8. QUALITY MANAGEMENT

Contents:

8.1 Quality concepts

8.2 Software Quality Assurance(SQA)

8.3 Software reliability

8.4 Software maintenance

8.5 ISO standards

8.6 CMMI

8.7 SQA plan

8.8 Software certification

8.1 Quality concepts

Quality Software

- Quality software refers to a software which
  - is reasonably bug or **defect free,**
  - is **delivered in time** and within the specified budget,
  - **meets the requirements** and/or expectations,
  - and is **maintainable**.
- Quality is conformance to requirements.
- Quality is fitness for use.

McCall's Software Quality Factors

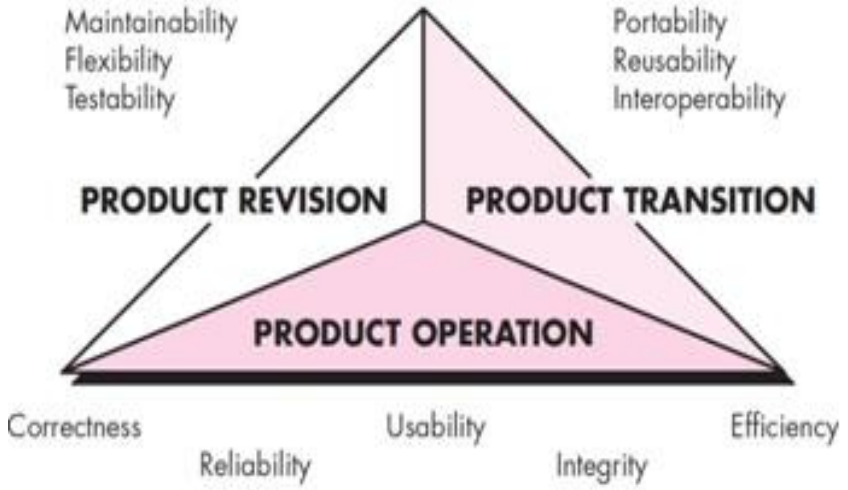


Fig. McCall's software quality factors

1. **Correctness** –The extent to which a software **meets its requirements specification.**
2. **Reliability** –The extent to which a software performs its intended functions **without failure.**
3. **Usability** –The extent of **effort required to learn**, operate and understand the functions of the software.
4. **Integrity** –The extent to which the software **can control an unauthorized person** from the accessing the data or software.
5. **Efficiency** –The **amount of hardware resources and code**, the software needs to perform a function.
6. **Maintainability** –The **effort required to detect and correct an error** during maintenance phase.
7. **Flexibility** –The **effort needed to improve or modify** an operational software program.
8. **Testability** –The **effort required to verify a software** to ensure that it meets the specified requirements.
9. **Portability** –The effort required to **transfer a program from one platform to another**.
10. **Re-usability** –The extent to which the program's **code can be reused in** other applications.
11. **Interoperability** –The effort required to **integrate two systems** with one another.

Software Quality Attributes

Safety	Understandability	Portability
Security	Testability	Usability
Reliability	Adaptability	Reusability
Resilience	Modularity	Efficiency
Robustness	Complexity	Learnability

Quality Management

- **Quality Management** – ensures that required level of product quality is achieved. Process involved are:
    1. **Defining procedures** and **standards**.
    2. **Applying procedures and standards** to the product and process.
    3. **Checking that procedures** are followed.
    4. **Collecting and analyzing** various quality data.

- The **quality documentation** is a record of progress and supports continuity of development as the development team changes.
- Quality management is particularly important for large, complex systems.
- For smaller systems, quality management needs less documentation and should focus on establishing a quality culture.

Quality management and software development

- Quality management provides an independent check on the software development process.
- The **QM team checks the project deliverables** to ensure that they are consistent with organizational standards and goals.
- They **also check process documentation**, which records the tasks that have been completed by each team working on this project.

