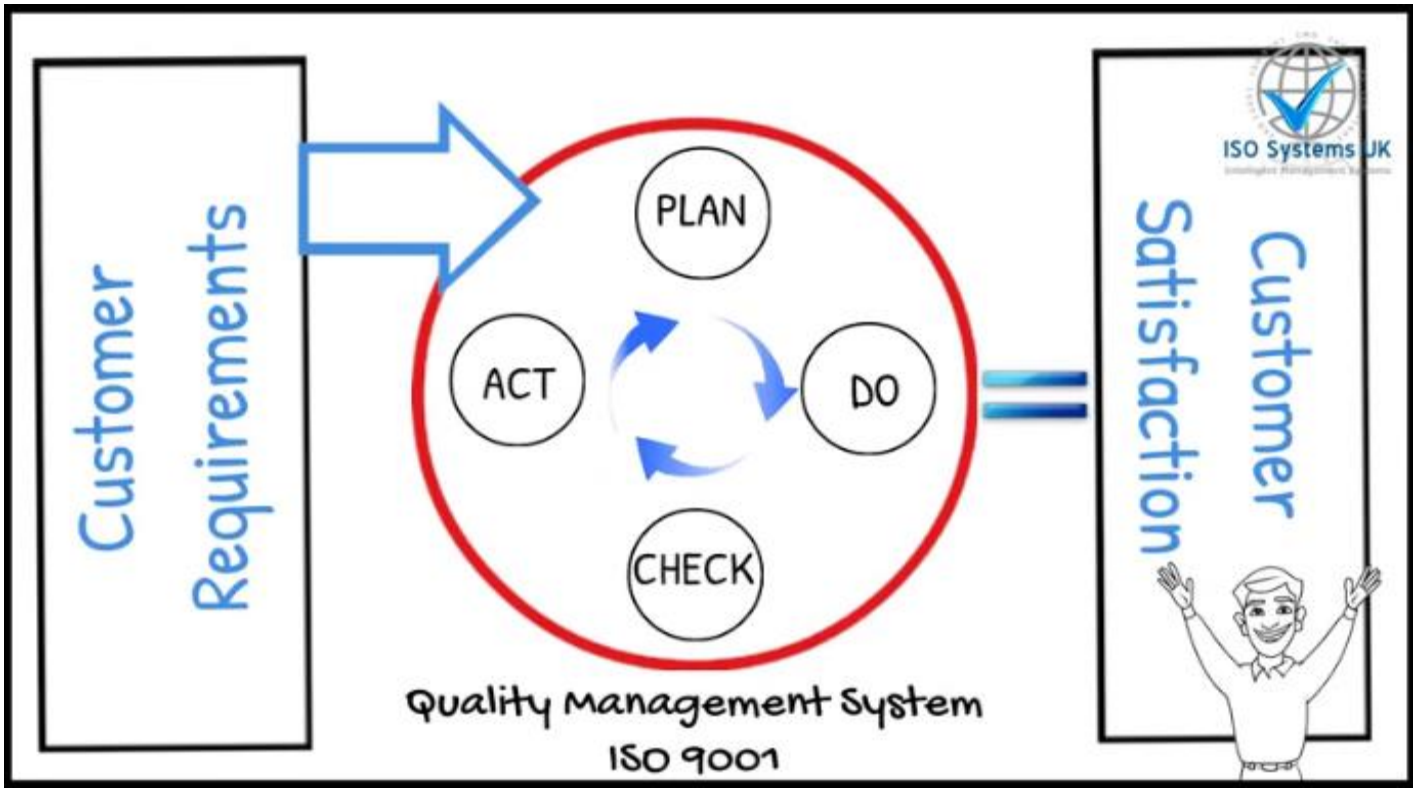
Unit 1

Introduction to Software Quality Assurance

Computer Science and Engineering Department@SoEEC

SQAT(CSE5310)

Software Quality Assurance, QMS



Quality Management System

Software Quality in Detail

Objective:- SQA incorporates all s/w development processes starting from defining requirements to coding until release. Its prime goal is to ensure quality.

- **Quality**
  - Generally excellence of standard or level
- **Software Quality**
  - When it is bug-free, delivered on time and within budget, meets requirements, and expectations, and is maintainable.
- **Some Problems Affecting SQ**
  - Tension between quality attributes
  - Customer vs Developer points of view
  - Requirements and associated problems
  - ambiguity
  - Incompleteness
  - Changes during development, etc

What is Software Quality?

Software Quality is *conformance to*:

- explicitly stated *functional and non-functional requirements*,
- explicitly documented *development standards*,
- *Implicit characteristics* that are expected of all professionally developed software

Software Quality Management

- Aimed to manage the quality of software and its development process

Encapsulate 3 major procedural activities

- **Software Quality Assurance(SQA)**

  Establish organizational procedures and standards for quality

- **Software Quality Plan (SQP)**

  Select applicable procedures and standards for a particular project and modify these as required

- **Software Quality Control (SQC)**

  Ensure that procedures and standards are followed by the software development team.

Software Quality

The degree to which a system, component, or process meets specified requirements

(IEEE Standard Glossary)

Quality is **Conformance** to **Specifications**

(British Défense Industries Quality Assurance Panel)

The degree to which a system, Component, or process meets customer or user needs or expectations.

(IEEE Standard Glossary)

Quality is **Conformance** to **requirements**

(Philip Crosby)

What is Quality?

Quality is Synonymous with customer needs and expectations

(R J Mortiboys)

The totality of features and characteristics of a product or a service that bear on its ability to satisfy stated or implied needs

ISO 8402 definition of QUALITY

Quality is meeting the (Stated) requirements of the customer-now and in the future (Mike Robinson)

1. Functionality
2. Reliability
3. Usability
4. Efficiency
5. Maintainability
6. Portability

ISO 9216 Model: Major Quality Characteristics

Problems with Software Quality

- Software specifications are usually **incomplete** and often **inconsistent**
- Some GAPs between :
  - Customer quality requirements (efficiency, reliability, etc.,)
  - Developer quality requirements (maintainability, reusability, etc.,)
- Some quality requirements are hard to specify in an unambiguous way
  - Directly measurable qualities (eg. Errors / KLOC)
  - Indirectly measurable qualities (e.g. Usability)

