CHAPTER 1:

SOFTWARE QUALITY ASSURANCE

Bộ môn Công Nghệ Phần Mềm
Khoa CNTT – Trường ĐH Ngoại Ngữ Tin Học TP.HCM

OBJECTIVES

HELP STUDENTS:

- The sortware quality challenge
- Get it target, love subject requirements
- Presentation get the concept related concept arrive QA/SQA/QC,
- Raw materials cause of part error soft
- Objectives and request when SQA
- Understand technical requirements ability, knowledge of the profession QA/SQA

THE SORTWARE QUALITY CHALLENGE

- The uniqueness of software quality assurance
- The environments for which SQA methods are developed

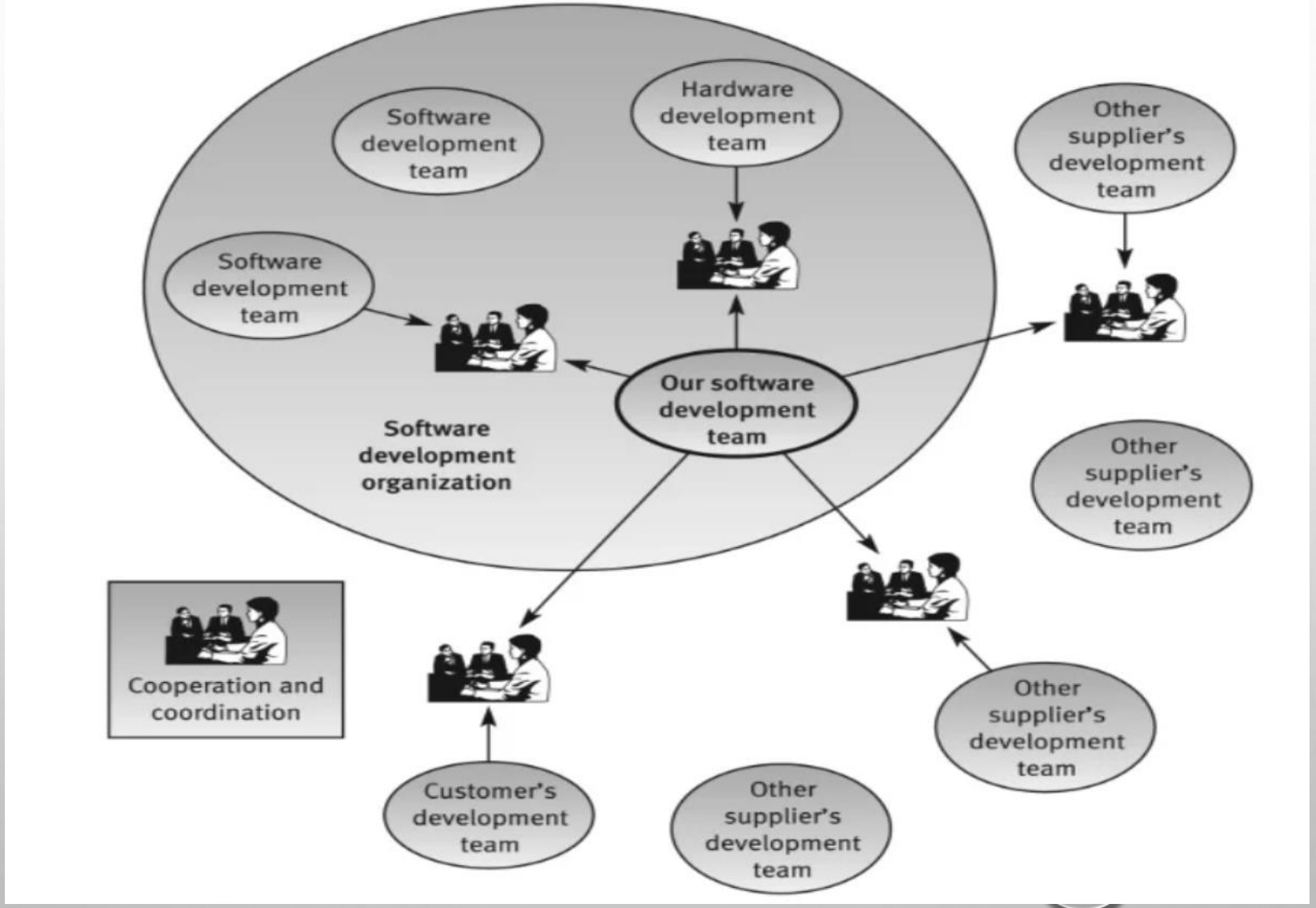
THE UNIQUENESS OF THE SOFTWARE DEVELOPMENT PROCESS

- High complexity
- Invisibility of the product
- Limited opportunities to detect defects ("bugs")