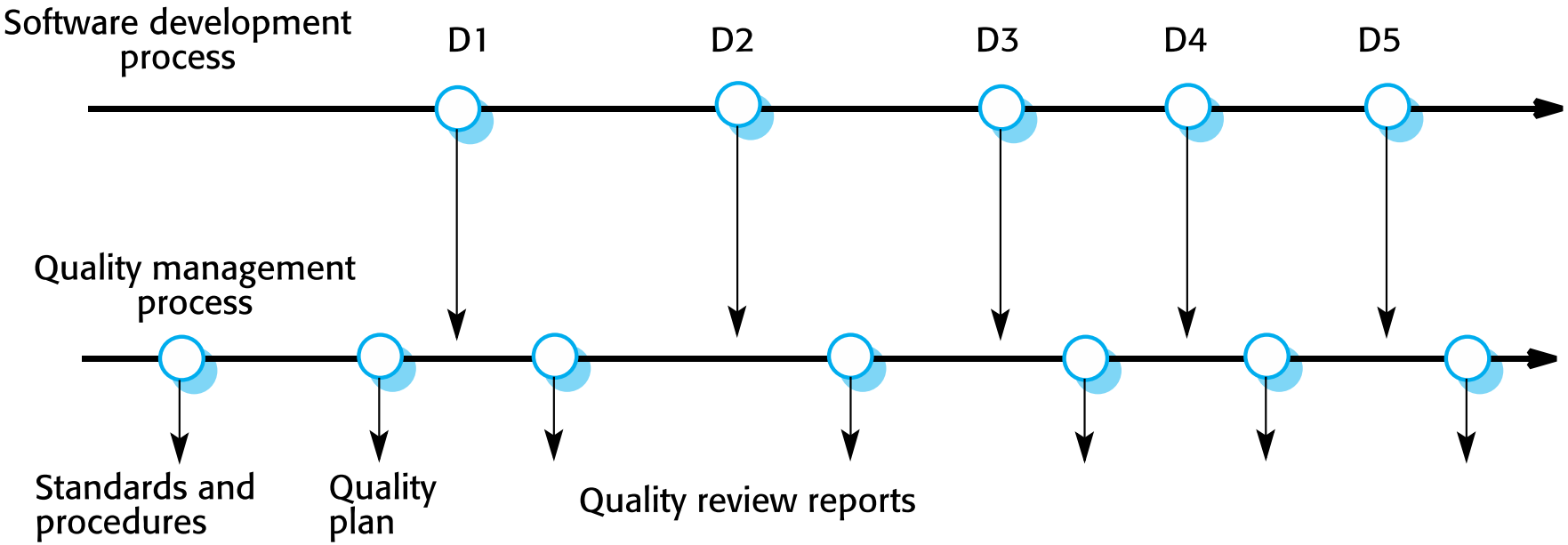


Figure: Quality management and software development

Process based quality

- The development process used has a significant influence on the quality of the software,
- and good processes are more likely to lead to good quality software.
- Process quality management and improvement can result in fewer defects in the software being developed.
- Defined processes are important, but quality managers should also aim to develop a "**quality culture**" in which everyone responsible for software development is committed to achieving a high level of product quality.

