Fair Use Disclaimer Notice

The material used in this presentation i.e., pictures/graphs/text, etc. is solely intended for educational/teaching purpose, offered free of cost to the students for use under special circumstances of Online Education due to COVID-19 Lockdown situation and may include copyrighted material - the use of which may not have been specifically authorized by Copyright Owners. It’s application constitutes Fair Use of any such copyrighted material as provided in globally accepted law of many countries. The contents of presentations are intended only for the attendees of the class being conducted by the presenter.

Outlines

- Quality
- Quality Control
- Quality Assurance
- Software Quality
- Software Quality Assurance

Software Quality Engineering

- **Software:**
  - Computer **software** is a **collection** of **data** or **computer instructions** that **tell** the computer **how to work**
  - **Software** is a **set of instructions**, **data** or **programs** used to **operate computers** and **execute specific tasks**

- **Quality:**
  - The **degree** of **excellence** of something
  - General **excellence** of **standard**

- **Engineering:**
  - The **branch** of **science** and **technology** concerned with the **design**, **building**, and **use of engines**, **machines**, and **structures**

Assurance

- **Certainty** about something
- **Confidence** of one's own **abilities**
- A **positive declaration** on a product or service, which gives **confidence**
- **Certainty** of a product or a service
- It provides a **guarantee** that the product will **work** without any problems as per the **expectations** or **requirements**

Quality Assurance (QA)

- **QA** is a system of activities **designed** to ensure *production* that meets pre-established *requirements* and *standards*
- An activity to ensure that an organization is providing the **best possible product** or **service** to customers.
- **QA** focuses on **improving** the processes to deliver **Quality Products** to the customer.
- An organization has to **ensure**, that processes are **efficient** and **effective** as per the quality standards defined for software products.
- **Quality Assurance** is popularly known as **QA Testing**.

Quality Assurance (QA)

- **QA** is a set of activities for *ensuring* **quality** in the **processes** by which products are developed. Hence, it is *process oriented*.
- **Everyone** in team involved in *developing* the **product** is responsible for **quality assurance**.
- **QA** aims to prevent **defects** with a focus on the process used to make the product. It is a *proactive quality process*.
- The **goal** of **QA** is to improve *development* and test *processes* so that *defects do not arise* when the product is being developed. Therefore, it is a *managerial tool*.
- Example: **Verification**

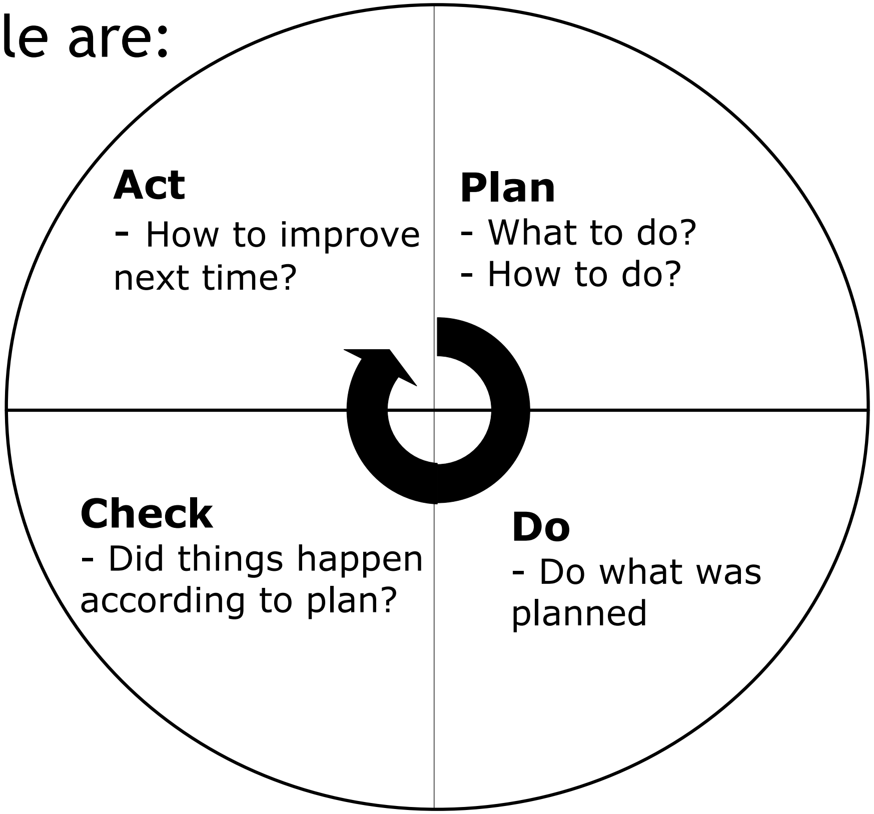
Quality Assurance (QA)

- **What** does it **do**?
  - Prevents **quality problems** through planned and systematic activities including documentation.

- **How** does it **do**?
  - Establish a **good quality management system** and the assessment of its adequacy.
  - Periodic conformance **audits** of the operations of the system.

