity)

Maturity level indicates level of process capability:

- Initial
- Repeatable
- Defined
- Managed
- Optimizing





# Level 1: Initial

- ⇒ Initial : The software process is characterized as **ad hoc**, and occasionally even chaotic.
- ⇒ Few processes are defined, and success depends on individual effort.
  - ⇒ At this level, frequently have difficulty making commitments that the staff can meet with an orderly process
  - ⇒ Products developed are often over budget and schedule
  - ⇒ Wide variations in cost, schedule, functionality and quality targets
  - ⇒ Capability is a characteristic of the individuals, not of the organization





## Level 2: Repeatable

- ✧ **Basic process management** processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
  - ✧ Realistic project commitments based on results observed on previous projects
  - ✧ Software project standards are defined and faithfully followed
  - ✧ Processes may differ between projects
  - ✧ Process is disciplined
  - ✧ earlier successes can be repeated



## Level 3: Defined

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



# Level 4: Managed

- ⌘ Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.
  - ⌘ Narrowing the variation in process performance to fall within acceptable quantitative bounds
  - ⌘ When known limits are exceeded, corrective action can be taken
  - ⌘ Quantifiable and predictable
    - ⌘ predict trends in process and product quality



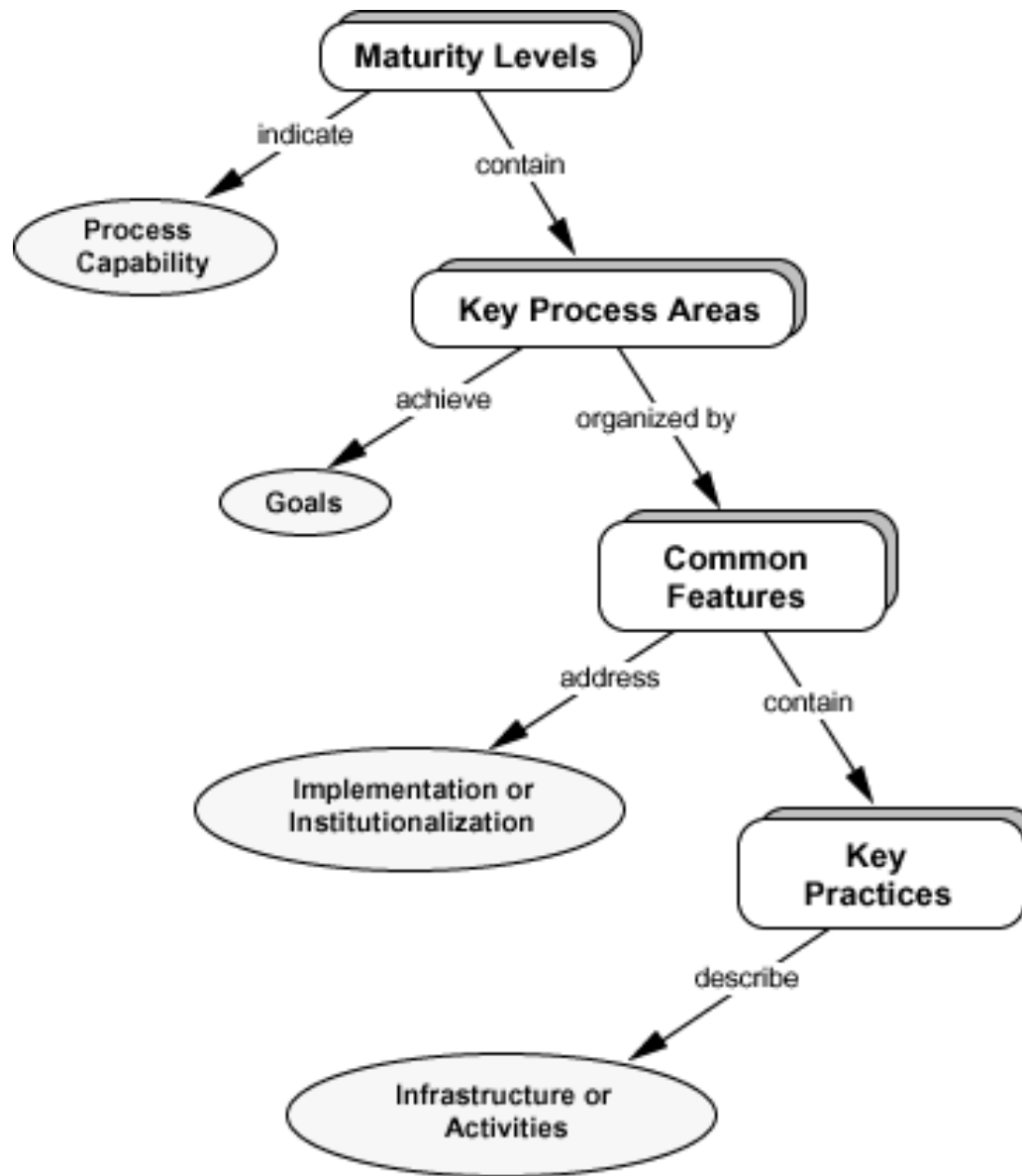
# Level 5: Optimizing

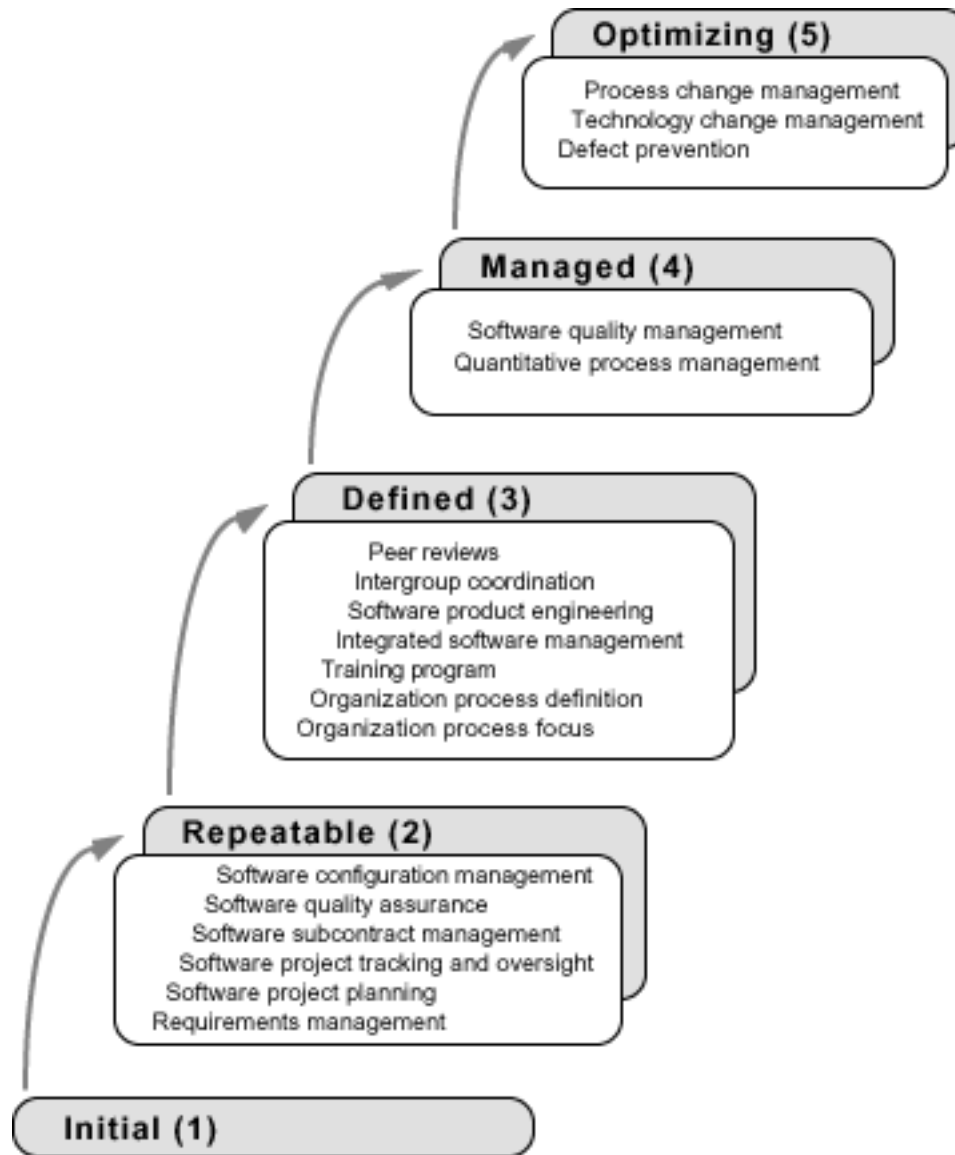
- ↻ Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.
- ↻ Goal is to prevent the occurrence of defects
  - ↻ Causal analysis
- ↻ Data on process effectiveness used for cost benefit analysis of new technologies and proposed process changes



