Contents

1 Quality assurance and standards

- Standards may be international, national, organizational or project based
- Product standards define characteristics that **all components should exhibit** e.g. a common programming style and how the software process should be enacted

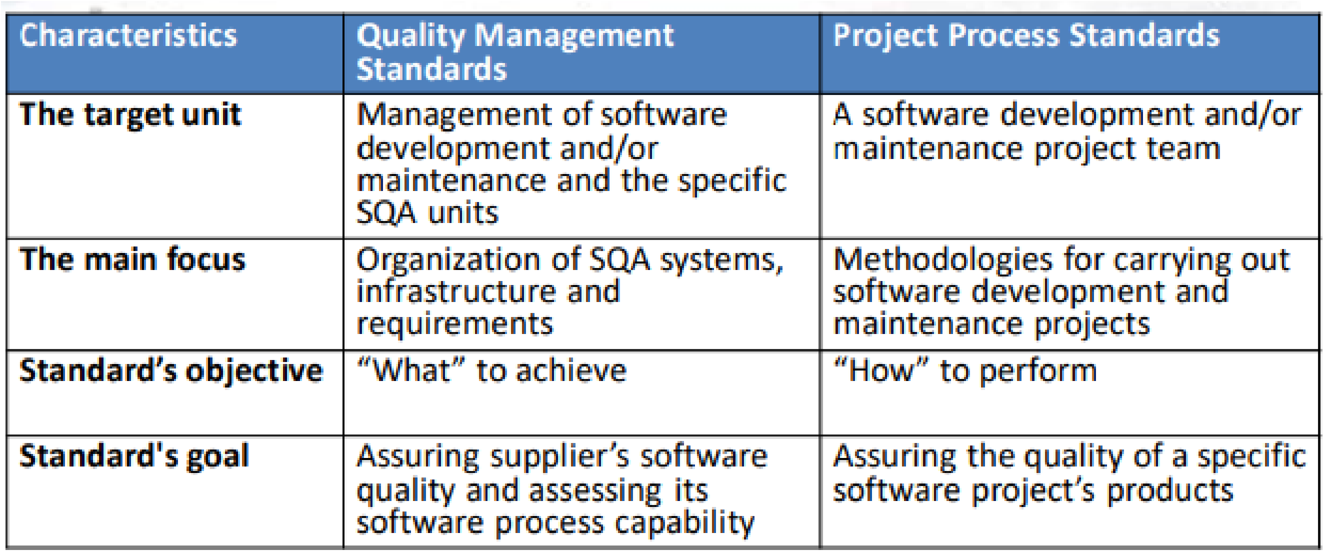
1.1 Importance of standards

- Encapsulation of best practice: **avoids repetition** of past mistakes
- Framework for quality assurance process: it involves **checking standard compliance**
- **Provide continuity**: new staff can understand the organisation by understand the standards applied

1.2 The benefits of using standards

- The ability to **apply methodologies and procedures** of the highest professional level.
- Better **mutual understanding** and coordination amongst teams.
- **Greater cooperation** between the software developer and external participants in the project. Suppliers and customers, based on the adoption of standards **as part of the contract**.

2 SQA classes



Characteristics	Quality Management Standards	Project Process Standards
The target unit	Management of software development and/or maintenance and the specific SQA units	A software development and/or maintenance project team
The main focus	Organization of SQA systems, infrastructure and requirements	Methodologies for carrying out software development and maintenance projects
Standard's objective	"What" to achieve	"How" to perform
Standard's goal	Assuring supplier's software quality and assessing its software process capability	Assuring the quality of a specific software project's products

2.1 Certification standards

- Enable a software development organization to demonstrate **consistent ability** to assure acceptable quality of its software products or maintenance services. Certification is granted by an **external body**
- Serve as an agreed-upon basis for customer and **supplier evaluation** of the suppliers quality management system. Accomplished by performance of a quality audit by the customer.