Quality Assurance Process

- QA has a defined cycle called PDCA cycle or Deming cycle.
- The Phases of this cycle are:
  - Plan
  - Do
  - Check
  - Act


Act
- How to improve next time?
Plan
- What to do?
- How to do?
Check
- Did things happen according to plan?
Do
- Do what was planned

These steps are **repeated** to ensure that processes followed in the organization are evaluated and improved on a periodic basis

Quality Assurance Process

- **Plan** - Organization *plans* and *establishes* the **process** related objectives and *determines* the processes that are required to **deliver** a high-Quality end product
- **Do** - *Development* and *testing* of **Processes** and also "**do**" **changes** in the processes
- **Check** - *Monitoring* of processes, *modify* the processes, and *check* whether it **meets** the predetermined objectives
- **Act** - *Implement* **actions** that are necessary to **achieve improvements** in the processes

An organization must use **Quality Assurance** to ensure that the product is *designed* and *implemented* with **correct procedures**.
This helps **reduce** problems and errors, in the **final product.**

Quality Control (QC)

- Software Engineering **process** used to ensure **quality** in a product or a service
- QC **does not deal** with the **processes** used to create a product
- QC examines the **quality** of the "*end products*" and the final outcome
- The main **aim** of **Quality control** is to check whether the products **meet** the specifications and requirements of the customer
  - If an *issue* or *problem* is identified, it needs to **be fixed** before delivery to the customer
- QC also evaluates **people** on **their quality level skill sets** and imparts *training* and *certifications*.
  - This **evaluation** is required for the service based organization and helps provide "**perfect**" service to the customers

Quality Control (QC)

- **QC** is a set of activities for *ensuring* **quality** in products. The activities **focus** on *identifying* defects in the actual products produced. Hence, it is **product oriented**.

- **Quality control** is usually the **responsibility** of a specific team that **tests** the product for defects.

- **QC** aims to *identify* (and correct) **defects** in the finished product. Quality control, therefore, is a **reactive process** as well as a **corrective tool**.

- The **goal** of QC is to *identify* **defects** after a product is developed and before it's **released**.

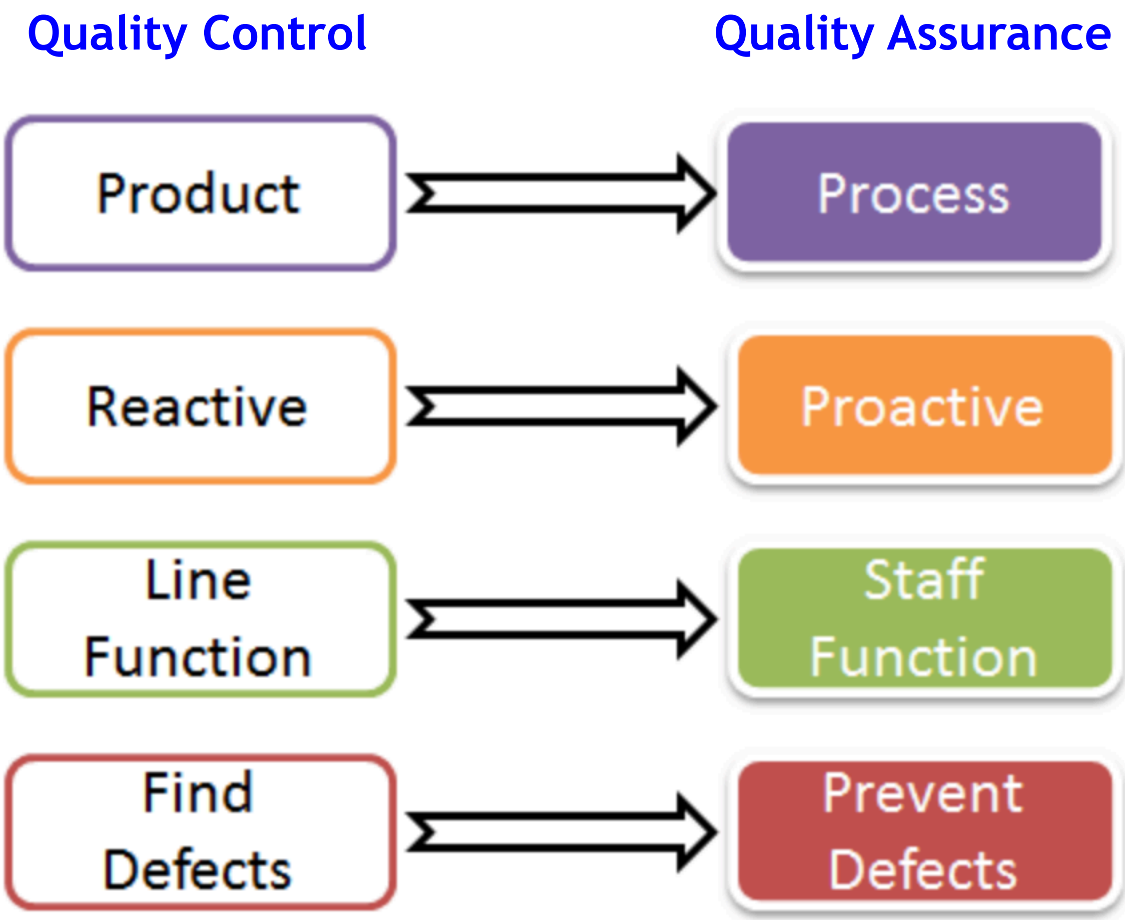
- Example: **Validation**

Quality Control (QC)

- **What** does it **do**?
  - Perform the **activities** or **techniques** used to *achieve and maintain* the product quality, process and service.

- **How** does it **do**?
  - Finding & eliminating **sources** of *quality problems* through tools & equipment so that customer's requirements are continually met.
    - When **statistical tools & techniques** are applied to finished products (process outputs), they are called as **Statistical Quality Control** (SQC) & comes under **QC**.
    - **Statistical Tools & Techniques** can be applied in both QA & QC. When they are applied to processes (process inputs & operational parameters), they are called **Statistical Process Control** (SPC); & it becomes the part of **QA**.