KLOC : Thousands (Kilos) of Lines of Code

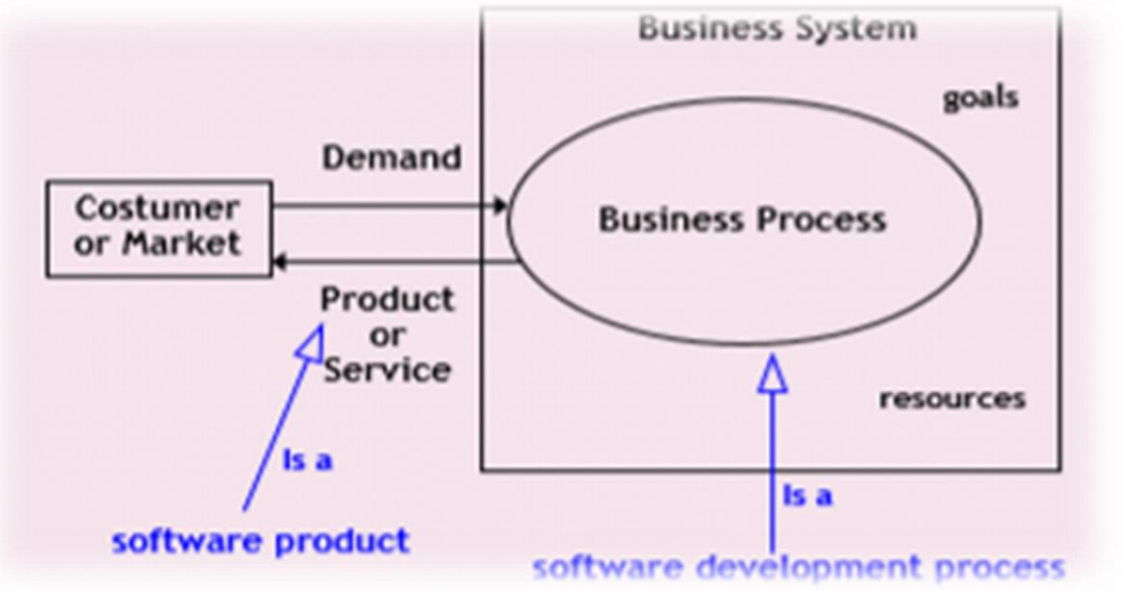
Quality management is not just about reducing defects !

Five views of Software Quality

1. **Transcendental View**: Quality is viewed to be something ideal. In this case, no effort is made to express it using concrete measures.
2. **User View**: In this view, the user is concerned with whether or not the product is fit for use. The product should meet user needs and expectations.
3. **Manufacturing View**: In this view, Quality is seen as confirming requirements.
4. **Product View:** If a product is manufactured with good internal characteristics then it will have good external quality
5. **Value-Based View**: The central view, in this case, is how much a customer is willing to pay for a certain level of quality

Software Quality Management

- Managing the quality of the software process and products
- Product and process



S/W Process and Product Quality

- The quality of a developed product **is influenced** by the quality of the production process.
- Particularly important in software development as some **Product Quality attributes are hard to assess**
- However, there is a very complex and poorly understood between **software processes and product quality**

What is Software Product ?

Software Product = Computer Programs (Source code and Executables) + associated documentation

- Software products may be
  1. **Custom**: Developed for a particular customer, according to its specification
  2. **Generic**: { " Package") - Developed for a general market. To be sold to a range of different customers.
- Types of software products

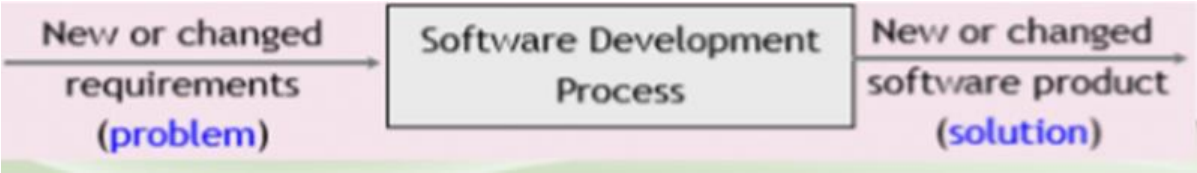
  Business Support Software: Includes software engineering tools in the software engineering business

  Personal productivity software: Spreadsheets, word processing tools,

  Embedded Software : real-time systems, networked systems ,and so on,

Software Development Process

- A set of activities whose goal is the **development or evolution of a software product**
  - To be followed/instantiated in individual software development projects
- It's the main business process in a **software development business**
- **Generic activities in all software processes are :**
  - **Specification** – What the system should do and its development constraints
  - **Development**: production of the software system
  - **Validation**: checking that the software is what the customer wants
  - **Evolution**: Changing the software in response to changing demands.



S/W Product Quality

- The software product should deliver the required functionality (**functional requirements**) with the required quality attributes (**non-functional requirements**)
- Quality attributes are **frequently conflicting** and increase development costs, so there is a need for weighting and balancing
- Software engineering is concerned with the **cost-effective development of good software**

A product should meet its specification

Product Quality Factors

Product Quality Factors

Principal Product Quality Factors (2)

Process Quality :

- **A good process** is usually required to **produce a good product**.
- For manufactured goods, the process is the **principal quality determinant,**
- For design-based activity other factors are also involved especially the **capabilities of the designers**
- For large projects with average capabilities, the **development process determines product quality**

People quality:

- **For small projects**, the **capabilities of the developers** are the main determinant
- **Corollary**: Do you need lower-quality people and higher-quality processes in larger projects?

Product Quality Factors

Principal Product Quality Factors (3)

Development Technology

is particularly significant for small projects

Budget and Schedule

Budget influences the quality

In all projects, if an unrealistic schedule is improved then product quality will be suffered

Product quality Attributes (1)

- Efficiency
- Usability
- Dependability
  - Reliability
  - Availability
  - Security
  - Safety
- Maintainability
- Resilience
- Robustness
- Testability
- Understand ability
- Adaptability
- Modularity
- Simplicity
- Portability
- Reusability
- Leamability

Product Quality Attributes

Product Quality Attributes (2)

- Attributes of good software (beyond delivering the required functionality) :
  - **Efficiency** :
    - software should not make wasteful use of system resources (disk and memory space, CPU time, etc.,) and should present appropriate response times.
  - **Usability** (ease of Use)
    - Software must be usable by the users for which it was designed
  - **Dependability** (reliability, availability, security, safety …)
    - Software must be trustworthy
  - **Maintainability** (ease of maintenance)
    - Software must evolve to meet changing needs
    - Software costs more to maintain than it does to develop for systems with a long life, maintenance costs may be several times development costs.

Product Quality Issues

Does usage and time cause degradation of a software product quality?

By definition, should not, but ……

- A program running continuously for a long period of time (without shutting down) may work increasingly slower or even crash
  - E.g because of memory leaks or memory fragmentation
  - Fortunately, original quality is restored by shutting down and restarting
- **Performance** decreases with the number of concurrent users and the size of the data
  - May req. hardware upgrade and, consequently software upgrade (good 4 business)
- **Maintainability decreases with time**
  - May req. preventive maintenance (migration to new technologies, etc)
- **Software becomes obsolete very quickly**
  - Because of the fast evolution of technology, requirements or knowledge
  - Sometimes software is used for a longer time than expected
  - Requires continuous innovation and evolution

Process based Quality

- Straight forward link between process and product in manufactured goods
- More complex for software because :
  - The individual skills and experience is particularly important in software development
  - External factors such as the novelty of an application or the need for an accelerated development schedule may impair product quality
- Care must be taken not to impose inappropriate process standards.