2.2 Assessment standards

- Serve organizations as a **tool for self-assessment** of their ability to carry out software development projects.
- Serve for **improvement of development** and maintenance processes by application of the standard directions
- Help purchasing organizations **to determine the capabilities** of potential suppliers.

3 ISO 9000

International set of standards for quality management. Applicable to a range of organisations from manufacturing to service industries. **ISO 9001**:

- is the current standard to which organisations can be certified
- applicable to organisations which design, develop and maintain products
- is a generic model of the quality process that must be instantiated for each organisation

3.1 ISO 9001 Certification

Quality standards and procedures should be documented in an organisational quality manual. **External body** may certify that an organisations quality manual conforms to **ISO 9001** standards. Customers are, increasingly, demanding that suppliers are **ISO 9001** certified

3.2 ISO 9001 principles

- Customer focus
  - Understand the needs of existing and future customers
  - Align organizational objectives with customer needs and expectations
  - Meet customer requirements
  - Measure customer satisfaction
  - Manage customer relationships
  - Aim to exceed customer expectations
- Leadership
  - Establish a vision and direction for the organization
  - Set challenging goals
  - Model organizational values
  - Establish trust
  - Equip and empower employees
  - Recognize employee contributions
- Involvement of people
  - Ensure that peoples abilities are used and valued
  - Make people accountable
  - Enable participation in continual improvement
  - Evaluate individual performance
  - Enable learning and knowledge sharing
  - Enable open discussion of problems and constraints
- Process approach
  - Manage activities as processes
  - Measure the capability of activities
  - Identify linkages between activities
  - Prioritize improvement opportunities
  - Deploy resources effectively
- Continual improvement
  - Improve organizational performance and capabilities
  - Align improvement activities

  - Empower people to make improvements
  - Measure improvement consistently
  - Celebrate improvements

- Factual approach to decision making
  - Ensure the accessibility of accurate and reliable data
  - Use appropriate methods to analyze data e Make decisions based on analysis
  - Balance data analysis with practical experienc

- Mutually supportive supplier relationships
  - Identify and select suppliers to manage costs, optimize resources, and create value
  - Establish relationships considering both the short and long term
  - Share expertise, resources, information, and plans with partners
  - Collaborate on improvement and development activities
  - Recognize supplier successes

4 Capability Maturity Model (CMM)

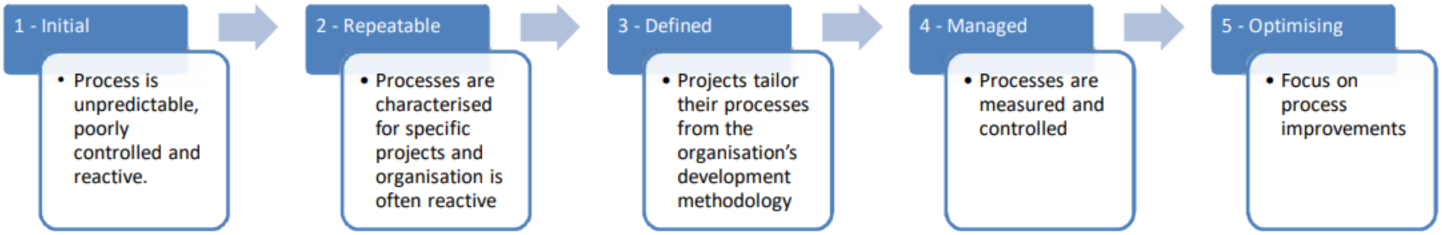
The Capability Maturity Model (CMM) is a methodology used to develop and refine an organization's software development process.

- *Quantitative* management methods increases the organization's capability to control the quality and improve the productivity.

- Application of the five-level capability maturity model that enables to evaluate the achievements and determine the efforts needed to reach the next capability.

- Generic process areas that define the what, **not how** enables the model's applicability to a wide range of implementation organizations:
  - Allows use of any life cycle model.
  - Allows use of any design methodology, development tool and programming language.
  - Does not specify any particular documentation standard.