QC vs. QA



Quality Control	Quality Assurance
Product	Process
Reactive	Proactive
Line Function	Staff Function
Find Defects	Prevent Defects

QC and QA Activities

Quality Control Activities	Quality Assurance Activities
Walkthrough	Quality Audit
Testing	Defining Process
Inspection	Tool Identification and selection
Checkpoint review	Training of Quality Standards and Processes

These activities are concerned with QA and QC of any product and not essentially software

QA becomes **SQA** (Software Quality Assurance) [*with respect to Software*]

QC becomes **Software Testing** [*with respect to Software*]

Software Quality

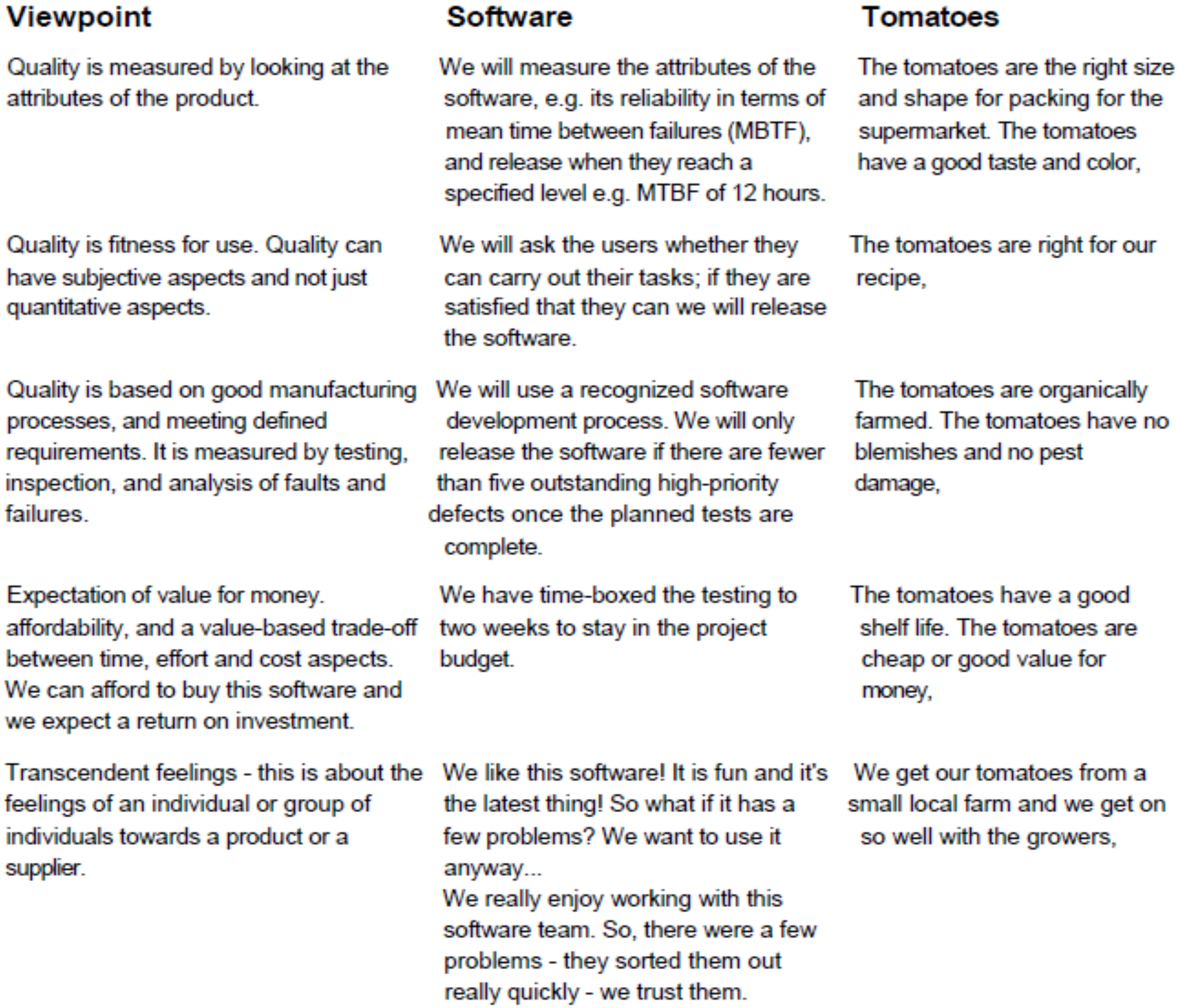
- **Quality** is a **complex** concept. It means different thing to different people - **Subjective**.
- **Software quality** is the **degree** to which software possesses a desired combination of attributes (e.g., ***reliability***, ***interoperability***) – **IEEE Definition**
  - Maintainability is per excellence, a **yardstick** by which **quality** of a **software** can be **judged**. Paradoxical as it may seem, software that has been developed with a view to being changed is likely to need changing less than any other. It will be software quality - Manns and Coleman
  - **Quality** is **conformance** to requirements - Crosby
  - **Software quality** mean **fitness** for purpose - Ould
  - **Quality** is **all the features** that allow a product to satisfy stated or implied needs at an affordable cost - ISO-8402
  - Given **five views** of **quality**; *transcendent*, *product based*, *user based*, *manufacturing based*, *value based* - Gravin

Software Quality

- **Five views** of **quality** in a comprehensive manner as follows:

- **Transcendental View**
  - It is not specific to software quality alone but has been applied in other complex areas of everyday life.
- **User View**
  - It perceives **quality** as fitness for purpose. According to this view, while evaluating the quality of a product, one must ask the key question: "Does the product satisfy user needs and expectations?"
- **Manufacturing View**
  - **Quality** is understood as conformance to the specification. The quality level of a product is determined by the extent to which the product meets its specifications.
- **Product View**
  - **Quality** is viewed as tied to the inherent characteristics of the product. A product's inherent characteristics, that is, internal qualities, determine its external qualities.
- **Value Based View**
  - **Quality** in this perspective, depends on the *amount a customer is willing to pay* for it.