Flow of Process based Quality

Process Quality attributes

Process Characteristic	Description
Understandability	To what extent is the process explicitly defined and how easy is it to understand the process definition ?
Viability	Do the process activities culminate in clear results so that the progress of the process is extremely visible?
Supportability	To what extent can the process activities be supported by CASE tools ?
Acceptability	Is the defined process acceptable to and usable by the engineers responsible for producing the software product ?

Process Quality attributes

Process Characteristics	Description
Reliability	Is the process designed in such a way that process errors are avoided or trapped before they result in product errors ?
Robustness	Can the process continue in spite of unexpected problems ?
Maintainability	Can the process evolve to reflect changing organisational requirements or identified process improvements ?
Rapidity	How fast can the process of delivering a system from a given specification be completed ?

Quality factor Model

McCall's Quality factor model

Eleven (11) quality factors grouped into 3 Categories

Product Operation Factors	Product Revision Factors	Product Transition Factors
1.Correctness	6.Mainability	9.Portability
2.Efficiency	7.Flexibility	10.Reusability
3.Integrity	8.Testability	11.Interoperability
4. Reliability		
5. Usability		

The quality compromise

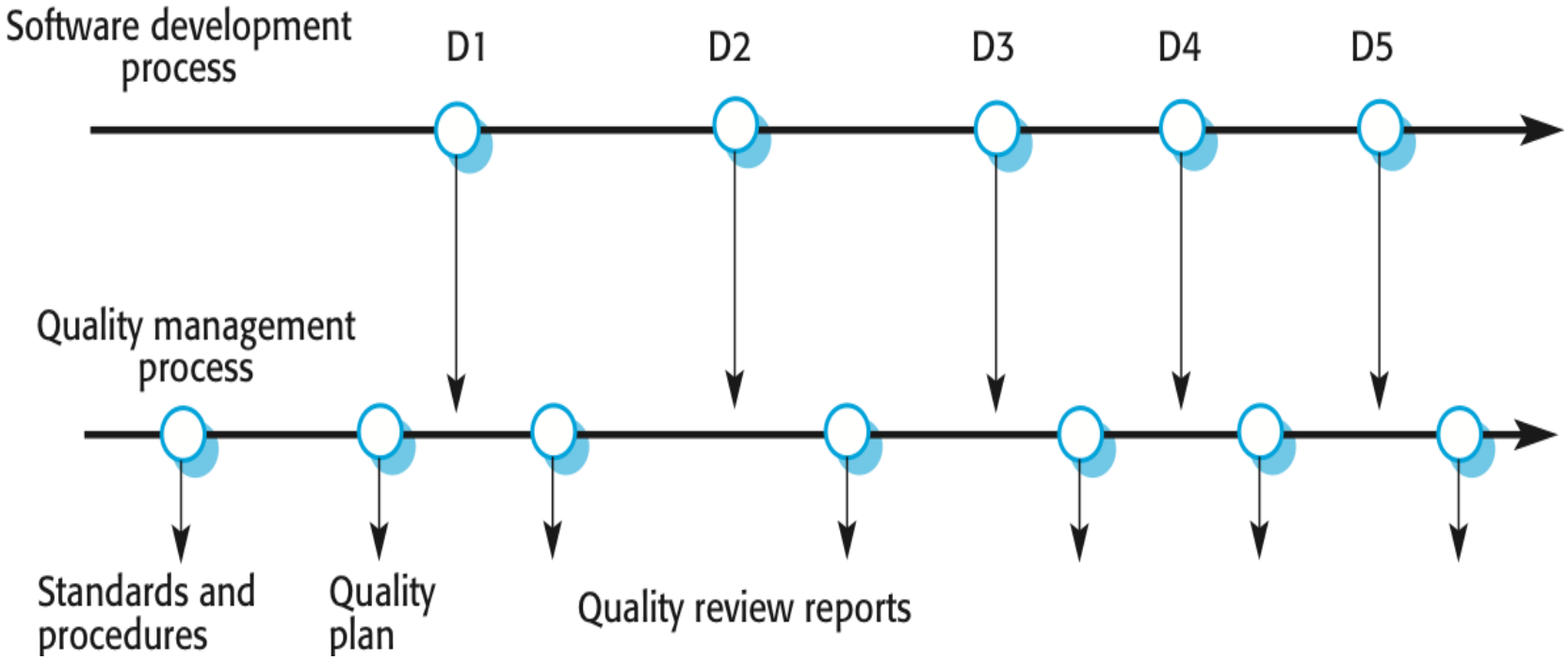
We cannot wait for

- Specifications to improve before paying attention to quality – and
- Procedures must be put into place to improve quality in spite of imperfect specification

Hence we compromise …….

Quality management is therefore not just concerned with reducing defects but also with other product qualities.

Quality Management & S/W Development



Quality Concept Issues

- **Quality of design:** refers to the characteristics that designers specify for an item.
- **Quality of conformance:** is the degree to which the design specifications are followed during manufacturing.
- **Quality control**: is the series of inspections, reviews, and tests used throughout the development cycle to ensure that each work product meets the requirements placed upon it.
- **Quality policy:** refers to the basic aims and objectives of an organization regarding quality as stipulated by the management.
- **Quality assurance:** consists of the auditing and reporting function of management.

Quality Concept issues cont.....

- **Cost of quality:** includes all costs incurred in the pursuit of quality or in performing quality-related activities such as appraisal costs, failure costs, and external failure costs.
- **Quality planning:** is the process of assessing the requirements of the procedure and of the product and the context in which these must be observed.
- **Quality testing:** is an assessment of the extent to which a test object meets given requirements
- **Quality assurance plan:** is the central aid for planning and checking quality assurance.
- **Quality assurance system:** is the organizational structure, responsibilities, procedures, process, and resources for implementing quality management.

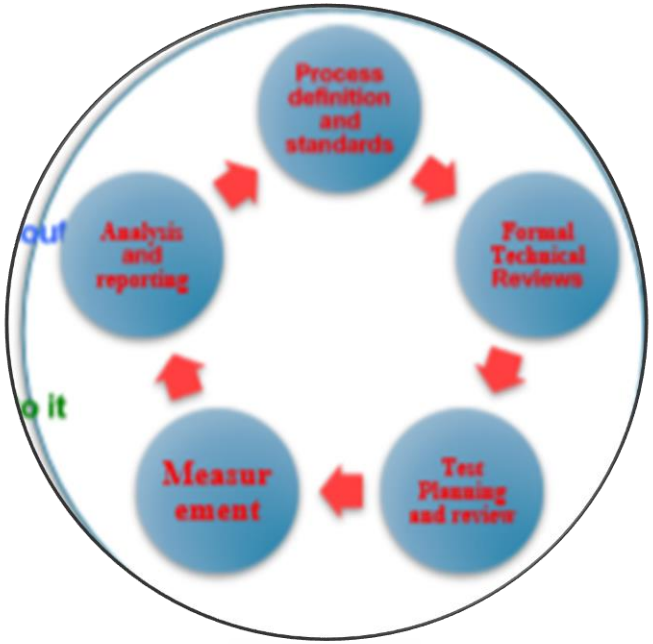
SQA Standards and S/W Quality Assurance

SQA-IEE Definition

- **Software Quality Assurance is** A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.
- SQA is an umbrella activity that is applied throughout the software process.