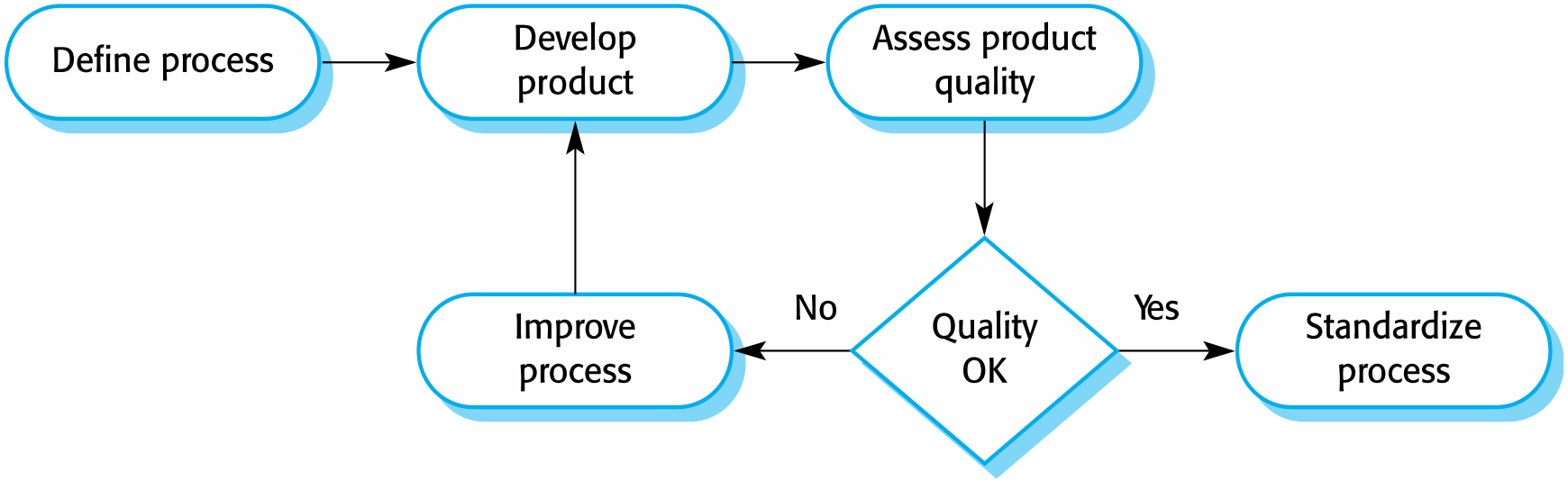
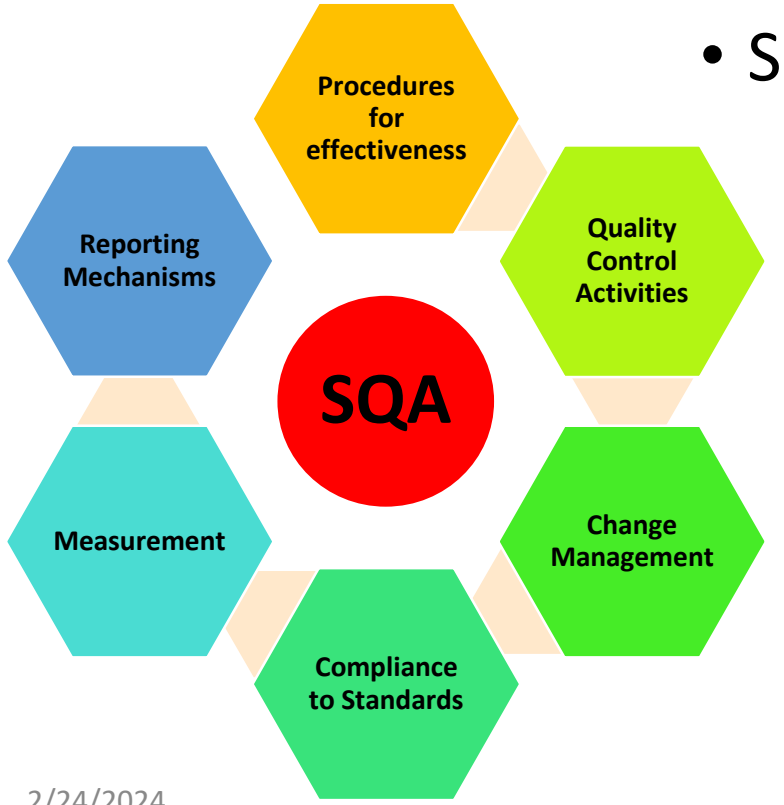


Figure: Process based quality

8.2 Software Quality Assurance(SQA)

Software Quality Assurance (SQA)

- Simply, a **way to assure quality** in the software
- Software Quality Assurance (SQA) is defined as a planned and systematic approach to the evaluation of the quality of software product standards, processes, and procedures.



- SQA encompasses :
  1. **procedures** for the effective application of methods and tools,
  2. oversight of **quality control activities** such as technical reviews and software testing,
  3. procedures for **change** management,
  4. procedures for **assuring compliance to standards,** and
  5. **measurement and reporting** mechanisms.

Major activities in SQA

1. **Creating an SQA Management plan**
   - Laying down proper plan for carrying out SQA
   - What SQA approach going to be followed
   - What engineering activities to be carried
   - team is okay?

2. **Setting the checkpoints.**
   - Set up different check points for evaluating quality of project activities.