Software Quality



Viewpoint	Software	Tomatoes
Quality is measured by looking at the attributes of the product.	We will measure the attributes of the software, e.g. its reliability in terms of mean time between failures (MBTF), and release when they reach a specified level e.g. MTBF of 12 hours.	The tomatoes are the right size and shape for packing for the supermarket. The tomatoes have a good taste and color,
Quality is fitness for use. Quality can have subjective aspects and not just quantitative aspects.	We will ask the users whether they can carry out their tasks; if they are satisfied that they can we will release the software.	The tomatoes are right for our recipe,
Quality is based on good manufacturing processes, and meeting defined requirements. It is measured by testing, inspection, and analysis of faults and failures.	We will use a recognized software development process. We will only release the software if there are fewer than five outstanding high-priority defects once the planned tests are complete.	The tomatoes are organically farmed. The tomatoes have no blemishes and no pest damage,
Expectation of value for money. affordability, and a value-based trade-off between time, effort and cost aspects. We can afford to buy this software and we expect a return on investment.	We have time-boxed the testing to two weeks to stay in the project budget.	The tomatoes have a good shelf life. The tomatoes are cheap or good value for money,
Transcendent feelings - this is about the feelings of an individual or group of individuals towards a product or a supplier.	We like this software! It is fun and it's the latest thing! So what if it has a few problems? We want to use it anyway... We really enjoy working with this software team. So, there were a few problems - they sorted them out really quickly - we trust them.	We get our tomatoes from a small local farm and we get on so well with the growers,

Software Quality Assurance

- **SQA** is a set of activities for *ensuring* **quality** in software engineering processes.
- It *ensures* that **developed software** *meets* and *complies* with the defined or **standardized quality specifications**.
- **SQA** is an ongoing process within the Software Development Life Cycle (SDLC) that routinely *checks* the **developed software** to *ensure* it meets the **desired quality measures**.

Software Quality Assurance

- **SQA** includes the following activities
  - Process definition and implementation
  - Auditing
  - Training
- **Processes** could be
  - Software Development Methodology
  - Project Management
  - Configuration Management
  - Requirements Development/Management
  - Estimation
  - Software Design
  - Testing, etc.
- Once the **processes** have been defined and implemented, **Quality Assurance** has the following responsibilities
  - Identify the *weaknesses* in the processes
  - Correct those *weaknesses* to continually improve the process

Software Testing (QC) vs. SQA

Software Testing (QC)	Software Quality Assurance
Software Testing is to test a *product* for *problems* before the product goes live	**SQA** is about engineering process that *ensures quality*
Involves actives concerning verification of product Example - **Review Testing**	**Involves** activities related to the implementation of processes, procedures, and standards. Example - **Audits Training**
Product focused	**Process** focused
Corrective technique	**Preventive** technique
Reactive measure	**Proactive** measure
The **scope** of Software Testing applies to a particular product being tested	The **scope** of SQA applied to all products that will be created by the organization

Software Quality Assurance: Best Practices

- **Create** a Robust Testing Environment
- **Select** release criteria carefully
- **Apply** automated testing to *high-risk areas* to save money. It helps to fasten the entire process.
- **Allocate** Time Appropriately for each process
- It is **important** to prioritize bugs fixes based on software usage
- Form **dedicated security** and performance testing team
- **Simulate** customer accounts similar to a production environment

Software Quality Assurance: 5 Primary Functions

1. Technology transfer

- This function **involves** getting a product design document as well as trial and error data and its **evaluation**. The documents are distributed, checked and approved

2. Validation

- Validation **master plan** for the entire system is prepared. Approval of test criteria for ***validating*** product and process is set. Resource planning for execution of a validation plan is done

3. Documentation

- This function **controls** the distribution and archiving of documents. Any change in a document is made by adopting the proper change control procedure. Approval of all types of documents