4.1 Structure



1 - Initial
- Process is unpredictable, poorly controlled and reactive.

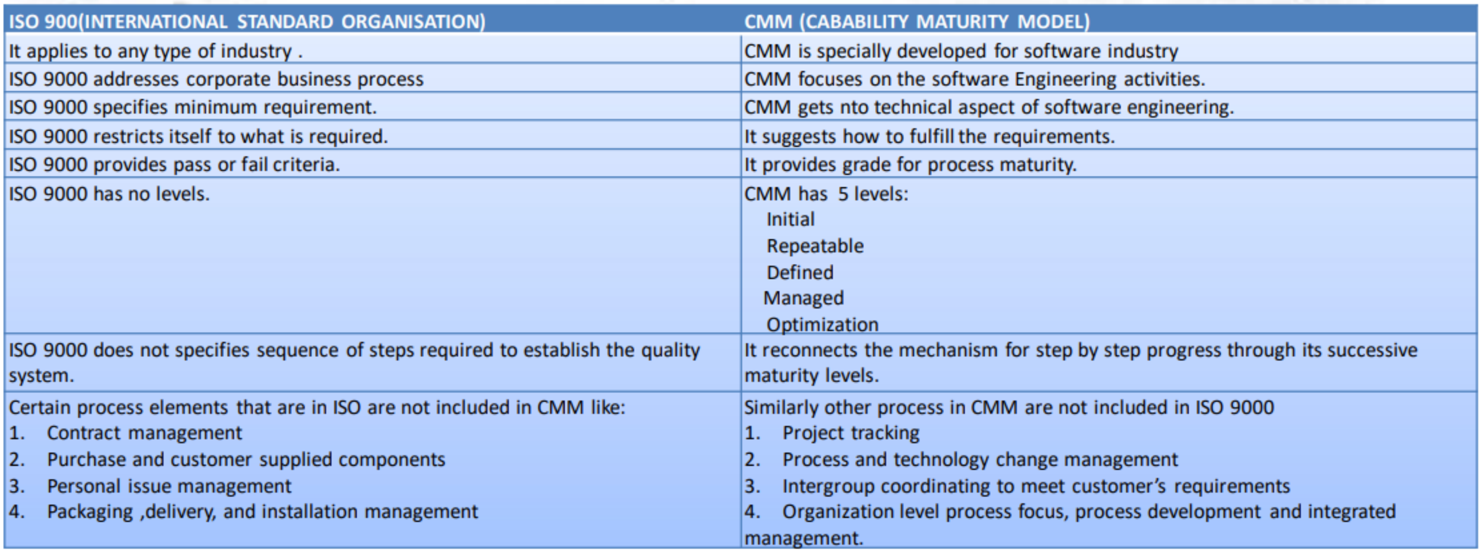
2 - Repeatable
- Processes are characterised for specific projects and organisation is often reactive

3 - Defined
- Projects tailor their processes from the organisation's development methodology

4 - Managed
- Processes are measured and controlled

5 - Optimising
- Focus on process improvements

4.2 Comparison of ISO 9000 vs CMM



ISO 900(INTERNATIONAL STANDARD ORGANISATION)	CMM (CABABILITY MATURITY MODEL)
It applies to any type of industry .	CMM is specially developed for software industry
ISO 9000 addresses corporate business process	CMM focuses on the software Engineering activities.
ISO 9000 specifies minimum requirement.	CMM gets nto technical aspect of software engineering.
ISO 9000 restricts itself to what is required.	It suggests how to fulfill the requirements.
ISO 9000 provides pass or fail criteria.	It provides grade for process maturity.
ISO 9000 has no levels.	CMM has 5 levels: Initial Repeatable Defined Managed Optimization
ISO 9000 does not specifies sequence of steps required to establish the quality system.	It reconnects the mechanism for step by step progress through its successive maturity levels.
Certain process elements that are in ISO are not included in CMM like: 1. Contract management 2. Purchase and customer supplied components 3. Personal issue management 4. Packaging ,delivery, and installation management	Similarly other process in CMM are not included in ISO 9000 1. Project tracking 2. Process and technology change management 3. Intergroup coordinating to meet customer's requirements 4. Organization level process focus, process development and integrated management.