THE CHARACTERISTICS OF THE SQA ENVIRONMENT PROCESS

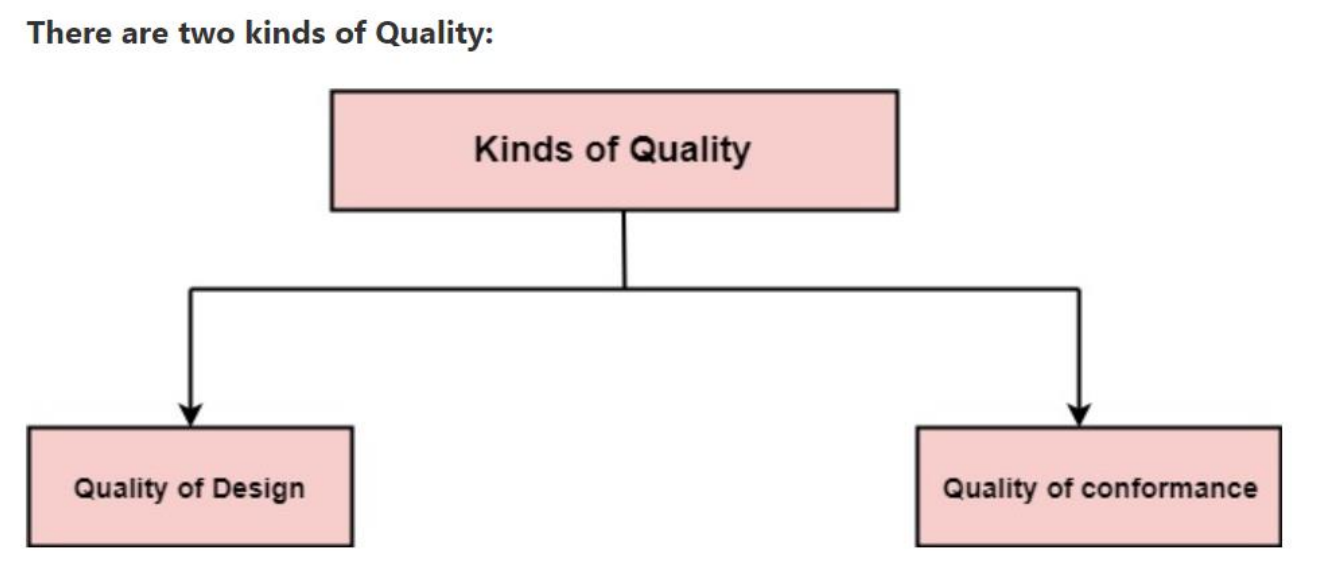
- Being contracted
- Subjection to customer-supplier relationship
- Requirement for teamwork
- Need for cooperation and coordination with other development teams
- Need for interfaces with other software systems
- Need to continue carrying out a project while the team changes
- Need to continue maintaining the software system for years

COOPERATION AND COORDINATION SCHEME FOR A SOFTWARE DEVELOPMENT PROJECT TEAM



WHAT IS QUALITY?

- Quality defines to any measurable characteristics such as correctness, maintainability, portability, testability, usability, reliability, efficiency, integrity, reusability, and interoperability.



WHAT IS QUALITY?

- **Quality of design:** quality of design refers to the characteristics that designers specify for an item. The grade of materials, tolerances, and performance specifications that all contribute to the quality of design.
- **Quality of conformance:** quality of conformance is the degree to which the design specifications are followed during manufacturing. Greater the degree of conformance, the higher is the level of quality of conformance.

- **SOFTWARE QUALITY:** Software quality is defined as the conformance to explicitly state functional and performance requirements, explicitly documented development standards, and inherent characteristics that are expected of all professionally developed software.
- **QUALITY CONTROL:** Quality control involves a series of inspections, reviews, and tests used throughout the software process to ensure each work product meets the requirements place upon it. Quality control includes a feedback loop to the process that created the work product.
- **QUALITY ASSURANCE:** Quality assurance is the preventive set of activities that provide greater confidence that the project will be completed successfully.
- **QUALITY ASSURANCE** focuses on how the engineering and management activity will be done?

➔ As anyone is interested in the quality of the final product, it should be assured that we are building the right product.

➔ It can be assured only when we do inspection & review of intermediate products, if there are any bugs, then it is debugged. This quality can be enhanced.

QUALITY ASSURANCE (QA)

What is Quality?

- Quality is extremely hard to define, and it is simply stated: "Fit for use or purpose." It is all about meeting the needs and expectations of customers with respect to functionality, design, reliability, durability, & price of the product.

What is Assurance?

- Assurance is nothing but a positive declaration on a product or service, which gives confidence. It is certainty of a product or a service, which it will work well. It provides a guarantee that the product will work without any problems as per the expectations or requirements

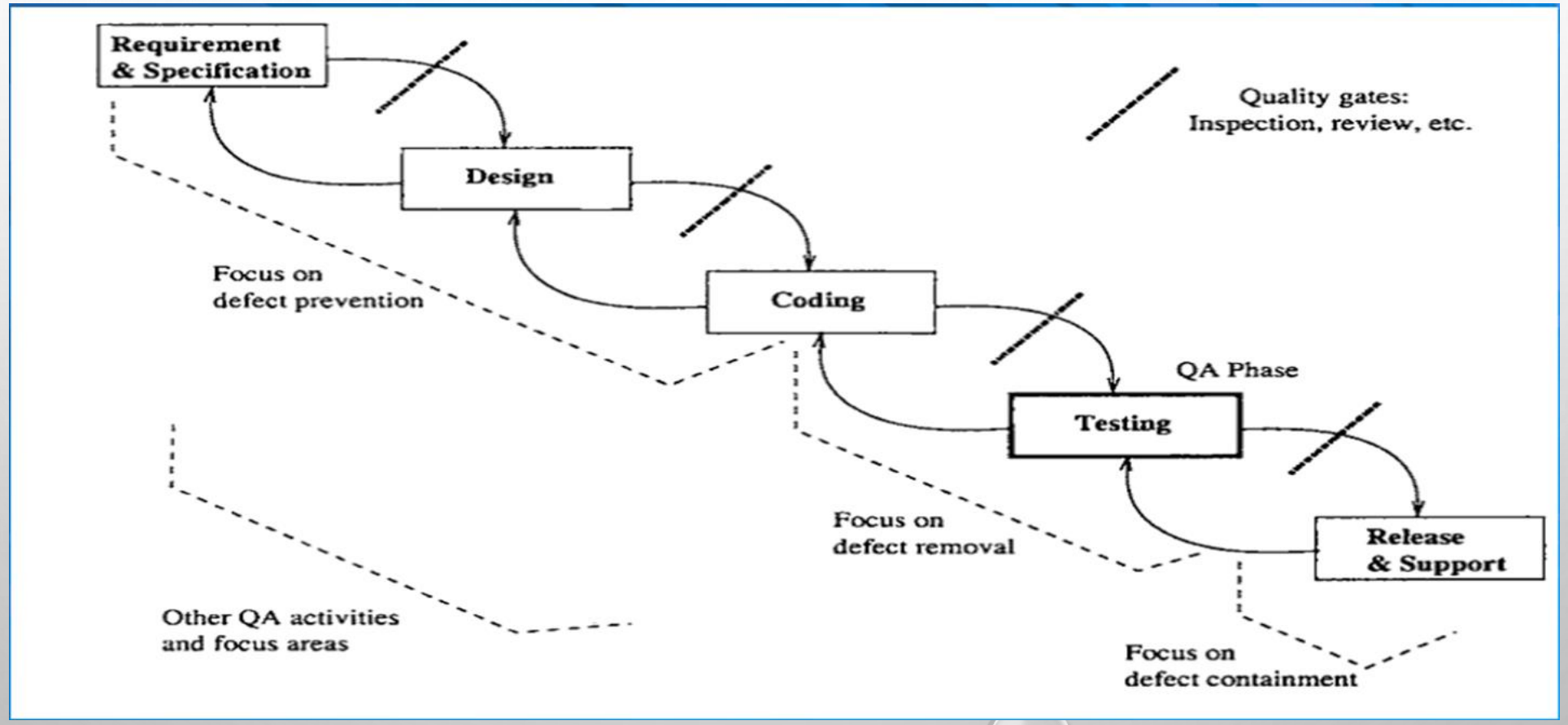
QUALITY ASSURANCE (QA)