4. Assuring Quality of products

5. Quality improvement plans

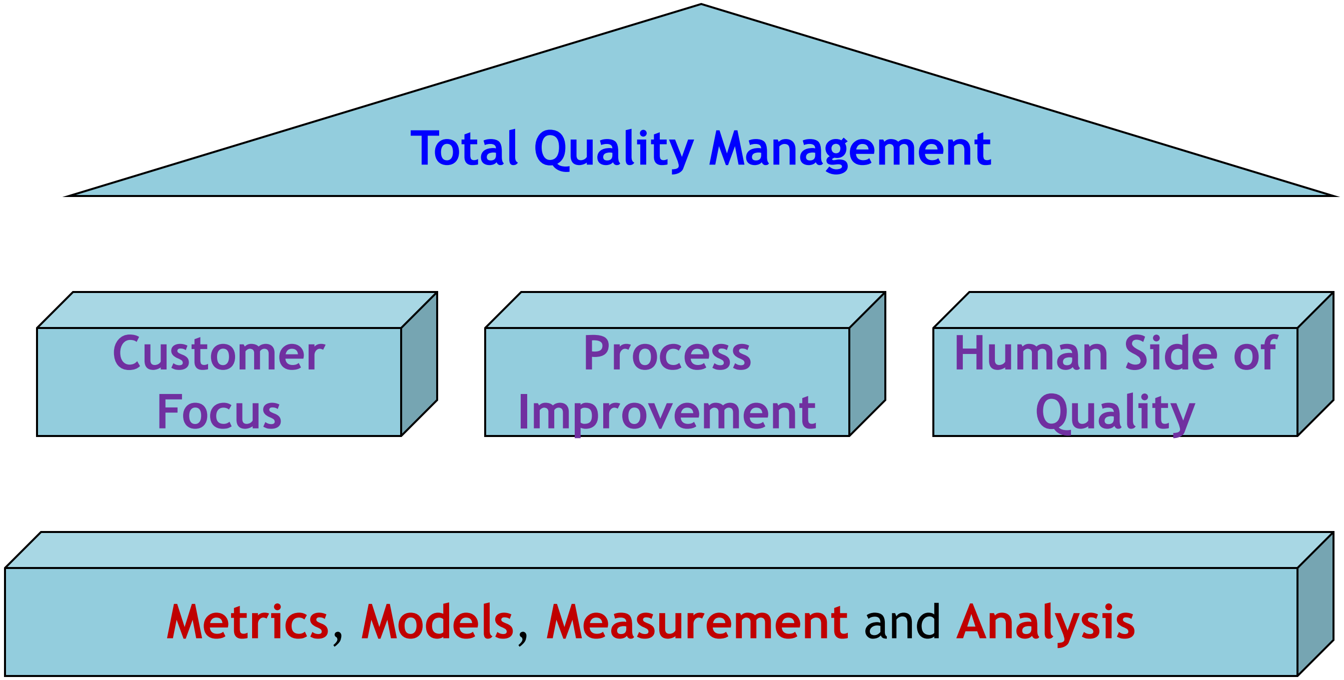
Total Quality Management (TQM)

- **Continuous improvement** in **performance** at every level & area
- **Ensuring customer satisfaction**
- **Combines Management & Statistical** techniques
- **Strive** for **zero defect** product
- **Requires commitment**, **leadership** and **training**

TQM introduced **quality management** through *statistical analysis*

Set of practices successfully introduced by **Dr. Edward Deming** in Japan after World War II

TQM Framework


Total Quality Management

Customer Focus

Process Improvement

Human Side of Quality

Metrics, Models, Measurement and Analysis

TQM: Implementations/Certifications

- Few **major implementations** for software are:
  - Six Sigma
  - Capability Maturity Model Integration (CMMI)
  - ISO 9001:2000
  - Test Maturity Model (TMM)

Six Sigma

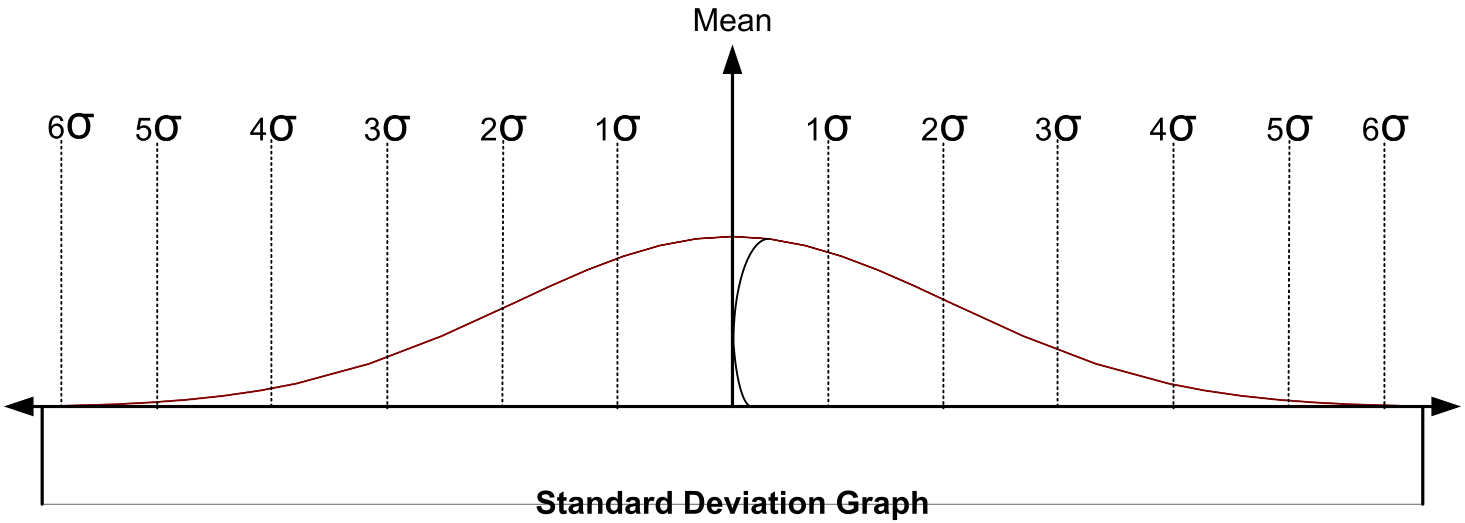
- A **quality management technique** introduced by **Motorola** and followed by General Electric (GE) and Allied Signals.

- **Six Sigma** targets **99.99966%** accuracy i.e., only 3.4 defective products are allowed per million production.

- **Heavily** based on statistical measurements, analysis and improvement.

Six Sigma: Concept

- **Standard Deviation** (σ) = square root [sum (x - μ)^2 / N].
- **6σ** from mean cover **99.99966%** area under the curve



CMMI

- The **most widely known** and acclaimed **model** for technology companies.

- **Developed** by Software Engineering Institute (SEI) at Carnegie Mellon University, Pittsburg, USA, based on funding from DoD (Department of Defense).

