

# Maturity Models

Course Code: CSC4133

Course Title: Software Quality and Testing



**Dept. of Computer Science**  
**Faculty of Science and Technology**

<b>Lecturer No:</b>	<b>4</b>	<b>Week No:</b>	<b>2</b>	<b>Semester:</b>	
<b>Lecturer:</b>	<i>Prof. Dr. kamruddin Nur, <a href="mailto:kamruddin@aiub.edu">kamruddin@aiub.edu</a></i>				

# Lecture Outline



- Basic Idea in Software Process
- Capability Maturity Model (CMM) & Capability Maturity Model Integration (CMMI)
- Test Process Improvement (TPI)
- Testing Maturity Model (TMM)

# Objectives and Outcomes



- **Objectives:** To understand the basic idea in software process, to understand different maturity models.
- **Outcomes:** Students are expected to be able to explain the basic idea in software process; be able to explain CMM and CMMI and their different levels, be able to explain TPI and TMM models.

# Basic Idea in Software Process



- A process comprises a set of **activities** that are executed to develop products.
  - The activities find expressions in the form of methods, techniques, strategies, procedures, and practices.
  - The activities heavily rely on information repositories, such as documents, standards, and policies.
- Different processes are driven by different goals and availability of resources.

# Basic Idea in Software Process



- It is useful to follow a defined process because of the following benefits:
  - The process can be **repeated** in subsequent projects
  - The process can be **evaluated** by using a variety of **metrics**, such as **cost**, **quality**, and **time** to deliver
  - Actions can be taken to **improve the process** to achieve better results

# Basic Idea in Software Process



- A software process comprises the following tasks –
  - Gathering requirements
  - Constructing a functional specification
  - Designing the system
  - Writing code
  - Testing the system
  - Delivering the system
  - Maintaining the system

# Testing in Software Process



- Software testing is treated as a distinct process because it involves a variety of unique activities, techniques, strategies, and policies.
  - Testing is performed to reveal defects and show to what extent the software possesses different quality attributes, such as reliability and performance.
  - Testing begins almost at the same time a project is conceptualized.
  - Testing is carried out by **different people** at different stages of system development.
  - A number of **different techniques** can be applied at each level of testing.
  - A number of **different strategies** can be applied at each level of testing.
  - A number of **metrics** can be monitored to gauge the progress of testing.
  - Testing is influenced by **organizational policies**.
  - Testing can be performed as a combination of **manual** and **automated** modes of execution of test cases.

# Maturity Models



- To be able to improve a defined process, organizations need to evaluate its capabilities and limitations.
  - Example: The **Capability Maturity Model (CMM)** allows an organization to evaluate its software development processes.
    - The CMM model supports incremental process improvement.
- A separate model, known as the **Testing Maturity Model (TMM)**, has been developed to evaluate a testing process.
- For an organization to be able to improve their testing process, the **Test Process Improvement (TPI)** model has been developed.



# Capability Maturity Model (CMM)



- In 1986, funded by US Department of Defense, the Software Engineering Institute (SEI) of Carnegie Mellon University developed the CMM
- In the CMM model, the maturity level of an organization tells us to what extent an organization can produce low cost, high quality software.
- The model's aim is to improve existing software development processes, but it can also be applied to other processes.
- Having known the current maturity level, an organization can work to reach the next higher level.

# Capability Maturity Model (CMM)



- The CMM is a framework for evaluating and improving the present capability of a software development company.
- The CMM is used to assess & improve the maturity level of an organization to develop software products.
- There are five maturity levels in the CMM.
  - Each level is characterized by a set of key process areas(KPA), except Level 1.

# CMM Levels



- There are five(5) levels in CMM:

Level 1: Initial

Level 2: Repeatable

Level 3: Defined

Level 4: Managed

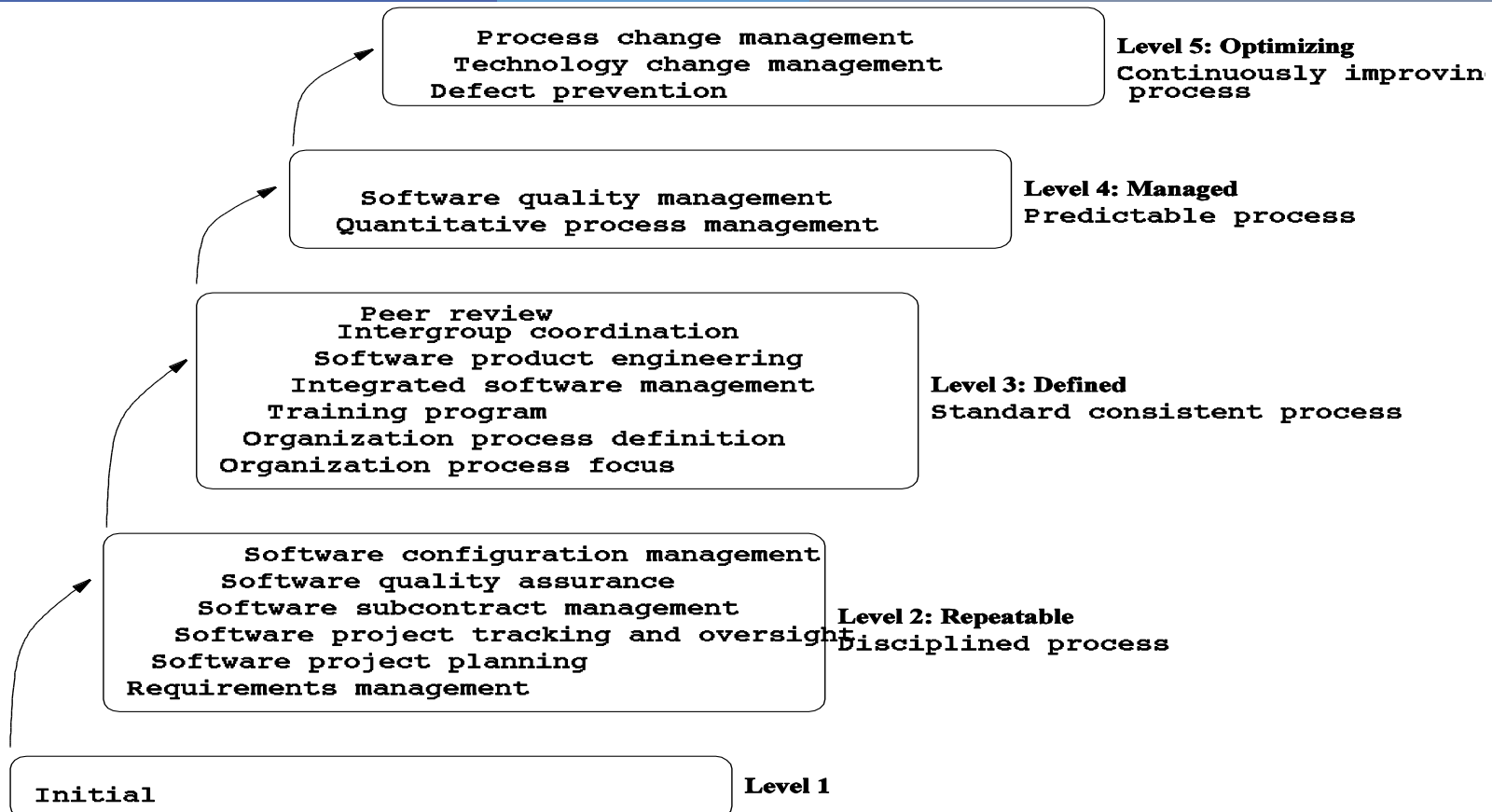
Level 5: Optimizing



# Five Maturity Levels of CMM

- At the **Initial** level, **processes are disorganized, even chaotic**. Success is likely to depend on individual efforts, and is **not** considered to be **repeatable**, because processes would not be **sufficiently defined** and **documented** to allow them to be **replicated**.
- At the **Repeatable** level, **basic project management techniques are established**, and **successes could be repeated**, because the requisite processes would have been made **established, defined, and documented**.
- At the **Defined** level, **an organization has developed its own standard software process** through greater attention to **documentation, standardization, and integration**.
- At the **Managed** level, **an organization monitors and controls its own processes** through **data collection and analysis**.
- At the **Optimizing** level, **processes are constantly being improved** through **monitoring feedback** from current processes and introducing innovative processes to better **serve** the organization's particular needs.

# Five Levels of CMM



# Capability Maturity Model Integration (CMMI)



- CMMI is the **successor** of the capability maturity model (CMM)
- The first version of the **CMMI** was released in **2002**
- CMMI was developed by a group of experts from industry, government (US DoD), and the Software Engineering Institute (SEI) at **CMU**.
- CMMI is a model for **judging** the maturity of software processes of an **organization** & for **identifying** the **key practices** that are required to **increase** the maturity of these processes.