Quality Assurance in Software is defined as a procedure to ensure the quality of software products or services provided to the customers by an organization. Quality assurance focuses on improving the software development process and making it efficient and effective as per the quality standards defined for software products. Quality Assurance is popularly known as QA Testing.

Software quality assurance (SQA) is a process which assures that all software engineering processes, methods, activities and work items are monitored and comply against the defined standards. These defined standards could be one or a combination of any like ISO 9000, CMMI model, ISO15504, etc.

SQA incorporates all software development processes starting from defining requirements to coding until release. Its prime goal is to ensure quality.

QA IN SOFTWARE DEVELOPMENT LIFE CYCLE



SOFTWARE QUALITY ASSURANCE

- **Software quality assurance** is a planned and systematic plan of all actions necessary to provide adequate confidence that an item or product conforms to establish technical requirements.
- A set of activities designed to calculate the process by which the products are developed or manufactured.

SOFTWARE QUALITY ASSURANCE ACTIVITIES

1. **Prepares an SQA plan for a project:** the program is developed during project planning and is reviewed by all stakeholders. The plan governs quality assurance activities performed by the software engineering team and the SQA group. The plan identifies calculation to be performed, audits and reviews to be performed, standards that apply to the project, techniques for error reporting and tracking, documents to be produced by the SQA team, and amount of feedback provided to the software project team.
2. **Participates in the development of the project's software process description:** the software team selects a process for the work to be performed. The SQA group reviews the process description for compliance with organizational policy, internal software standards, externally imposed standards (e.G. ISO-9001), and other parts of the software project plan

SOFTWARE QUALITY ASSURANCE ACTIVITIES

3. **Reviews software engineering activities to verify compliance with the defined software process:** the SQA group identifies, reports, and tracks deviations from the process and verifies that corrections have been made.
4. **Audits designated software work products to verify compliance with those defined as a part of the software process:** the sqa group reviews selected work products, identifies, documents and tracks deviations, verify that corrections have been made, and periodically reports the results of its work to the project manager.

SOFTWARE QUALITY ASSURANCE ACTIVITIES

5. **Ensures that deviations in software work and work products are documented and handled according to a documented procedure:** deviations may be encountered in the project method, process description, applicable standards, or technical work products.
6. **Records any noncompliance and reports to senior management:** non-compliance items are tracked until they are resolved.

QUALITY CONTROL

- **SOFTWARE QUALITY CONTROL (SQC)** is a set of activities for ensuring quality in software products. Software quality control is limited to the review/ testing phases of the software development life cycle and the goal is to ensure that the products meet specifications/ requirements.
- **SQC** is a set of activities designed to evaluate the quality of a component or system.

QUALITY CONTROL ACTIVITIES

1. Reviews

- Requirement review
- Design review
- Code review
- Deployment plan review
- Test plan review
- Test cases review

2. Testing Levels

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

QUALITY ASSURANCE V/S QUALITY CONTROL

- Quality assurance (QA) is a process to avoid mistakes/blunders and defects in a product and prevent obstacles when it is delivered to the customers. ISO 9000 defines quality assurance as “part of quality management focused on providing confidence that quality requirements will be fulfilled”. The complete quality assurance process is called the PDCA cycle or deming cycle.
- Quality control (QC) is a process by which one reviews the quality of all the factors influencing the manufacturing of a product. ISO 9000 defines quality control as “A part of quality management focused on fulfilling quality requirements”.

Quality Assurance (QA) is the set of actions including facilitation, training, measurement, and analysis needed to provide adequate confidence that processes are established and continuously improved to produce products or services that conform to specifications and are fit for use.	**Quality Control (QC)** is described as the processes and methods used to compare product quality to requirements and applicable standards, and the actions are taken when a nonconformance is detected.
QA is an activity that establishes and calculates the processes that produce the product. If there is no process, there is no role for QA.	**QC** is an activity that demonstrates whether or not the product produced met standards.
QA helps establish process	**QC** relates to a particular product or service
QA sets up a measurement program to evaluate processes	**QC** verified whether particular attributes exist, or do not exist, in a explicit product or service.
QA identifies weakness in processes and improves them	**QC** identifies defects for the primary goals of correcting errors.
Quality Assurance is a managerial tool.	Quality Control is a corrective tool.
Verification is an example of QA.	Validation is an example of QC.

SOFTWARE - IEEE DEFINITION

DEFINITION OF SOFTWARE IS REALLY NOT SIMPLE. SIMPLY CODE?