- **CMMI** is a **major** release after of all CMM models.

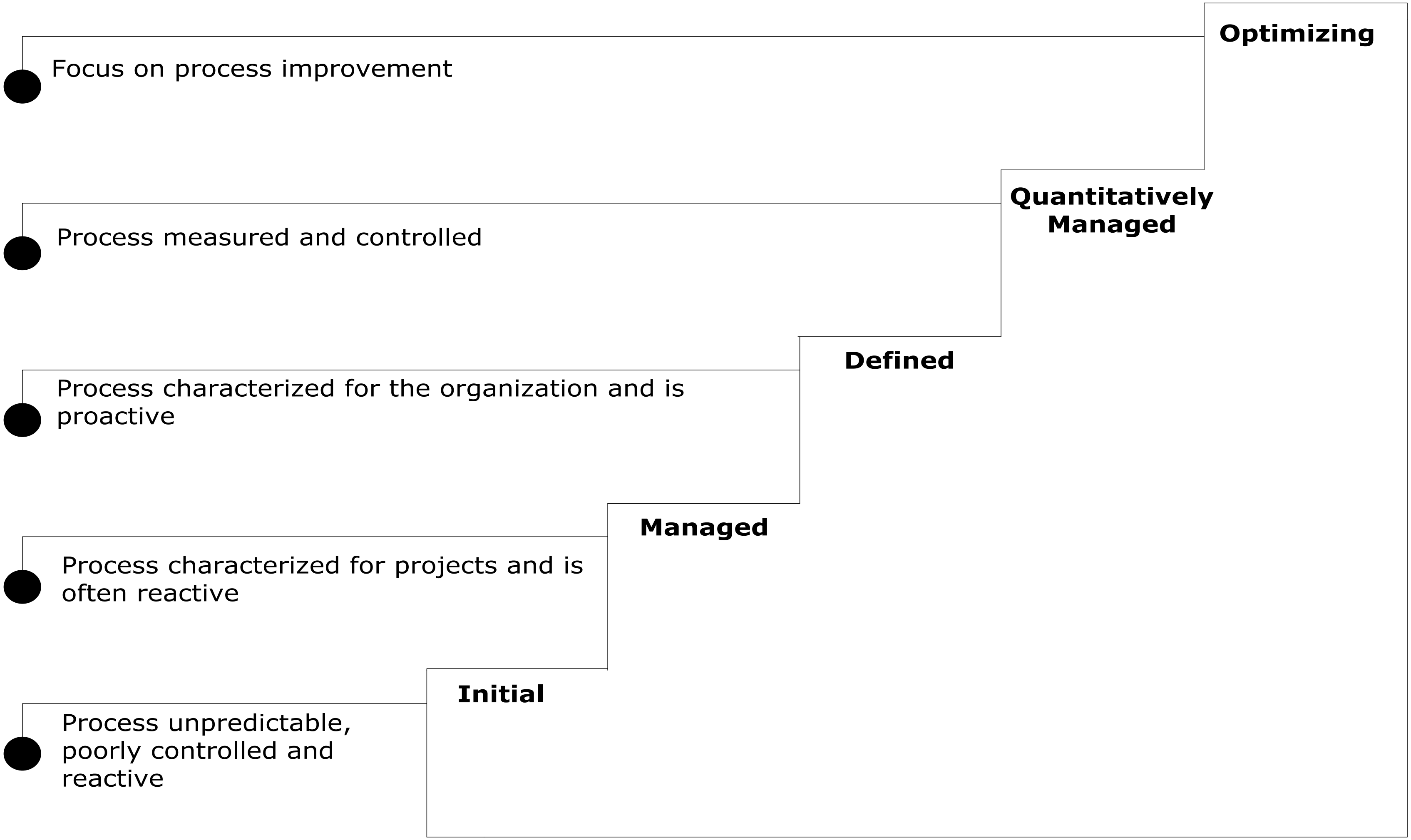
- **SEI** has announced the sun setting CMM as it would not offered after 2005.

CMMI

- The **CMMI** has two representations:
  - **Continuous**
  - **Staged**

- *Staged representation* is based on old CMM-SW model, having five maturity levels. These are:
  - Initial
  - Managed
  - Defined
  - Qualitatively Managed
  - Optimizing

CMMI: 5 Maturity Levels



Optimizing

- Focus on process improvement

Quantitatively Managed

- Process measured and controlled

Defined

- Process characterized for the organization and is proactive

Managed

- Process characterized for projects and is often reactive

Initial

- Process unpredictable, poorly controlled and reactive

CMMI: 5 Maturity Levels

- **Level 1** - **Initial:** In this stage the **quality environment** is unstable. Simply, **no processes** have been followed or documented
- **Level 2** - **Repeatable: Some processes** are followed which are repeatable. This level ensures processes are followed at the project level.
- **Level 3** - **Defined: Set of processes** are defined and documented at the organizational level. Those defined processes are subject to some degree of improvement.
- **Level 4** - **Managed:** This level uses **process metrics** and effectively controls the processes that are followed.
- **Level 5** - **Optimizing:** This level **focuses** on the continuous improvements of the processes through learning & innovation.