SQA Encompasses

- A quality management approach
- Effective software engineering in technology
- Formal technical reviews that are applied throughout the software process.
- A multi-tiered testing strategy
- Control of software documentation and changes to it
- A procedure to assure compliance with software development standards.
- Measurement and reporting techniques.



SQA Activities in S/W Development

The Objective of SQA activities in Software Development

- Assuring an acceptable level of confidence that the software will conform to functional technical requirements.
- Assuring an acceptable level of confidence that the software will conform to managerial scheduling and budgetary requirements.
- Initiation and management of activities for the improvement and greater efficiency of software development and SQA activities.

Software Quality Assurance and Software Engineering.

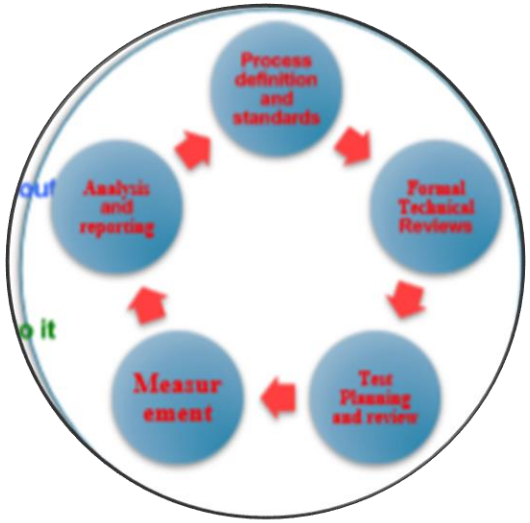
- The characteristics of software engineering especially the systematic disciplined and quantitative approach at its core. Make the software engineering environment a good infrastructure for achieving SQA objectives.
- The **methodologies and tools that are applied by software** engineering determine, to a considerable extent the level of quality to be expected from the software process and the maintenance services.

SQA Activities Graph

SQA activities are performed on every software project Use of metrics is an important part of developing a strategy to improve the quality of both software processes and work products.

S/W Quality assurance group

- **The SQA group** is a team of people with the necessary training and skills to ensure that
- All necessary actions are taken during the development process.
- So that the resulting software conforms to established technical requirements.
- SQA group must look at software from the customer's perspective assessing its technical merits.



SQA Group Activities

Prepare SQA plan for the project

- Participate in the development of the project's software process description.
- Review software engineering activities to verify compliance with the defined software process.
- Audit designated software work products to verify compliance with those defined as part of the software process.
- Ensure that any deviations in software or work products are documented and handled according to a documented procedure.
- Record any evidence of noncompliance and reports them to management.

Components of the S/w quality Assurance

- Most software quality assurance activities can be categorized into software testing that is verification and validation, software configuration management, and quality control.
- But the success of a software quality assurance program also depends on a coherent collection of standards , practices, conventions, and specifications.

Components of the S/W Quality Assurance

Can be divided into six classes

1. Pre-project components
2. Components of project life cycle activities assessment
3. Components of infrastructure error prevention and improvement
4. Components of software quality management
5. Components of standardization, certification, and SQA system assessment.
6. Organizing for SQA the human components

Pre-Project Components: to assure that

- The project commitments have been adequately defined considering the resources required , the schedule and budget.
- The development and quality plans have been correctly determine

Components of the S/W Quality Assurance

1. **Pre-Project software quality components:**

The SQA components of this class are meant to improve the preparatory steps taken prior to initiating work on the project itself , the components are

- Contact review
- Development and Quality plans

2. **Components of project life cycle activities assessment**

The project life cycle is composed of two stages

- The development life cycle state and
- The operation maintenance stage

- The **first stage** components detect design and programming components are divided into four subclasses :

  Reviews, Expert, Opinions, software testing

- The **second stage** components include specialized maintenance components as well as developing life cycle components, which are applied mainly for functionality-improving maintenance tasks

Components of the software quality assurance

3. Components of infrastructure error prevention and improvement

The main objective of these components, which are applied throughout the operation, are

To eliminate, or at least reduce the rate of errors, based on the organization's accumulated SQA experience

4. Components of software quality management

The goal of this class component is to

Control development and maintenance activities and the introduction of early managerial support actions that mainly prevent and minimize schedule and budget failures and their outcomes.

Components of the S/w Quality Assurance

5. Components of standardization , certification and SQA system assessment

The main objective of this class are utilization of international professional knowledge

- Improvement of coordination of the organizational quality systems with other organizations.
- Assessment of the achievements of quality systems according to a common scale

The various standards may be classified into two main groups

- **Quality management standards and**
- **Project management standards.**

Components of the s/w Quality Assurance

6. Organizing the SQA-the human components

The SQA organizational base includes managers, testing personnel, the SQA unit, and practitioners interested in software quality.

The main objectives of these members are to

- Initiate and support the implementation of SQA components
- Detect deviations from SQA procedures and methodology and suggest improvement

IEEE standards for SQA PLAN

- An important part of achieving quality is to plan for quality
- A software quality assurance plan is not merely another name for a test plan, though test plans are included in an SQA plan.
- The IEEE standards association has developed a standard (std 730-1989) for software quality assurance plans.

The following is part of the sections specified in IEEE std 730-1989

- **Purpose**: This section shall list the software covered and the portions of the software life cycle covered.
- **Reference Documents**: This section shall list all the documents referenced in the plan.

IEEE STANDARDS for SQA PLAN

3. Management

3.1 **Organization**: this selection describes the structure of an organization and its responsibilities, and usually includes an organization chart.

3.2 **Tasks**: this section lists all of the tasks to be performed, the relationship between tasks and checkpoints, and the sequence of the tasks

3.3 **Responsibilities**: this section shall list the responsibilities of each organization unit.

4. Documentation :

4.1 **Purpose** : required document and state how documents will be evaluated

4.2 **Minimum documents**: this section shall describe the minimum required documentation , usually including the following :

SRS – Software Requirements Specification

SDD- Software Design Description

SVVP-Software Verification and Validation Plan

SVVR-Software Verification and Validation Report

User documentation & SCMP – SW configuration management plan

IEEE STANDARDS for SQA PLAN

5. Standards, Practices, Conventions, and Metrics

This section shall identify the S , P , C and M to be applied and how compliance is to be monitored and assured.

The minimal contents include documentation standards , logic structure standards, coding standards, testing standards, selected SQA product, and process metrics.

6. Reviews and Audits : This section shall define what reviews / audits will be done , how they will be accomplished , and what further actions are reuired.

