**The software 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

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

**The nine causes of software errors are:**

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

**Software quality is:**

(1) The degree to which a system, component, or process meets specified requirements.

(2) The degree to which a system, component, or process meets customer or user needs or

expectations.

**Software quality factors:**

The need for comprehensive software quality requirements

• Classification of requirements into software quality factors

• Product operation factors

• Product revision factors

• Product transition factors

• Alternative models of software quality factors

• Who is interested in defining quality requirements?

• Software compliance with quality factors

**McCall’s software quality factors model:**

Software quality factors

Product operation factors

Product revision factors

Product transition factors

**Requirement Document**: A comprehensive definition of requirement

**Product operation factors:**

Correctness- Defined in a list of the software system’s required outputs.

**•** Reliability- Deals with failures to provide services.

• Efficiency- Deals with the hardware resources needed to perform all the functions

• Integrity- Deals with the system security, namely, the prevention of the access to unauthorized

persons.

• Usability- Deals with the scope of staff resources needed to train a new employee and to

operate the system.

**Six Classes of SQA Components**:

Pre-project components

• Software project life cycle components

• Infrastructure components for error prevention and improvements

• Management SQA components

• SQA standards, system certification and assessment components

• Organizing for SQA – the human components

**Software project life cycle components:**

Reviews (Formal design reviews & Peer reviews)

• Expert opinions (Formal design reviews)

• Software testing

• Software maintenance components

• Assurance of the quality of external

participants’ work

**Management SQA components:**

Project progress control

• Software quality metrics

• Software quality costs

**Common contract situations:**

Participation in a tender

• Proposal submission according to customer’s RFP

• Receipt of an order from a company’s customer

• Internal request from another department in the organization

**Contract review stages:**

Proposal draft review

Contract draft review

Contract review

**Contract draft review- Objectives:**

No unclarified issues remain in the contract draft

All understandings reached subsequent to the proposal are correctly documented

No “new” changes, additions, or omissions have entered the contract draft