ISO 9001:2000

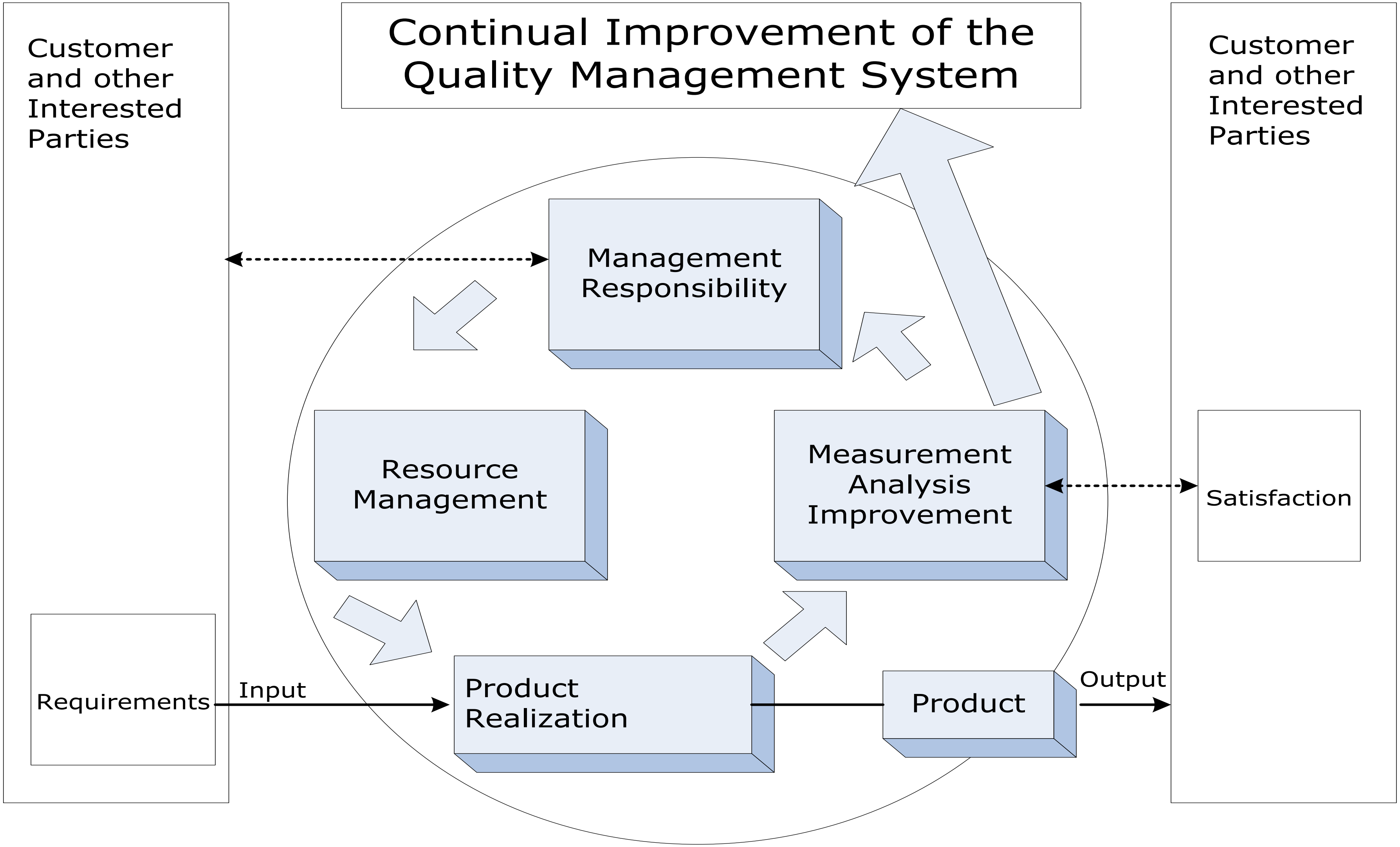
- A **Quality Management System** (QMS) by ISO.
- **Generic standard** for all industries; **not specifically** designed for *IT/Software*.
- **Recognized** by Register of Certified Auditors (IRCA) and International Auditors and Training Certificate Association (IATCA)
- An **organization** who wishes to **be certified as ISO 9000** is audited based on their functions, products, services and their processes.

ISO 9001:2000

- The **main objective** is to review and verify whether the organization is ***following the process*** as expected and check whether existing processes need ***improvement***.

- **ISO 9000** helps:
  - Increase the ***profit*** of the organization
  - Improves Domestic and International ***trade***
  - Reduces ***waste*** and increase the ***productivity*** of the employees
  - Provide ***Excellent*** customer ***satisfaction***