7. Tests: this section shall include all tests that are not included in SVVP (Software Verification and Validation Plan)

8. Problem reporting : This section shall define practices and procedures for reporting , tracking and resolving problems , including organizational responsibilities.

IEEE Standards for SQA PLAN

9. Tools, Techniques, and Methodologies: This section shall identify the special software tools , techniques, and methodologies and describe their use.

10.Code control : this section shall define the methods and facilities to maintain controlled versions of the software .

11. Medica control : this section shall define the methods and facilities to identify , store, and protect the physical media.

12. Supplier control (for outsourcing) : this section shall state provisions for assuring that software provided by suppliers meets standards.

13. Records : this section shall identify documentation to be retained and methods to collection , maintain, and safeguard the documentation.

14.Training: This section shall identify necessary training activities

15. Risk Management : This section shall specify methods and procedures for risk management.

End of WEEK-1

Unit-1 Week-2 Quality Standards

Quality Standards

Process

ISO 9000 for Process mapping

SEI CMM for Software process improvement

SEI CMM for integrated software process

People :

PCMM for people involvement in quality initiatives (competencies & Talent)

People & Process (mix and match)

Six-Sigma initiative for continuous improvement an existing processes

Other Quality methodologies

Bootstrap

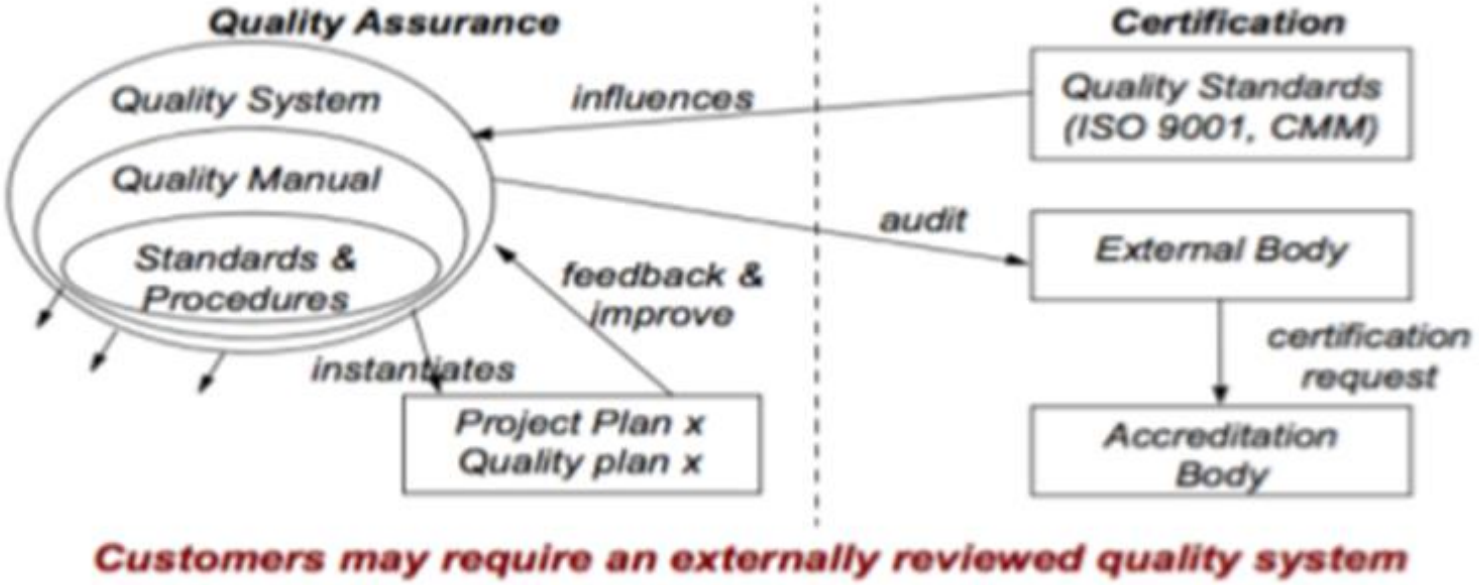
TRILLUM

Quality improvement paradigm

Kaizen

Quality System flow

A quality plan should be an instance of an organization's quality system



ISO 9000

ISO 9000 is an international set of standards for a quality management application to a range of organizations from manufacturing to service industries.

ISO 9001 is a generic model for quality processes, applicable to organizations whose business processes range all the way from design and development to production, installation, and servicing.

-ISO 9001 must be instantiated for each organization
-ISO 90003 interpret: ISO 9001 for the software developer

ISO =international organization for Standardization
-ISO main site: http://www.iso .ch/
-ISO 9000 main site: http://www.tc176.org/

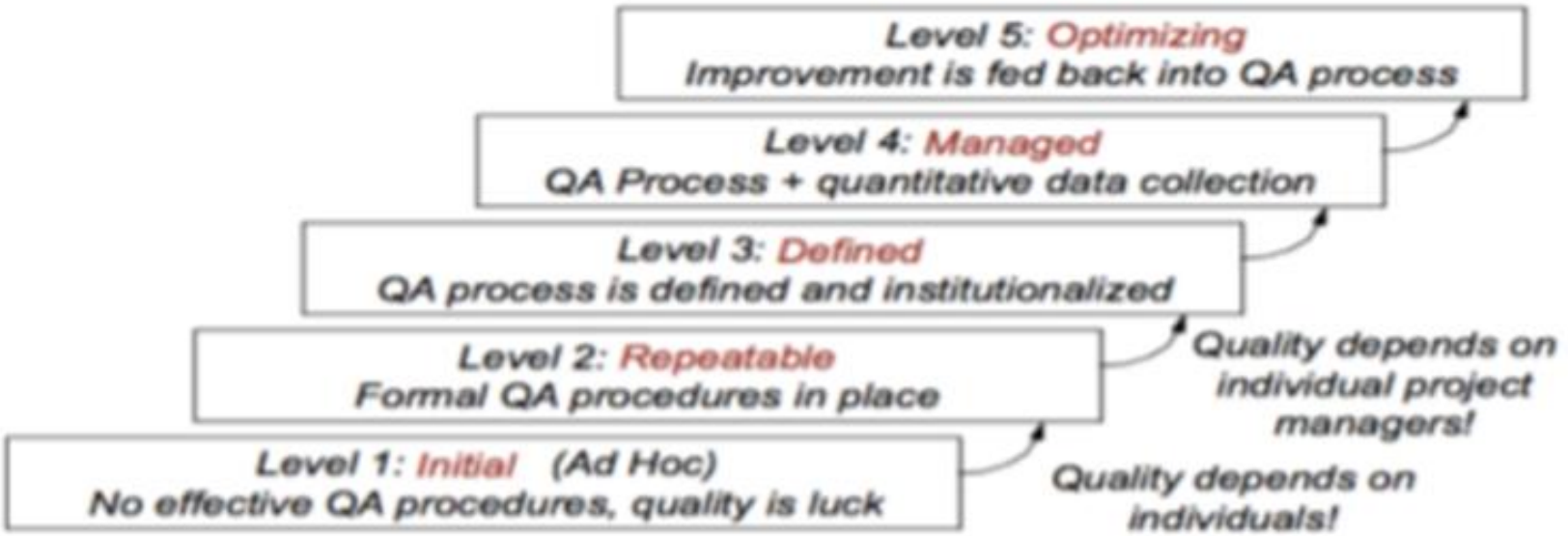
Capability Maturity Model (CMM)