ACCORDING TO THE IEEE:

- Software is: computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.

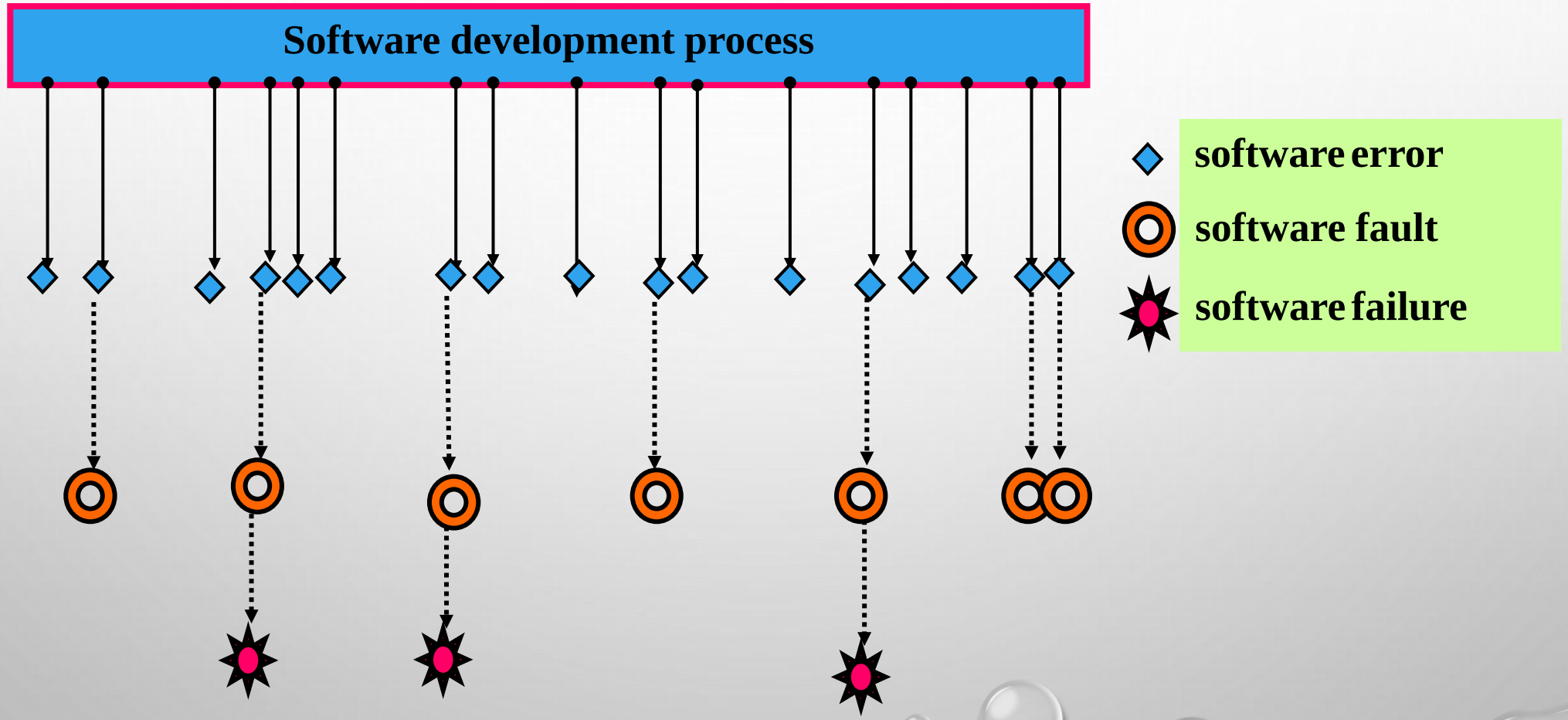
ISO DEFINITION (FROM ISO 9000-3) lists four components necessary to assure the quality of the software development process and years of maintenance:

- Computer programs (code)
- Procedures
- Documentation
- Data necessary for operating the software system.

BASIC DEFINITIONS

- SOFTWARE ERROR – MADE BY PROGRAMMER
  - Syntax (grammatical) error
  - Logic error (multiply instead of adding two operands)
- SOFTWARE FAULT
  - All software errors may not cause software faults
  - That part of the software may not be executed
- SOFTWARE FAILURES – HERE'S THE INTEREST.
  - A software fault becomes a software failure when/if it is activated.
  - Faults may be found in the software due to the way the software is executed or
  - Other constraints on the software's execution, such as execution options.

SOFTWARE ERRORS, SOFTWARE FAULTS AND SOFTWARE FAILURES



THE NINE CAUSES OF SOFTWARE ERRORS

1. Faulty requirements definition
2. Client-developer communication failures
3. Deliberate deviations from software requirements
4. Logical design errors
5. Coding errors
6. Non-compliance with documentation and coding instructions
7. Shortcomings of the testing process
8. User interface and procedure errors
9. Documentation errors

1.FAULTY REQUIREMENTS DEFINITION

- Usually considered the root cause of software errors
- Incorrect requirement definitions
  - Simply stated, ‘wrong’ definitions (formulas, etc.)
- Incomplete definitions
  - Unclear or implied requirements
- Missing requirements
  - Just flat-out ‘missing.’ (E.G. Program element code)
- Inclusion of unneeded requirements
  - (Many projects have gone amuck for including far too many requirements that will never be used.
  - Impacts budgets, complexity, development time, …

2. CLIENT-DEVELOPER COMMUNICATION FAILURES

- Misunderstanding of instructions in requirements documentation (written / graphical instructions)
- Misunderstanding of written changes during development.
- Misunderstanding of oral changes during development.
- Lack of attention
  - to client messages by developers dealing with requirement changes and
  - to client responses by clients to developer questions
- Very often, these very talented individuals come from different planets, it seems.
- Clients represent the users; developers represent a different mind set entirely sometimes!