3. **Apply software engineering techniques**
   - Techniques such as interviews, Functional Analysis System Technique (FAST)
   - Project estimation using WBS, SLOC, FP

4. **Executing FTR**
   - Formal Technical Review (FTR) done to evaluate the quality and design of the prototype.
   - Meeting conducted with technical staff
   - Helps in detecting errors in the early phase of SDLC.

5. **Having multi-testing strategies**
   - Tested from different angles to ensure better quality.

6. **Enforcing Process Adherence**
   - means complying with standard process.
   - Process should be stick to defined standard procedures.

Benefits and Disadvantages of SQA

Advantages:

- SQA **produce high quality software.**
- High quality application **saves time and cost.**
- SQA is beneficial for **better reliability.**
- High quality commercial software **increase market share** of company.
- **Improving the process** of creating software.

Disadvantages:

- adding **more resources,**
- employing **more workers** to help maintain quality

8.3 Software reliability

Software Reliability

- "the **probability of failure-free operation** of a computer program in a specified environment for a specified time".
- describes the acceptance limit of its overall quality
- Software Reliability means Operational reliability
  - ability of a system or component to perform its required functions under static conditions for a specific period
- Simply, software reliability is the probability of a failure-free operation of a program for a specified time in a specified environment

Software Reliability Metrics

1. **Rate of Occurrence of Failure (ROCOF)**

   It measures the frequency of occurrence of unexpected behavior (i.e. failures).

2. **Mean Time to Failure (MTTF)**

   Average time between two successive failures observed over a large no. of failures.

3. **Mean-Time to Repair (MTTR)**

   Measures the average time it takes to track the errors and to fix them.

4. **Mean-Time Between Failure (MTBF)**

   MTBF=MTTF+MTTR

   E.g. MTBF of 300 hrs indicates that the next failure is expected after 300 hrs.

5. **Probability of Failure on Demand (POFOD)**

   It measures the likelihood of the system failing when a service request is made.

6. **Availability**

   It is a measure of how likely shall the system be available for use over a given period of time.

$$\text{Availability} = \frac{\text{MTTF}}{\text{MTBF}} \text{ x } 100\%$$

8.4 Software Maintenance

Software Maintenance

- Software maintenance is the **process of making changes** to a software system **after it has been deployed.**
- It involves various activities aimed at keeping the software in good working order, **improving** its performance, **fixing bugs**, **adapting** it to new environments, and **adding new features**.
- Software maintenance is a crucial phase in the software development life cycle and typically involves the following activities:
    1. **Corrective** Maintenance
    2. **Adaptive** Maintenance
    3. **Perfective** Maintenance
    4. **Preventive** Maintenance

Why Software Maintenance?

- Software Maintenance is needed for:-
  1. Correct errors
  2. Change in user requirement with time
  3. Changing hardware/software requirements
  4. To improve system efficiency
  5. To optimize the code to run faster
  6. To modify the components
  7. To reduce any unwanted side effects.

The maintenance is required to ensure that the system continues to satisfy user requirements.

8.5 ISO standards

ISO Standards

- International Standard Organization (ISO) is a consortium of 63 countries to formulate and foster standardization.
- The ISO 9000 is a series of 3 standards ISO 9001, 9002, and 9003.
  - **ISO 9001:** Applies to organizations engaged in design, development, production, and servicing of goods, so applicable to most software development organization.
  - **ISO 9002:** Applies to organizations which do not design products but are only involved in production. E.g. steel and car manufacturing industries (they buy plant and product design from others).
  - **ISO 9003:** Applies to organizations involved only in installation and testing of products.

ISO 9001 Standard

- ISO 9001 is a **standard that sets out the criteria for a quality management** system and is applicable to any organization, regardless of its size or the industry it operates in.
- The ISO 9001 standard **focuses on ensuring that the organization has quality management procedures** in place and that it follows these procedures.