The SEI process maturity model classifies

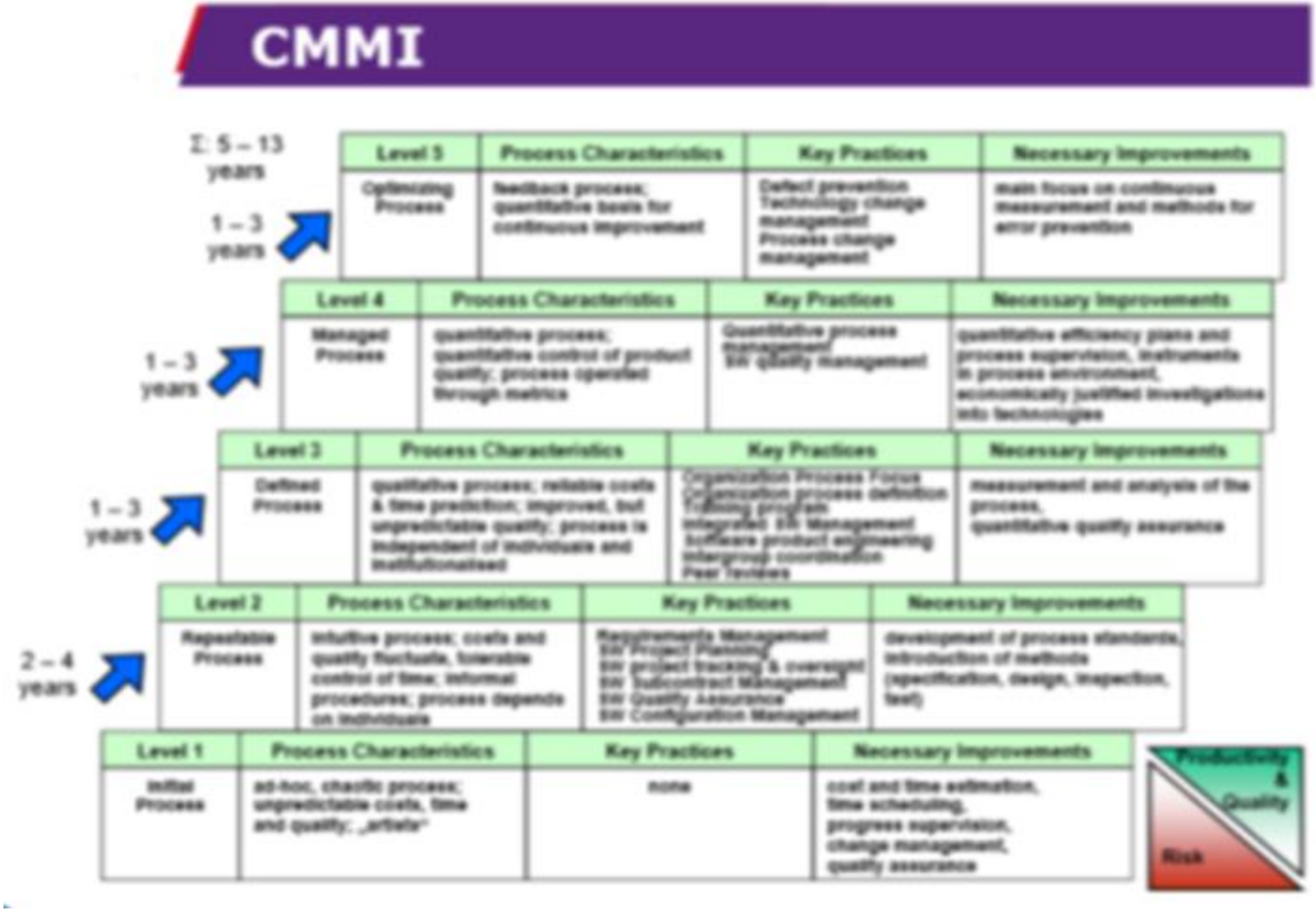
- **how well contractors manage software processes**

CMM Links -

http://www.sei.cmu.edu/cmm/cmm.sum.html
http://www2.umassd.edu/SWPI/processframework/cmm/cmm.html



CMMI


CMMI
Σ: 5 – 13 years
1 – 3 years
Level 5
Process Characteristics
Key Practices
Necessary Improvements
Optimizing Process
feedback process; quantitative basis for continuous improvement
Defect prevention
Technology change management
Process change management
main focus on continuous measurement and methods for error prevention
1 – 3 years
Level 4
Process Characteristics
Key Practices
Necessary Improvements
Managed Process
quantitative process; quantitative control of product quality; process operated through metrics
Quantitative process management
SW quality management
quantitative efficiency plans and process supervision, instruments in process environment, economically justified investigations into technologies
1 – 3 years
Level 3
Process Characteristics
Key Practices
Necessary Improvements
Defined Process
qualitative process; reliable costs & time prediction; improved, but unpredictable quality; process is independent of individuals and institutionalised
Organization Process Focus
Organization process definition
Training program
Integrated SW Management
Software product engineering
Intergroup coordination
Peer Reviews
measurement and analysis of the process,
quantitative quality assurance
2 – 4 years
Level 2
Process Characteristics
Key Practices
Necessary Improvements
Repeatable Process
intuitive process; costs and quality fluctuate, tolerable control of time; informal procedures; process depends on individuals
Requirements Management
SW Project Planning
SW project tracking & oversight
SW Subcontract Management
SW Quality Assurance
SW Configuration Management
development of process standards, introduction of methods (specification, design, inspection, test)
Level 1
Process Characteristics
Key Practices
Necessary Improvements
Initial Process
ad-hoc, chaotic process; unpredictable costs, time and quality; „artistic"
none
cost and time estimation, time scheduling, progress supervision, change management, quality assurance
Productivity & Quality
Risk

PCMM and SIX-SIGMA

PCMM:

- Strong emphasis on learning and professional development , both at the individual and organizational levels
- Culture that empowers individuals and promotes a participatory environment
- Effective professional mentoring programs
- Emphasis on continuous improvement

SIX-SIGMA

- It is a customer-based approach realizing that defects are expensive.
- It is a structured application of tools & techniques applied on a project basis to achieve sustainable results.
- Six Sigma processes will produce less than 3.4 defects/mistakes for a million opportunities i.e. 99.99% perfection [called five 9's]

REVIEWS

IEEE definition Review process :

- A process or meeting during which a work product or set of products is presented to project personnel, managers, users, customers, or other interested parties for comment or approval.