3. DELIBERATE DEVIATIONS FROM SOFTWARE REQUIREMENTS

- **Developer reuses previous / similar work to save time.**
- **Often reused code needs modification which it may contain contain unneeded / unusable extraneous code.**

- **Book suggests developer(s) may overtly omit functionality due to time / budget pressures.**
  - **Another bad choice; system testing will uncover these problems to everyone's dismay!**
  - **I have never seen this done intentionally – but understand it!**
- **Developer inserting unapproved 'enhancements' (perfective coding; a slick new sort / search….); May also ignore some seemingly minor features, which sometimes are quite major.**
  - **Have seen this and it too causes problems and embarrassment during reviews.**

- **4. LOGICAL DESIGN ERRORS**
  - **Definitions that represent software requirements by means of erroneous algorithms.**
    - **Yep! Wrong formulas;**
    - **Wrong decision logic tables;**
    - **Incorrect descriptions in text**
  - **Process definitions: procedures specified by systems analyst not accurately reflecting the real business process**
    - **Note: all errors are not necessarily software errors.**
    - **This seems like a procedural error, and likely not a part of the software system…but they are errors nonetheless!**
  - **Erroneous definition of boundary condition – a common source of errors**
    - **The "absolutes" like 'no more than' "fewer than," "n times or more;" "the first time," etc.**

4. LOGICAL DESIGN ERRORS

- **Omission of required software system states**
  - **If rank is >= O1 and RPI is numeric, then….Easy to miss an action based on the software system state.**
- **Omission of definitions / procedures concerning reactions to illegal operation of the software system.**
  - **Including not only code to detect an illegal operation but failure to design the computer software reaction to this: gracefully terminate, sound alarm, etc. ??**

5. CODING ERRORS

- Too many to try to list.
  - Syntax errors (grammatical errors)
  - Logic errors (program runs; results wrong)
  - Run-time errors (crash during execution)
- Know the differences. Be also able to supply an example. Examples do not ‘define’ something, but they may be used to ‘support’ or ‘supplement’ a definition.

6. NON-COMPLIANCE WITH DOCUMENTATION AND CODING INSTRUCTIONS

- Non-compliance with published templates (formats)
- Non-compliance with coding standards
  - (Standards and Integration Branch)
  - Size of program;
    - Other programs must be able to run in environment!
  - Coding: Data Elements and Codes: AFM 300-4;
  - Development Procedures: Required documentation manuals and operating instructions; AFDSDCM 300-8, etc…
- SQA Team: testing not only execution software but coding standards; manuals, messages displayed; resources needed; resources named (file names, program names,…)

7. SHORTCOMINGS OF THE TESTING PROCESS

- Likely the part of the development process cut short most frequently!
- Incomplete test plans
  - Parts of application not tested or tested thoroughly!
  - Superficial; boundary conditions…
  - Path testing, branch testing … (coverage measures)
- Failure to document and report detected errors and faults
  - So many levels of testing….we will cover.
- Failure to quickly correct detected faults due to unclear indications that there ‘was’ a fault
- Failure to fix the error due to time constraints
  - Many philosophies here depending on severity of error.

8. USER INTERFACE AND PROCEDURE ERRORS

Remember: to the user, the interface is the entire system.

If the Interface is unsatisfactory, this view will be absolutely conveyed ‘up the line.’

The ‘learnability,’ and utility of the interface.

9. DOCUMENTATION ERRORS

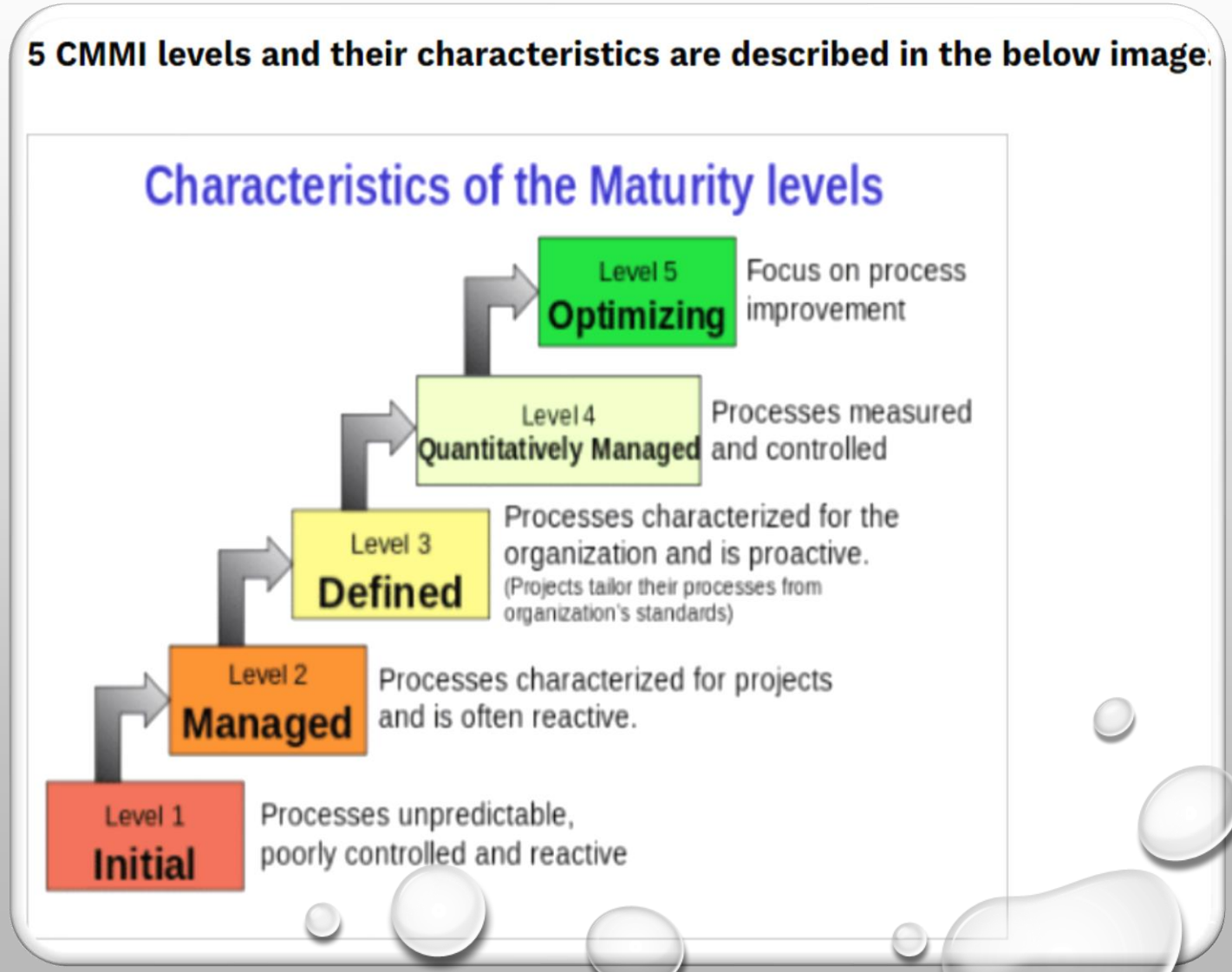
- Errors in the design documents
  - If Docs do not represent the implemented design, this is trouble for subsequent redesign and reuse
- Errors in the documentation in the User Manuals, Operators Manual, other manuals (Installation…)
- Errors in on-line help, if available.
- Listing of non-existing software functions
  - Planned early but dropped; remain in documentation!
- Many error messages are totally meaningless