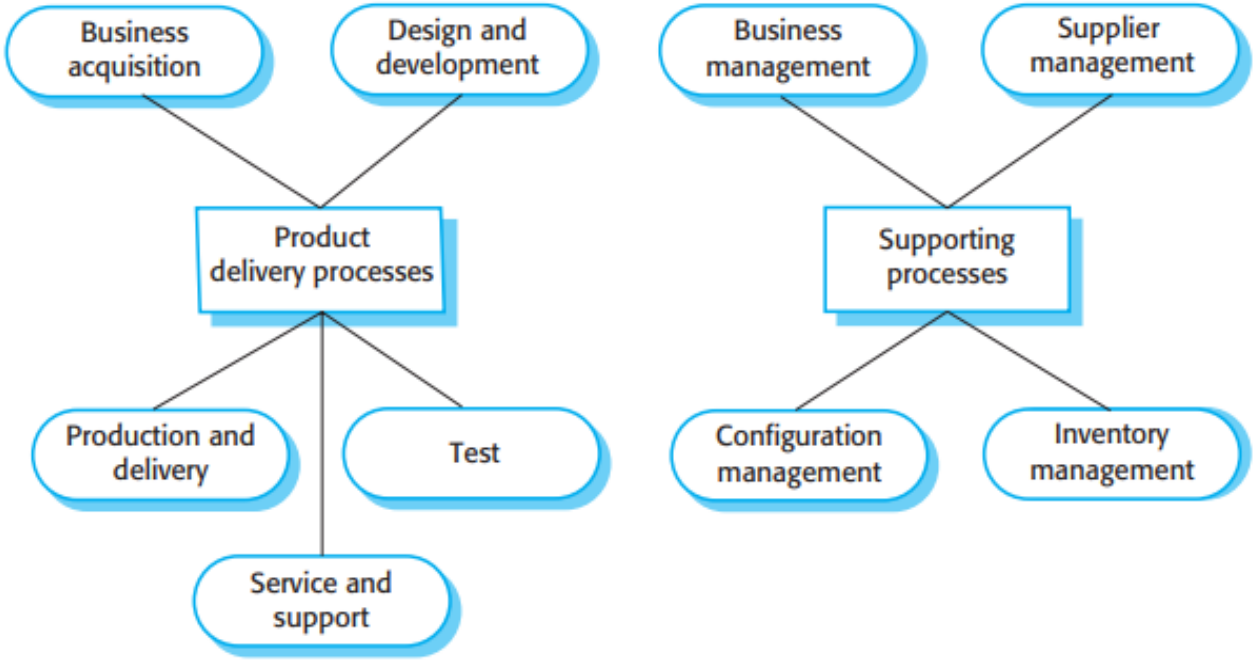


Figure: ISO 9001 core processes

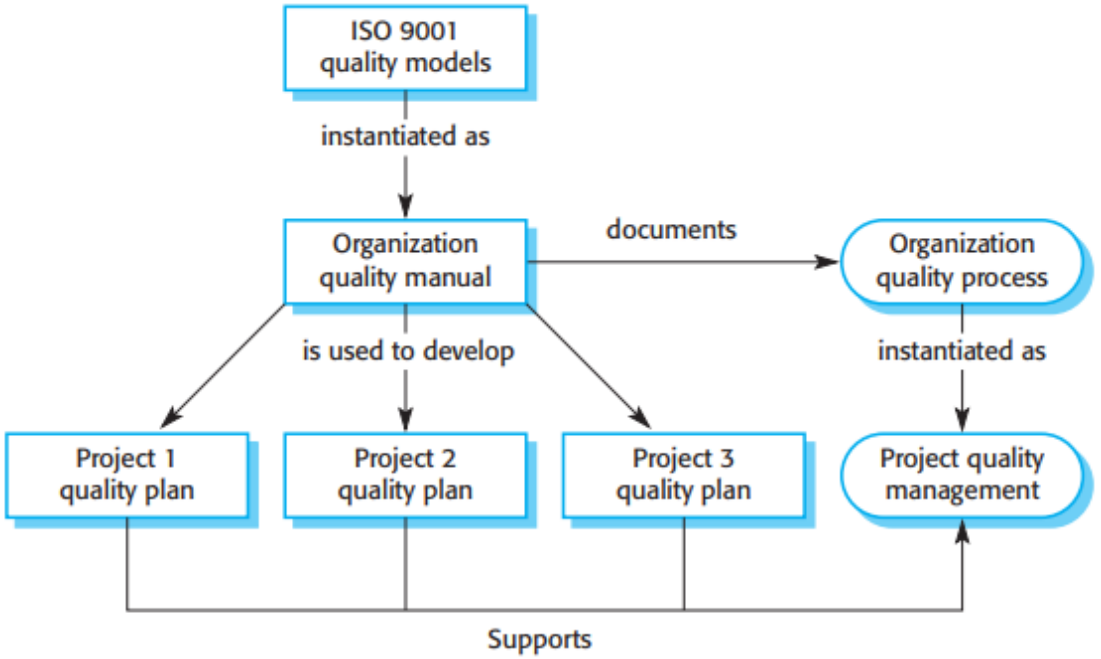


Figure: ISO 9001 and Quality Management

8.6 CMMI

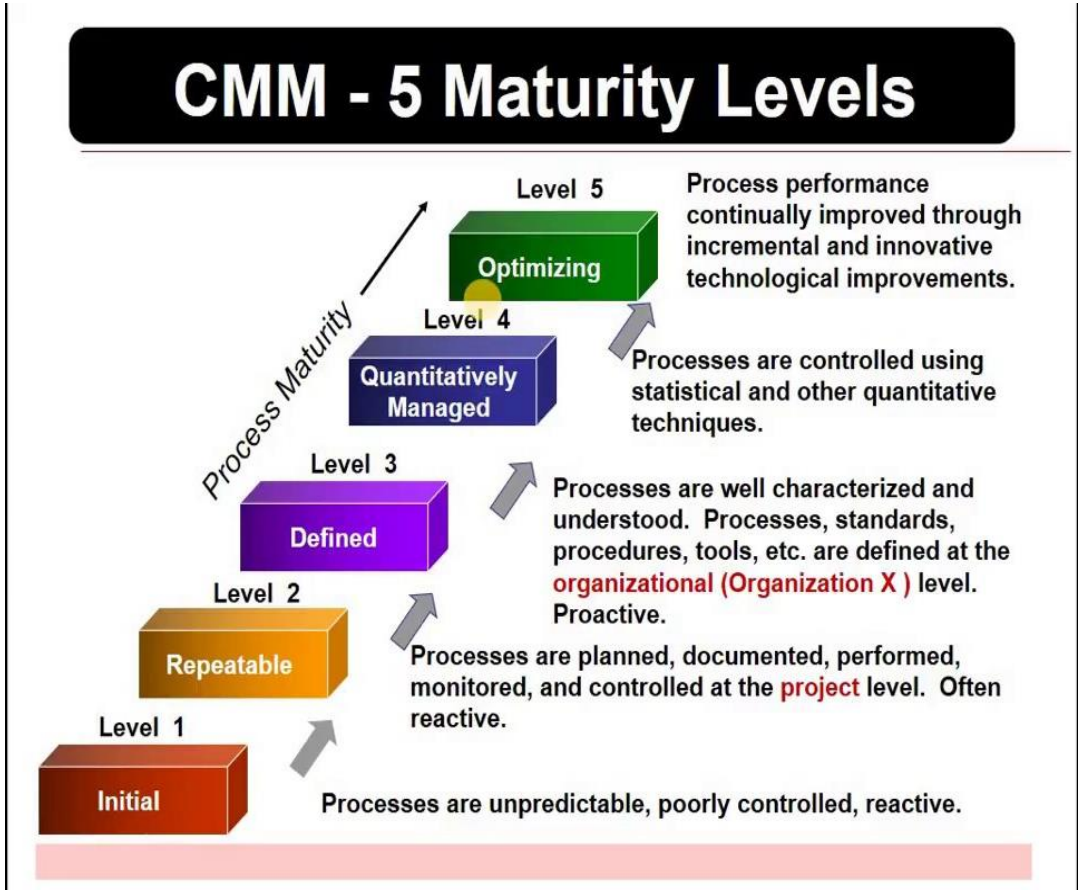
The CMMI Model

- CMMI (Capability Maturity Model Integration) **is a proven industry framework to improve product quality and development efficiency for both hardware and software.**
  - Sponsored by US Department of Defence in cooperation with Carnegie Mellon University and the Software Engineering Institute (SEI)
  - Many companies have been involved in CMMI definition such as Motorola and Ericsson
  - CMMI has been established as a model to improve business results

- CMMI, staged, **uses 5 levels** to describe the maturity of the organization, same as predecessor CMM
  - Vastly improved version of the CMM
  - Emphasis on business needs, integration and institutionalization

The CMMI Model

- The CMM is a benchmark for measuring the maturity of an organization's system process.
- It describes the maturity of the company based upon the project the company is dealing with and the clients.