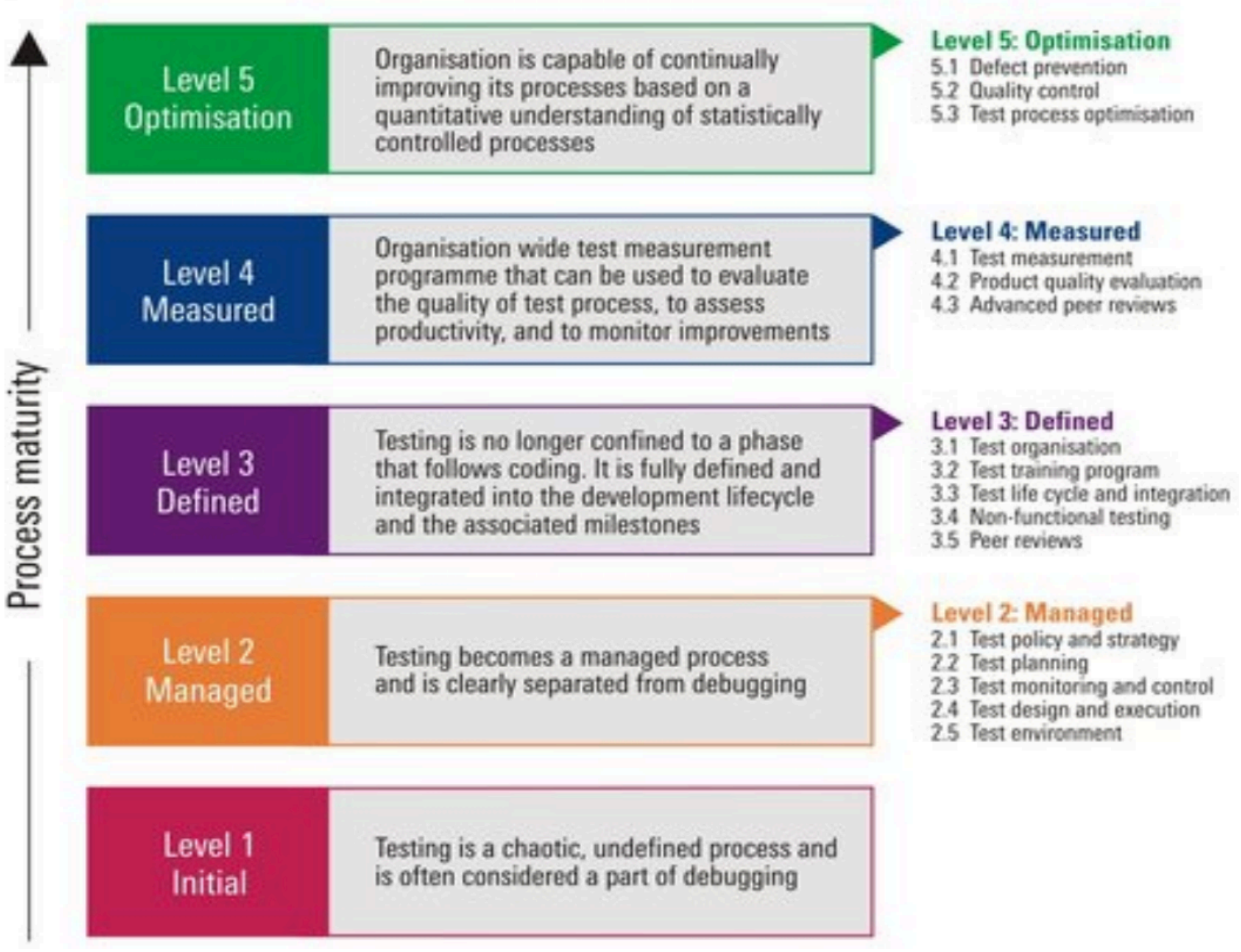
ISO 9001:2000


Continual Improvement of the Quality Management System
Customer and other Interested Parties
Customer and other Interested Parties
Management Responsibility
Resource Management
Measurement Analysis Improvement
Satisfaction
Requirements
Input
Product Realization
Product
Output

Test Maturity Model (TMM)

- This **model** (5 levels) *assesses* the maturity of processes in a Testing Environment.
- **Level 1** - **Initial:** There is **no quality standard** followed for testing processes and only ad-hoc methods are used at this level
- **Level 2** - **Definition: Defined** process. **Preparation** of test strategy, plans, test cases are done.
- **Level 3** - **Integration: Testing** is **carried out throughout** the software development lifecycle (SDLC) - which is nothing but integration with the development activities, E.g., V- Model.
- **Level 4** - **Management and Measurement: Review** of requirements and designs takes place at this level and criteria has been set for ***each level of testing***
- **Level 5** - **Optimization: Many preventive techniques** are used for testing processes, and tool support(*Automation*) is used to improve the ***testing standards*** and ***processes***.

Test Maturity Model (TMM)



Level 5 Optimisation — Organisation is capable of continually improving its processes based on a quantitative understanding of statistically controlled processes

Level 5: Optimisation
5.1 Defect prevention
5.2 Quality control
5.3 Test process optimisation

Level 4 Measured — Organisation wide test measurement programme that can be used to evaluate the quality of test process, to assess productivity, and to monitor improvements

Level 4: Measured
4.1 Test measurement
4.2 Product quality evaluation
4.3 Advanced peer reviews

Level 3 Defined — Testing is no longer confined to a phase that follows coding. It is fully defined and integrated into the development lifecycle and the associated milestones

Level 3: Defined
3.1 Test organisation
3.2 Test training program
3.3 Test life cycle and integration
3.4 Non-functional testing
3.5 Peer reviews

Level 2 Managed — Testing becomes a managed process and is clearly separated from debugging

Level 2: Managed
2.1 Test policy and strategy
2.2 Test planning
2.3 Test monitoring and control
2.4 Test design and execution
2.5 Test environment

Level 1 Initial — Testing is a chaotic, undefined process and is often considered a part of debugging

Quiz

- What is Good Software?
- Define Coupling and Cohesion
- Define Liskov’s Substitution Principle
- What is SOLID
- Define Law of Demeter

Quiz (online Material)

- A list of quality attributes and their definitions from Microsoft's online developer documentation.

https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ee658094(v=pandp.10)

- An in-depth report on quality attributes and the relationships between them.

https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=12433

- Details of the Coupling Factor measure.

http://www.arisa.se/compendium/node109.html#metric:CF

- More information about the Lack of Cohesion of Methods (LCOM) measure.

https://www.aivosto.com/project/help/pm-oo-cohesion.html#LCOM4

- Description of the Lack of Documentation metric and its calculation.

http://www.arisa.se/compendium/node121.html#metric:LOD

- SonarQube listing of various quality metrics

https://docs.sonarqube.org/latest/user-guide/metric-definitions/

Observer Pattern: https://sourcemaking.com/design_patterns/observer/cpp/1