# Internal Structure to Maturity Levels

- Except for level 1, each level is decomposed into key process areas (KPA)
- Each KPA identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing software capability.
  - commitment
  - ability
  - activity
  - measurement
  - verification





**The Key Process Areas by Maturity Level**

# Repeatable

L  
e  
v  
e  
l  
2  
K  
P  
As

Requirement Management

Software Project Planning

Software project tracking & Oversight

Software Subcontract management

Software Quality Assurance

Software Configuration Management





# Level 2 KPAs

- Requirements Management
  - Establish common understanding of customer requirements between the customer and the software project
  - Requirements is basis for planning and managing the software project
  - Not working backwards from a given release date!
- Software Project Planning
  - Establish reasonable plans for performing the software engineering activities and for managing the software project



# Level 2 KPAs

- Software Project Tracking and Oversight
  - Establish adequate visibility into actual progress
  - Take effective actions when project's performance deviates significantly from planned
- Software Subcontract Management
  - Manage projects outsourced to subcontractors
- Software Quality Assurance
  - Provide management with appropriate visibility into
    - process being used by the software projects
    - work products



# Level 2 KPAs

- Software Configuration Management
  - Establish and maintain the integrity of work products
  - Product baseline
  - Baseline authority

# Defined

L  
e  
v  
e  
l  
3  
K  
P  
As

Organization Process Focus

Organization Process Definition

Training Program

Integrated Software Management

Software Product Engineering

Intergroup Coordination

Peer Reviews



# Level 3 KPAs

- Organization Process Focus
  - Establish organizational responsibility for software process activities that improve the organization's overall software process capability
- Organization Process Definition
  - Develop and maintain a usable set of software process assets
    - stable foundation that can be institutionalized
    - basis for defining meaningful data for quantitative process management



# Level 3 KPAs

## ■ Training Program

- Develop skills and knowledge so that individual can perform their roles effectively and efficiently
- Organizational responsibility
- Needs identified by project

## ■ Integrated Software Management

- Integrated engineering and management activities
- Engineering and management processes are tailored from the organizational standard processes
- Tailoring based on business environment and project needs



# Level 3 KPAs

- Software Product Engineering
  - technical activities of the project are well defined (SDLC)
  - correct, consistent work products
- Intergroup Coordination
  - Software engineering groups participate actively with other groups
- Peer Reviews
  - early defect detection and removal
  - better understanding of the products
  - implemented with inspections, walkthroughs, etc



# Managed

L  
e  
v  
e  
l  
4  
K  
P  
As

Quantitative Process Management

Software Quality Management





# Level 4 KPAs

- Quantitative Process Management
  - control process performance quantitatively
  - actual results from following a software process
  - focus on identifying and correcting special causes of variation with respect to a baseline process
- Software Quality Management
  - quantitative understanding of software quality
    - products
    - process



# Optimizing

L  
e  
v  
e  
l  
5  
K  
P  
As

Process Change Management

Technology Change Management

Defect Prevention



# Level 5 KPAs

- Process Change Management
  - continuous process improvement to improve quality, increase productivity, decrease cycle time
- Technology Change Management
  - identify and transfer beneficial new technologies
    - tools
    - methods
    - processes
- Defect Prevention
  - causal analysis of defects to prevent recurrence



# What are the benefits ?

- Helps forge a shared vision of what software process improvement means for the organization
- Defines set of priorities for addressing software problems
- Supports measurement of process by providing framework for performing reliable and consistent appraisals
- Provides framework for consistency of processes and product



# Why measure software and software process?

Obtain data that helps us to better control

- schedule
- cost
- quality of software products



# Consistent measurement provide data for:

- Quantitatively expressing requirements, goals, and acceptance criteria
- Monitoring progress and anticipating problems
- Quantifying tradeoffs used in allocating resources
- Predicting schedule, cost and quality