Methodologies for reviewing documents :

- Reviews acquire special importance in the SQA process because
- They provide early direction and prevention passing of design and analysis errors downstream to stages where error detection and correction are much more complicated and costly.
- The methodologies for reviewing :
  1. Formal design review
  2. Peer reviews (inspections and walkthroughs)
  3. expert opinions

Standards for SW reviews are the subject of IEEE aid 1028 (IEEE, 1997)

REVIEWS Objectives (Direct Objectives)

1. To direct analysis & design errors as well as subjects where corrections, changes and competitors are required with respect to the original specifications and approved changes.
2. To identify new risks likely to affect the completion of the project
3. To locate deviations from templates and style procedure and conventions
4. To approve the analysis or design product approval allows the team to continue to the next development phase.

Reviews objectives (Indirect objectives)

- To provide an informal meeting place for the exchange of professional knowledge about development methods, tools, and techniques.
- To record analysis and design errors that will serve as a basis for future corrective actions.

DESIGN REVIEWS

Formal Design reviews (DR's)

Formal Design reviews also called

- Design Reviews (DR's)
- Formal Technical Reviews (FTR)

Without this approval :

- The development team cannot continue to the next phase of SW development project
- Formal design review can be conducted at any development milestone requiring the completion of an analysis or design document whether that document is a requirement specification or an installation plan.

The formal design review will focus on :

- The participants
- The prior preparations
- The DR session
- The recommended post – DR activities

A List of common formal design reviews

DPR — development plan review
SRSR— Software requirement specification review
PDR — Preliminary design review
DDR — Detail design review
DBDR — Data base design review
TPR —Test plan review
STPR — Software test procedure review
VDR —Version description review
OMR — operator manual review
SMR — Support manual review
TRR —Test readiness review
PRR — Product release review
IPR — Installation Plan review

The participants in a DR

All DRs are conducted by

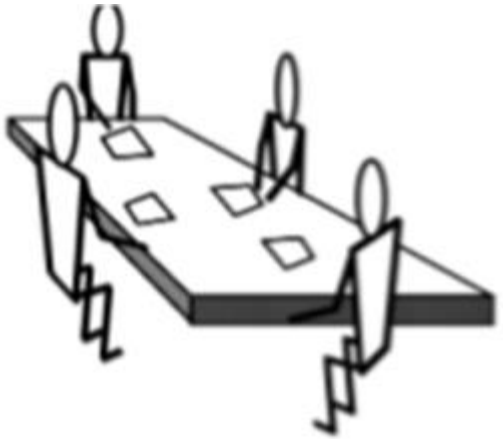
- A Review leader
- A Review Team

The review leader characteristics

- Knowledge & experience in the development of projects of type reviewed
- Seniority at a level similar to if not higher than that of the project leader
- A good relationship with the project leader and his team.
- A position external to the project team
- Small dev. Departments and software houses typically have difficulties, finding an appropriate candidate to lead the review team. One possible solution to this is the appointment of an external consultant.

Conducting the Reviews

1. Be prepared – evaluate the product before the review
2. Review the product, not the producer
3. Keep your tone mild, ask questions instead of making accusations
4. Stick to the review agenda
5. Raise issues, don't resolve them, limit debate and rebuttal, record issues or further discussion offline
6. Avoid discussions of style –stick to technical correctness, and enunciate problem areas, but don't attempt to solve every problem noted.
7. Schedule reviews as the project taks
8. Record and report all review results.



Design Review Session Plan – The DR Session

The agenda is the issue (a typical DR session agenda)

1. A short presentation of the design document
2. Comments made by members of the review team
3. Verification and validation in which each of the comments is discussed to determine the required actions (corrections, changes, and addition) that the project team has to perform.
4. Decisions about the design product (document), which determines the project progress.

- These decisions take the following three decision forms:

Full approval: enables immediate continuation to the next phase. It may be accompanied by demands for some minor corrections to be performed by the project plan.

Partial approval: approval of immediate continuation to the next phase for some parts of the project, with major action items demanded for the remainder of the project.

Denial of Approval: demands to repeat of the DR

Review Guidelines

- **Take written notes**: The recorder should collect the critical concerns so that wording and prioritization can be assessed by other reviewers as information is recorded.
- **Limit the number of participants** and insist upon advance preparation.
- **Develop a checklist** for each type of work product that is likely to be reviewed.
- **Allocate resources and time schedule** for formal technical reviews
- **Conduct meaningful** training for all reviewers.
- **Review your early reviews** –debriefing cn be beneficial in uncovering problems with the review process itself.
- **Avoid discussions** of style-stick to technical correctness.
- **Schedule reviews** as project tasks.
- **Record and report** all review results.

Review Effectiveness

Ensuring Quality through Reviews

Peer Reviews

- Difference between formal design reviews and peer reviews is really in both their participants and authority.

Reviews	Peer Reviews
Most participants hold superior positions to the project leaders and customer representatives	Most participants are equals , i.e members of his/her department and other units
Can be design reviews , formal technical reviews	Can be inspections, walkthroughs

Inspection	Walkthrough
Inspections emphasize the objective of corrective action more formal.	Walkthroughs are limited to comments on the document reviewed, informal
Inspections are a method to reduce validity and tighten process control.	

Participants of peer reviews

Inspection

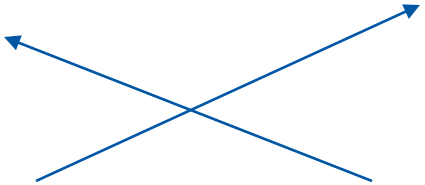
- Review Leader
- The author
- Specialized