SOFTWARE QUALITY ASSURANCE STANDARDS

- **Some of the most popular standards:**
  - **ISO 9000:** This standard is based on seven quality management principles which help the organizations to ensure that their products or services are aligned with the customer needs.

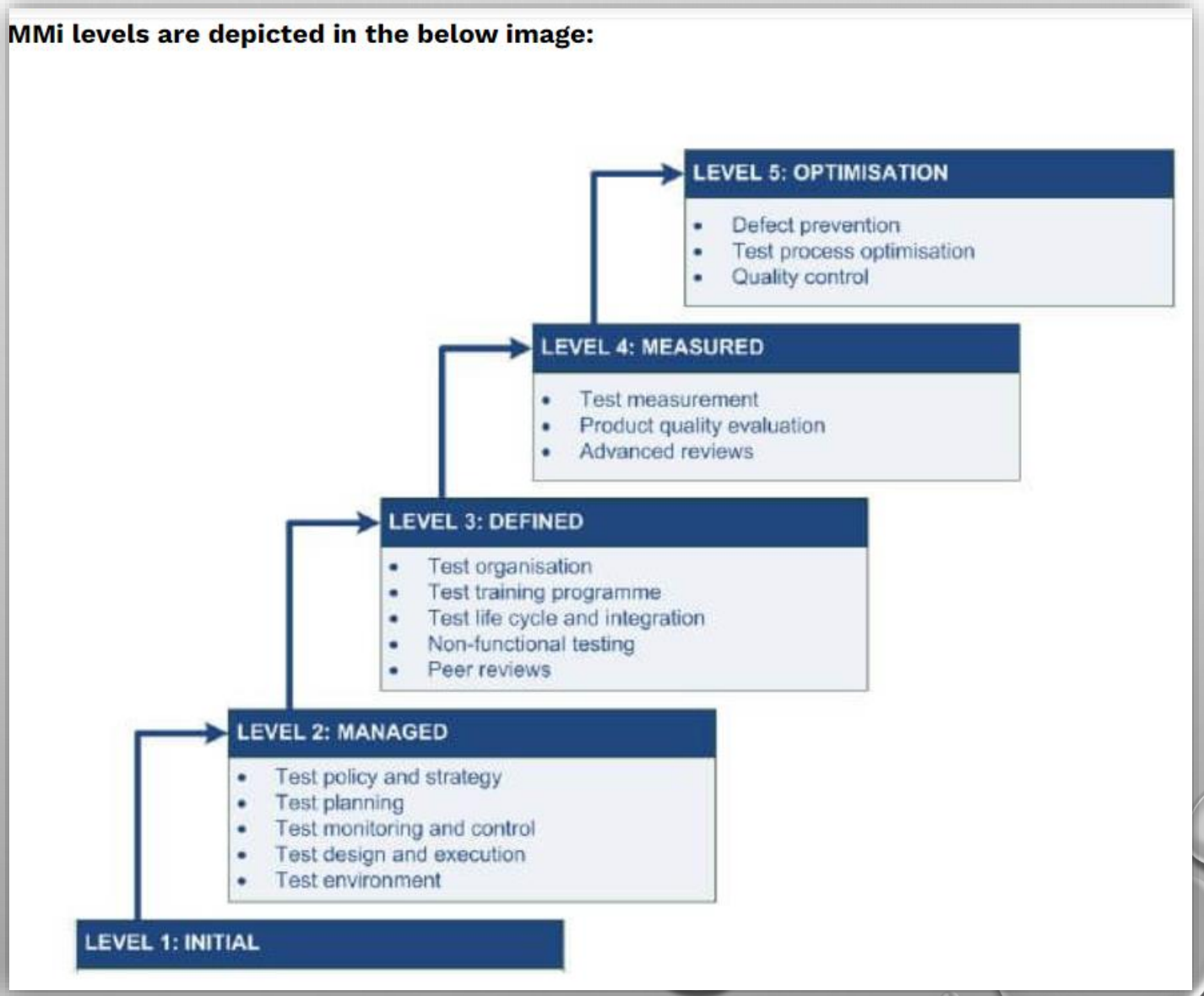
SOFTWARE QUALITY ASSURANCE STANDARDS

- **CMMI level:** CMMI stands for **capability maturity model integration**. This model originated in software engineering. It can be employed to direct process improvement throughout a project, department, or entire organization.