# Levels of CMMI



- Five maturity levels of CMMI:

Level 1 : Initial

Level 2 : Managed

Level 3 : Defined

Level 4 : Quantitatively Managed

Level 5 : Optimizing

# Main Focuses of different levels of CMMI



- 1) **Initial** ==> Ad-hoc/chaotic
- 2) **Managed** ==> Basic Project Management
- 3) **Defined** ==> Process Standardization
- 4) **Quantitatively Managed** ==> Quantitative management
- 5) **Optimized** ==> Continuous process improvement



# Test Process Improvement (TPI)



- A test process is a certain way of performing activities related to defect detection. A few such activities are as follows:
  - Identifying test goals
  - Preparing a test plan
  - Identifying different kinds of tests
  - Hiring test personnel
  - Designing test cases
  - Procuring test tools
  - Assigning test cases to test engineers
  - Prioritizing test cases for execution
  - Organizing the execution of test cases into multiple test cycles
  - Executing test cases
  - Reporting defects



# Test Process Improvement (TPI)

- It is important to improve test processes by following a defined model.
- The idea of improving test processes by following a model, namely the **Test Process Improvement (TPI)** model, was first studied by **Tim Koomen** and **Martin Pol**.

# Why does a test process need to be improved?



- A test process needs to be improved for three reasons:
  - 1) Quality: A better test process should give more insights into the quality characteristics of a system being tested.
  - 2) Lead Time: A better test process saves testing time, and thereby gives more time to other areas of system development.
  - 3) Cost: A better test process is expected to be carried out with a lower cost.



# Test Process Improvement (TPI)

- **How to improve a test process?**
- An intuitive approach to improving a test process is as follows:
  - Step 1: Determine an area for improvement.
  - Step 2: Evaluate the current state of the test process.
  - Step 3: Identify the next desired state and the means to achieve it.
  - Step 4: Implement the necessary changes to the process.

# Test Process Improvement (TPI)



- The TPI model supports gradual process improvement.
  - The **current status** of a test process is evaluated from different viewpoints, known as **key areas** – and **20 key areas** have been identified
  - The **status** of a test process w.r.t. a **key area** is represented in terms of **one** of **four levels** of **maturity** – **A, B, C, and D**



# Test Process Improvement (TPI)

## ■ **Maturity levels of test processes:**

- Based on the idea of dependencies and prioritization, a Test Maturity Matrix is constructed.
  - **Test Maturity Matrix** shows that the overall maturity of a test process can be represented on a *scale of 1—13*.
  - The 13 scales of maturity of a test process are classified into **three distinct segments** as follows:
    1. **Controlled (1—5):** All component activities are planned and executed in phases according to the plan.
    2. **Efficient (6—10):** All the key areas, except *Evaluation*, are raised to at least B level with some being at C.
    3. **Optimizing (11—13):** All the key areas have reached their respective highest maturity levels.
- ==> Optimizing a test process means performing testing tasks in the best possible manner from the standpoint of quality, time, and cost.**

# Applying the TPI model



- Analyze the current test process, in terms of the 20 key areas, and give each key area a rating – A, B, C, or D.
- Evaluate the current scale, between 1—13, of the test process by comparing the current status of the test process with the standard Test Maturity Matrix
- Identify the goal of the organization in terms of the next scale to be achieved. Identify the key areas where improvements must be achieved.
- Take actions to improve the key areas identified in the preceding step.



# Testing Maturity Model (TMM)

- Similar to the concept of evaluating and improving software development processes, there is a need for a framework to assess and improve testing processes.
- Continuous improvement of testing processes is an ideal goal of organizations.
- Evaluation plays a key role in process improvement.
- TMM was pioneered by Ilene Burnstein to help organizations evaluate and improve their testing processes.



# Testing Maturity Model (TMM)



- The TMM framework describes an evolutionary path of test process maturity in **five levels**, or **stages**.
- Each level is characterized by the concepts of
  - **Maturity goals**
  - **Supporting maturity goals**, and
  - **Activities, Tasks, and Responsibilities (ATRs)**