Professionals

- Designer
- Coder or Implementor
- Tester

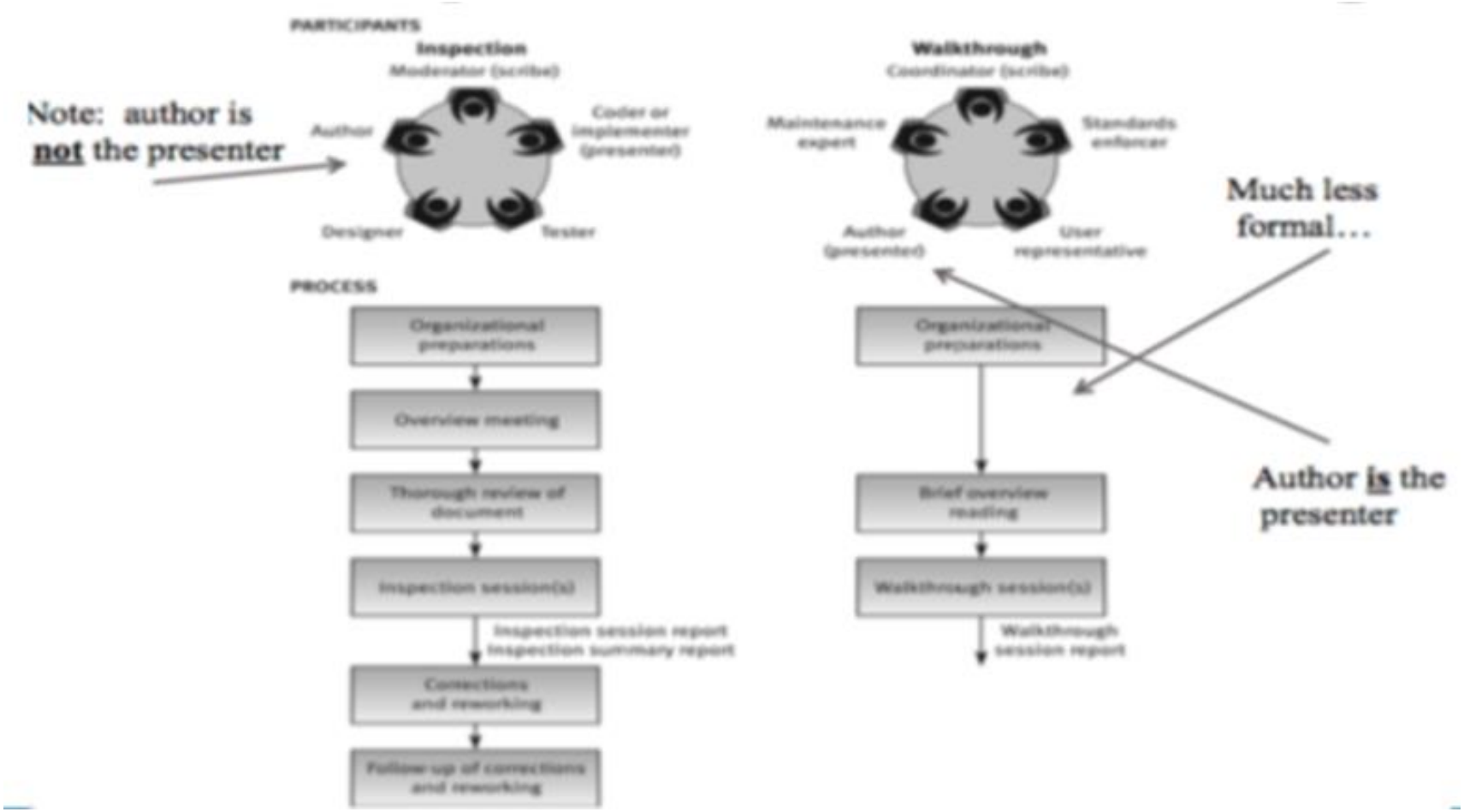
- **Walkthrough**

- Review leader
- The author
- Specialized
- Professionals
  - Standards enforcer
  - Maintenance expert
  - Under representative



M.E Fagan of IBM first defined inspectionsin 1976

Inspection and walkthrough



Difference between Inspection and Walkthrough

Walkthrough :

Walkthrough is a method of conducting informal group/individual review. In a walkthrough, the author describes and explain work product in a informal meeting to his peers or supervisor to get feedback. Here, validity of the proposed solution for work product is checked.

It is cheaper to make changes when design is on the paper rather than at time of conversion. Walkthrough is a static method of quality assurance. Walkthroughs are informal meetings but with purpose.

Inspection :

An inspection is defined as formal, rigorous, in depth group review designed to identify problems as close to their point of origin as possible. Inspections improve reliability, availability, and maintainability of software product.

Anything readable that is produced during the software development can be inspected. Inspections can be combined with structured, systematic testing to provide a powerful tool for creating defect-free programs.

Inspection activity follows a specified process and participants play well-defined roles.

An inspection team consists of three to eight members who plays roles of moderator, author, reader, recorder and inspector.

What can be inspected ?

- **Inspections can be held a various points in development process**
- Fagan recommended inspections on
  1. Detailed design
  2. Clearly compiled code
  3. Completion of unit test

Steps of inspection

- Planning
  a) Entry criteria for inspection type
  b) Moderator is selected-usually a peer or technical leader
  c) Management is encouraged not to look at individual inspection results
  d) Moderator verifies that the product meets entry criteria and schedules future steps.

Steps of Inspection

2. Overview

a) Presentation to inspectors
b) Purpose is educational only
c) Data collected is the author Preparation & presentation time

3. Preparation

a) Individual activity
b) Author collects all material
c) Inspectors study the material and complete the inspection log
d) Defects are noted at this step but not collected.

Agenda Includes

a) Introduction
b) Establishing readiness
c) Examining material and recording defects
d) Review defects
e) Determine disposition

Rework

a) Performed by the author

Follow-up

The moderator verifies that corrections are made & completes inspection management & defect summary report.

Inspection Roles

- **Author** – developer of work product
- **Moderator** – an inspector responsible for organizing and reporting on inspection
- **Reader** – an inspector who guides the examination of the product
- **Recorder** – an inspector who enters all the defects found on the defect list
- **Inspector** – Member of the inspection team, often chosen to represent specific roles – designer , tester, technical writer, SQA etc

General Opinions

- When done correctly, walkthroughs and inspections are **valuable defect-finding tools**
- When **not supported** by management or bought into by development personnel, they become **busy work** for developers.
- It is important for developers to not take criticism personally
- It is equally important for inspectors to look for defects and not criticize because developers didn't code exactly the way they would.

Sample review checklists

Sample Review checklists (1)

Software project planning

1. Is software scope unambiguously defined and bounded?
2. Are resources adequate for scope?
3. Have risks in all important categories been defined?
4. Are tasks properly defined and sequenced?
5. Is the basis for cost estimation reasonable?
6. Have historical productivity and quality data been used?
7. Is the schedule consistent?

Requirement Analysis :

- Is the information domain analysis complete, consistent and accurate?
- Does the data model properly reflect data object attributes and relationships?
- Are all requirements traceable to the system level?
- Has prototyping been conducted for the user? Customer?
- Are requirements consistent with schedule, resources, and budget?

Sample Review Checklists (II)

Design:

- Has modularity been achieved?
- Are interfaces defined for modules and external system elements?
- Are the data structures consistent with the information domain?
- Are the data structures consistent with the requirements?
- Has maintainability been considered?

Code:

- Does the code reflect the design documentation?
- Has proper use of language conventions been made?
- Has coding standards been observed?
- Are there incorrect or ambiguous?

Testing:

- Have test resources and tools been identified and acquired?
- Have both white and black box tests been specified?
- Have all the independent logic paths been tested?
- Have test cases been identified and listed with expected results?
- Are timing and performance to be tested?

Metrics Derived from Reviews

- Inspection time per page of the documentation
- Inspection time **per KLOC or FP**
- Inspection effort **per KLOC or FP**
- Errors uncovered **per reviewer hour**
- Errors uncovered **per preparation hour**
- Errors uncovered **per SE task (e.g Design)**
- Number of **minor errors (e.g. typos)**
- Number of **major errors (e.g. non-confirmation to requirement)**
- Number of **errors found during preparation.**

Thank You!!