SOFTWARE QUALITY ASSURANCE STANDARDS

- **Test maturity model integration (TMMI): based on CMMI**, this model focuses on maturity levels in software quality management and testing.



ELEMENTS OF SOFTWARE QUALITY ASSURANCE

There are 10 essential elements of SQA which are enlisted below for your reference:

1. Software engineering standards
2. Technical reviews and audits
3. Software testing for quality control
4. Error collection and analysis
5. Change management
6. Educational programs
7. Vendor management
8. Security management
9. Safety
10. Risk management

Summary:

- **SQA** is an umbrella activity that is employed throughout the software lifecycle.
- **Software quality assurance** is very important for your software product or service to succeed in the market and survive up to the customer's expectations.
- There are various activities, standards, and techniques that you need to follow to assure that the deliverable software is of high quality and aligns closely with the business needs.

QUALITY ENGINEERING

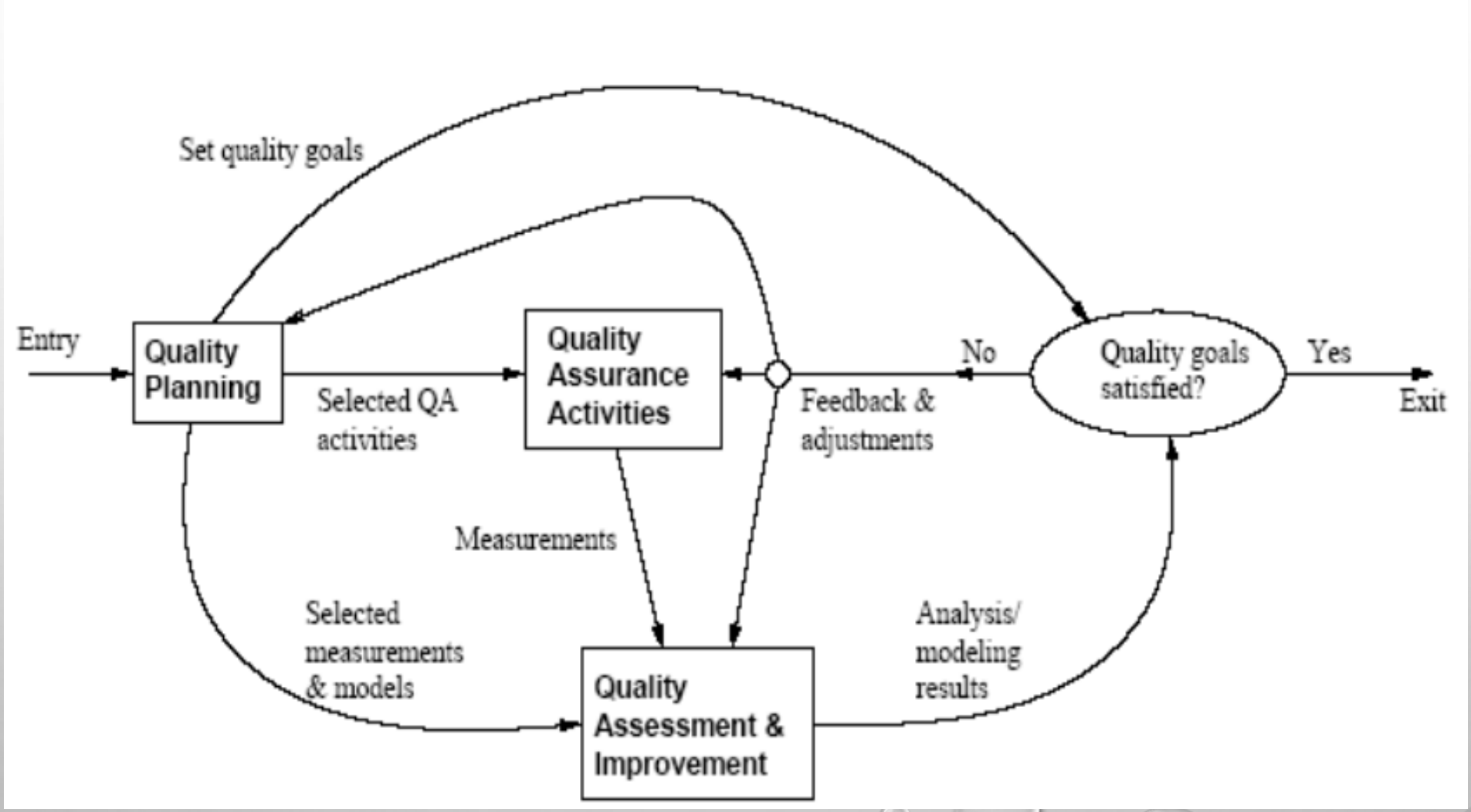
- **Quality engineering focuses on making sure that goods and services are designed, developed, and made to meet or exceed consumers' expectations and requirements**. It includes all the activities related to the analysis of a good's design and development. Quality engineers also make sure that the manufacturer makes the goods according to specifications.
- Quality engineering is the discipline of engineering concerned with the principles and practice of product and service quality assurance and control. In the software development, it is the management, development, operation and maintenance of IT systems and enterprise architectures with a high quality standard. *(Wikipedia)*