1. **Maturity Level 1: Initial**
   - Processes are disorganized, ad-hoc and chaotic.
   - Success of organization depends on the effort and competence of the people within.
   - Company has no standard process for software development.
   - Organization may not be able to repeat their past success.

2. **Maturity Level 2: Repeatable**
   - Basic project management techniques are established and success could be repeatable.
   - Program management is a key characteristic of a level 2 organization.

The CMMI Model

3. Maturity Level 3: Defined

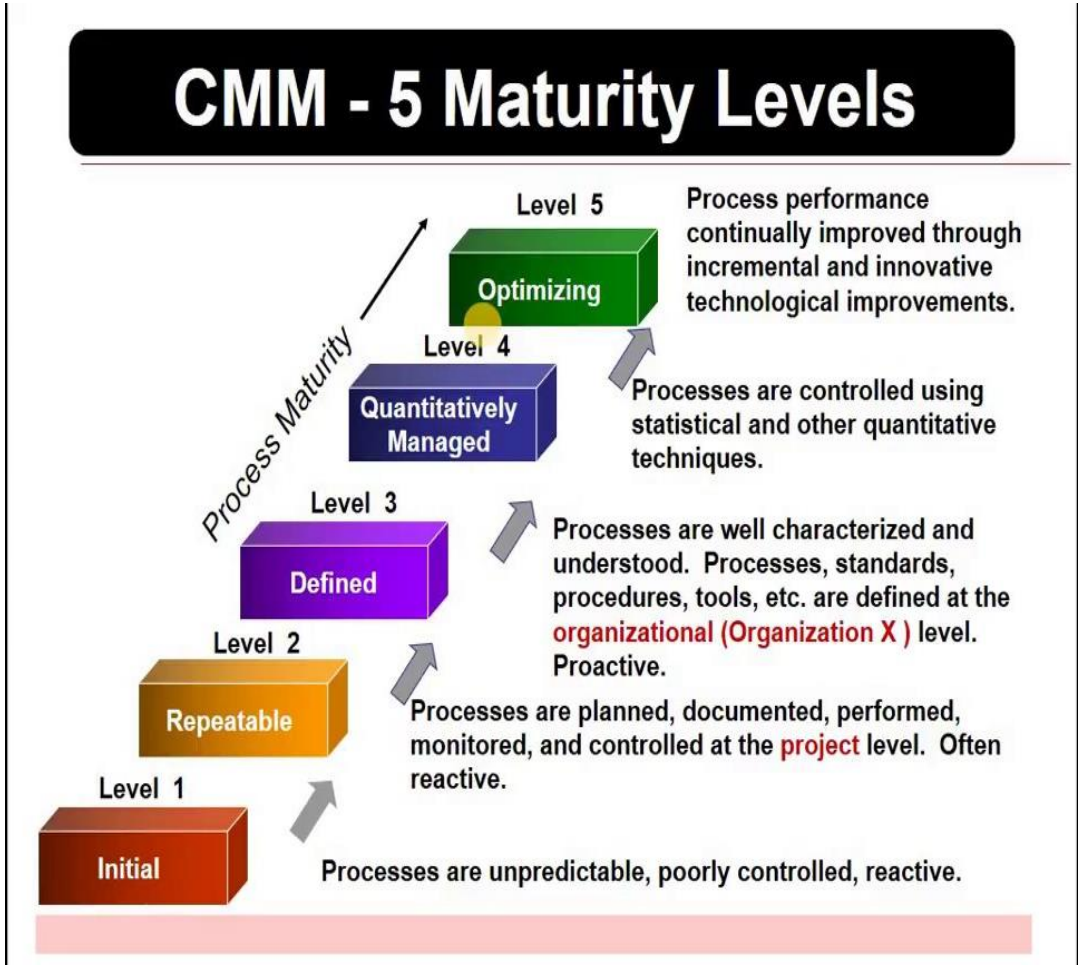
- At this level, organization has developed its own standards.
- The software process for both management and engineering activities are documented, standardized and integrated.