# Testing Maturity Model (TMM)

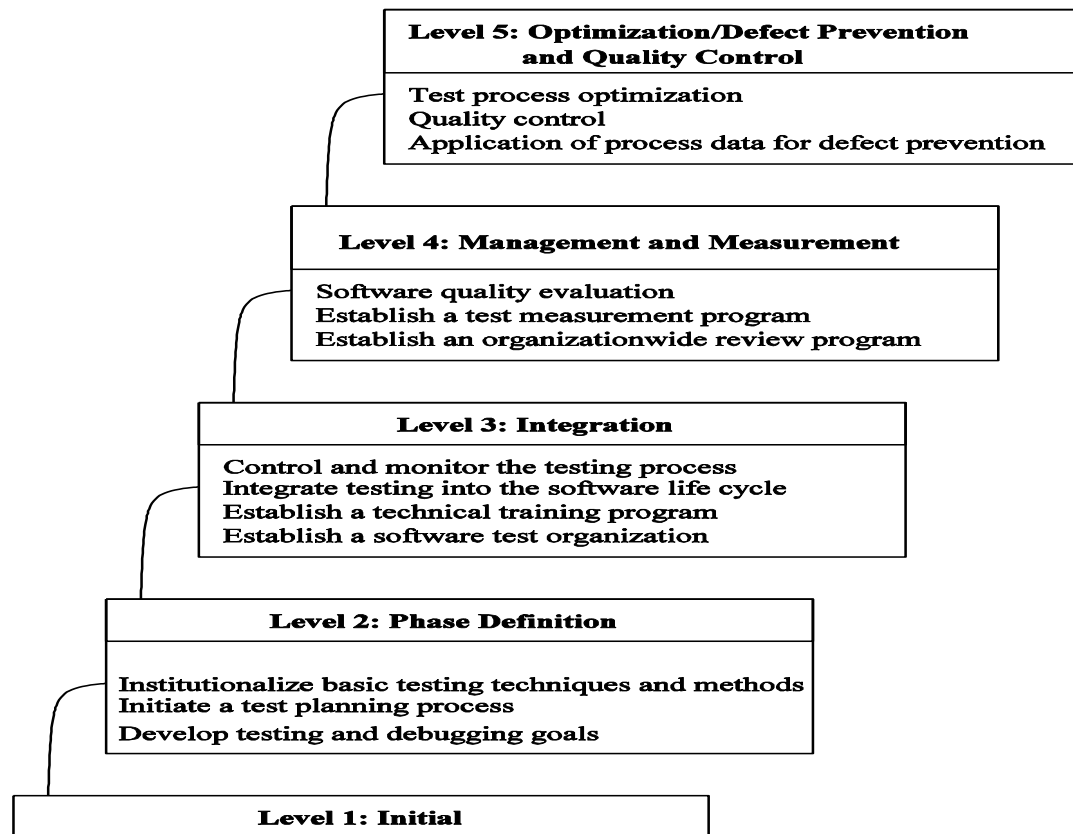
- **Maturity goals**
  - Each maturity level, except 1, contains certain maturity goals
  - For an organization to reach a certain level, the corresponding maturity goals must be met by the organization
- **Maturity sub-goals**
  - Maturity goals are supported by maturity sub-goals
- **ATRs**
  - Maturity sub-goals are achieved by means of ATRs
  - ATRs are further refined into “views” from the perspectives of three groups: Managers, Developers and test engineers, Customers

# TMM Levels



- 1) Initial
- 2) Phase definition
- 3) Integration
- 4) Management and Measurement
- 5) Optimization/Defect Prevention and Quality Control

# The 5-level structure of the TMM



# TMM Levels



- **Level 1 – Initial**

- A **chaotic process**
- There are **no maturity goals to be met at this level.**
- **Not** distinguished from debugging and ill defined
- **Testing begins after code is written.**
- **An organization performs testing to demonstrate that the system works.**
- No **serious effort** is made to track the progress of testing.
- **Test cases are designed and executed in an ad hoc manner.**
- In summary, testing is **not** viewed as a **critical, distinct** phase in software development.

# TMM Levels



- **Level 2 – Phase Definition:** The maturity goals are as follows:
  - **Testing** becomes a **defined phase** following coding
  - Develop testing and debugging goals
  - Identify testing as a separate function from debugging
  - **Initiate a test planning process:**
    - Identify test objectives
    - Analyze risks
    - Devise strategies
    - Develop test specifications
    - Allocate resources
  - Institutionalize basic testing techniques and methods

# TMM Levels



- **Level 3 – Integration:** The maturity goals are as follows:
  - Integrate testing into the software lifecycle.
  - Establish a software test group.
  - Establish a technical training program.
  - Control and monitor the testing process.

# TMM Levels



- **Level 4 – Management and Measurement:** The maturity goals are:
  - Establish a test management & measurement program
  - Establish an organization-wide review program
  - Evaluate software quality



# TMM Levels



- **Level 5 –Optimization/Defect Prevention and Quality Control:** The maturity goals are as follows:
  - Test process optimization
  - Application of process data for defect prevention
  - Statistical quality control
  - Testing process is well defined and managed
  - Testing costs and effectiveness are monitored
  - Automated tools are a primary part of the testing process



# Books

- Software Testing And Quality Assurance – Theory and Practice by Kshirasagar Naik & Priyadarshi Tripathy



## References

1. *Software Quality Engineering: Testing, Quality Assurance and Quantifiable Improvement*, by Jeff Tian, published by Wiley, ISBN 0-471-71345-7, is the required text.
2. *Software Quality Assurance: From Theory to Implementation*, by Daniel Galin
3. *Software Testing and Continuous Quality Improvement*, by William E. Lewis
4. *The Art of Software Testing*, by Glenford J. Myers, Corey Sandler and Tom Badgett
5. *Software Testing Fundamentals: Methods and Metrics* by Marnie L. Hutcheson