QUALITY ENGINEERING VS. QUALITY ASSURANCE

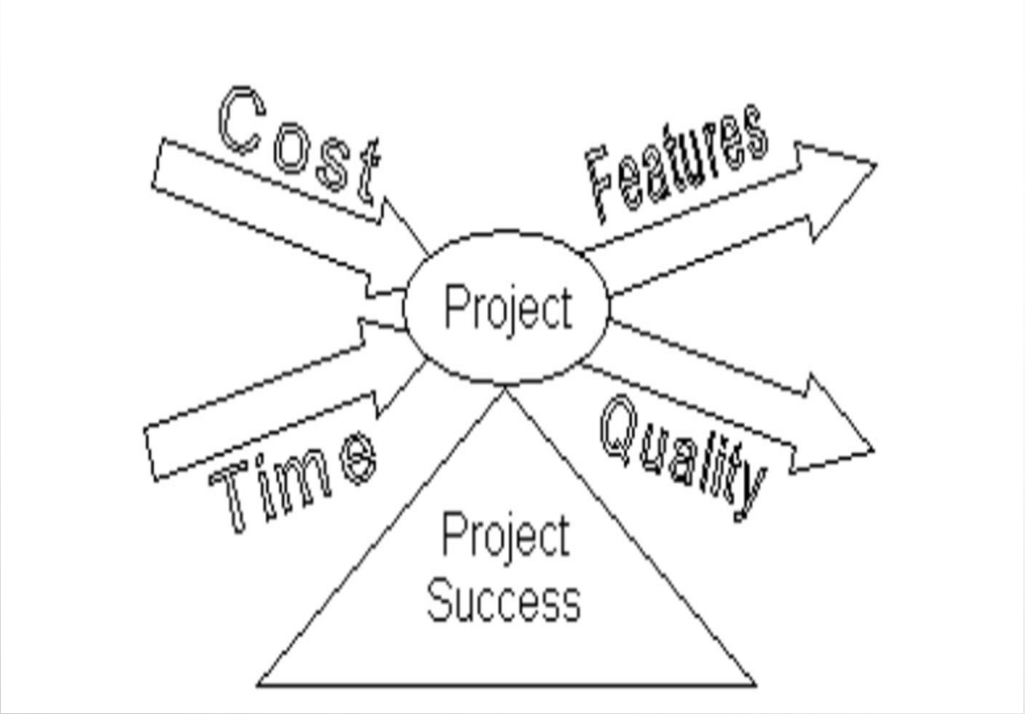
- **Quality assurance** is the overall process of making sure that a manufacturer makes things properly. The manufacturer needs a quality system to make that happen.
- **Quality engineering**, on the other hand, defines that system. It also maintains and improves it.
- **Quality assurance** focuses on maintaining quality in the production process, while **quality engineers** 'engineer' it. In other words, quality engineers devise the system.

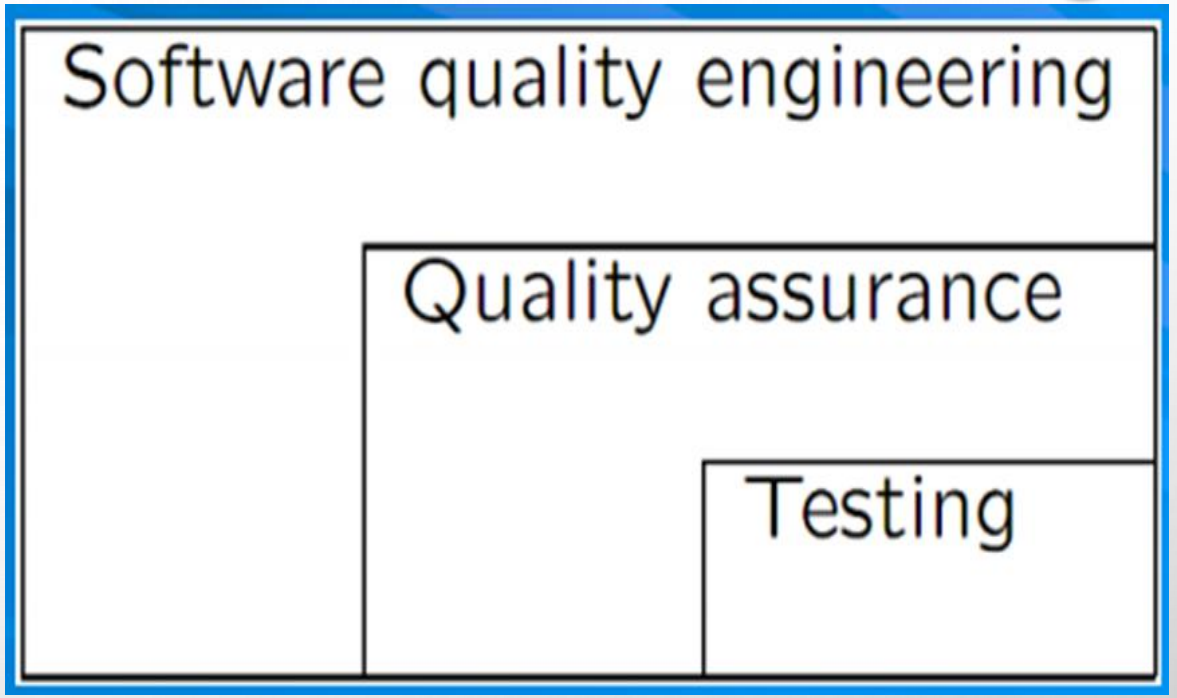
→ **Quality engineering and the 'voice of the customer'**

QUALITY ENGINEERING PROCESS



QUALITY ENGINEERING





1. **QA = TESTING + METHOD + TECHNIQUE + TOOLS → QUALITY PROCESS**

2. **QE = QA + MEASUREMENT + IMPROVEMENT→ QUALITY SYSTEM**

SUMMARY

- “SQA” stands for “software quality assurance”; “SQC” stands for software quality control.
- Software quality assurance is a software quality function which helps in assuring that all the processes, standards, and procedures taking place in a project are appropriate and are being implemented correctly; software quality control is a software quality function which helps in checking that a project follows particular procedures and processes, and the project produces the external and internal products which are required of them
- SQA comprises audits of quality management against a particular standard, and it is the control of processes; SQC includes control of products.