4. Maturity Level 4: Managed

- The performance and process is controlled and monitored through statistical and quantitative techniques.
- Organization set a quantitative quality goal.

5. Maturity Level 5: Optimizing

- Processes are constantly being input through monitoring feedback from current processes and introducing innovative process.



8.7 SQA plan

SQA plan

- A Software Quality Assurance (SQA) plan **is a comprehensive document** that outlines the approach, activities, processes, and resources **required to ensure the quality of a software product** throughout its development life cycle.
- The SQA plan **serves as a roadmap** for the entire quality assurance process and helps **ensure that the software meets the specified quality standards.**
- Below are the key components typically included in an SQA plan:
    1. Introduction
    2. Scope and Objectives
    3. SQA Processes
    4. Quality Standards and Criteria
    5. Testing and Review Activities
    6. Approval and Sign-off

8.8 Software certification

Software Certification

- Software certification refers to the **process of evaluating and officially recognizing that a software product or system meets specified standards, requirements, or criteria.**
- Certification is typically **carried out by an independent third party** or **a certifying authority** to provide assurance to users, customers, and stakeholders that the software meets certain quality and performance standards.
- Software certification can cover various aspects, including functionality, security, interoperability, and compliance with industry or regulatory standards.

Purpose of Software Certification

1. **Quality Assurance:** Certification ensures that a software product adheres to established quality standards and best practices.
2. **Interoperability:** Certifying compliance with interoperability standards ensures that the software can work seamlessly with other systems and technologies.
3. **Security:** Certification may include assessments of the software's security features to ensure it meets industry or regulatory security standards.
4. **Regulatory Compliance:** Some industries have specific regulations, and software may need to be certified to demonstrate compliance with these regulations.

Certification Processes:

1. **Testing and Evaluation:** The software undergoes thorough testing and evaluation to assess its compliance with predefined criteria.
2. **Documentation Review:** Certifying authorities may review documentation, design specifications, and other relevant materials to ensure they meet established standards.
3. **Audit and Inspection:** Certification processes may include on-site inspections or audits to verify that the development processes align with industry standards.

Certification Bodies:

1. **Industry-specific Organizations:** Some industries have specific organizations responsible for certifying software products within that industry.
2. **Government Agencies:** In certain cases, government agencies may be involved in certifying software, especially when regulatory compliance is a requirement.
3. **Independent Testing Laboratories:** Third-party testing organizations or laboratories may offer certification services, providing an unbiased evaluation of the software.

Certification Standards:

- **ISO Standards:** The International Organization for Standardization (ISO) has standards related to software quality, such as ISO/IEC 25010 for software product quality.
- **Industry-specific Standards:** Different industries may have their own set of standards that software must meet for certification.

What are SQA, SQP, SQC, and SQM?

SQA includes all 4 elements…

1. **Software Quality Assurance** – establishment of network of organizational procedures and standards leading to high-quality software
2. **Software Quality Planning** – selection of appropriate procedures and standards from this framework and adaptation of these to specific software project
3. **Software Quality Control** – definition and enactment of processes that ensure that project quality procedures and standards are being followed by the software development team
4. **Software Quality Metrics** – collecting and analyzing quality data to predict and control quality of the software product being developed

Short Answer Questions:

1. Define quality. What is software quality?
2. List any four software quality attributes.
3. Define Quality management.
4. Define SQA.
5. Define Software reliability. List any two software reliability metrics.
6. Define availability.
7. Define MTTF and MTTR.
8. Mention the types of software maintenance.

Brief Answer Questions:

1. What are the attributes of software quality? Explain each of them in short.
2. How is Capability Maturity Model Integration (CMMI) used for software process improvement?
3. How quality management is essential in software development? How is this performed?
4. Define SQA. What are the major activities in SQA? Mention its merits and demerits.
5. What is software maintenance? Explain its types with suitable examples.
6. Why software maintenance is required?
7. Explain the ISO 9001 core processes.
8. Explain the levels in CMMI model.
9. How is Capability Maturity Model Integration (CMMI) used for software process improvement?
10. What is SQA plan?
11. Who provides software certification? What is the purpose of software certification?
12. Explain the software certification process.

End of Chapter