5 Problems with standards

- Not seen as relevant and up-to-date by software engineers
- Involve too much bureaucratic form filling
- Unsupported by software tools so tedious manual work is involved to maintain standards

5.1 Overcoming the Problems

- Involve practitioners in development: Engineers should understand the rationale underlying a standard
- Review standards and their usage regularly: Standards can quickly become outdated and this reduces their credibility amongst practitioners
- Detailed standards should have associated tool support: Excessive clerical work is the most significant complaint against standards

6 Process and product quality

The quality of a developed product is influenced by the **quality of the production process**

6.1 Process-based quality

- It's complex to link process to product because:
  - The application of 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**

6.2 Practical process quality

- **Define** process standards such as how reviews should be conducted, configuration management, etc.
- **Monitor** the development process to ensure that standards are being followed
- **Report** on the process to project management and software procurer

7 Quality planning

- A quality plan sets out the **desired product qualities** and how these are **assessed** and define the most **significant quality attributes**
- It should define the quality assessment process
- It should set out which organisational standards should be applied and, if necessary, define new standards.

7.1 Structure

Quality plans should be short, succinct documents If they are too long, **no-one will read them**

- Product introduction
- Product plans
- Process descriptions
- Quality goals
- Risks and risk management

8 Management and its role in software quality assurance

8.1 The quality assurance organisational framework (actors)

Managers

- Top management executives, especially the executive in charge of SQA
- Software development and maintenance department managers
- Software testing department managers
- Project managers and team leaders of development and maintenance projects
- Leaders of software testing teams

Testers

- Members of software testing teams

SQA professionals and interested practitioners

- SQA trustees
- SQA committee members
- SQA forum members
- SQA unit team members

8.2 Top managements overall responsibilities for software quality

- Assure the **quality** of the companys software products and software **maintenance** services.
- Communicate the importance of product and service quality in addition to customer satisfaction to employees.
- Assure **full compliance** with customer requirements.
- Ensure that SQA objectives are established and accomplished.
- Initiate planning and **oversee implementation** of changes to adapt the SQA system to changes related to the organization's clientele, competition and technology.
- **Intervene directly** to resolve of crisis situations and minimize damages.
- **Ensure availability** of resources required by SQA systems.

8.3 Software quality policy requirements

Conformity to the organisation purpose and goals and commitment to:

- General software quality assurance concepts
- The quality standards adopted by the organization
- Allocate adequate resources for software quality assurance
- Continuous improvement of the organizations quality and productivity

8.4 The responsibilities of the executive in charge of software quality

- Preparation of an annual SQA activities program and budget
- Preparation of SQA system development plans
- Overall control of implementation of the annual SQA regular activities program and planned SQA development projects
- Presentation and advocacy of SQA issues to executive management

8.5 Typical items contained in management review reports

Management review is the name given to the periodic meeting convened to allow executives to obtain an overview of their organizations software quality issues.

- Periodic performance reports, including quality metrics
- Customer satisfaction feedback
- Follow up reports for SQA annual regular activity program and SQA development projects
- Summary of special quality events related to customers, suppliers, subcontractors, etc.
- Review of significant findings of internal and external quality audits as well as special surveys
- Identification of new software quality risks and unsolved pre-existing risks
- Recommendations for software quality management improvements.

8.6 The objectives of management reviews

- Assess achievement of quality objectives set for the organizations SQM system
- Initiate updates and improvements of the SQM system and its objectives
- Outline directions for remedying major SQA deficiencies and software quality management problems.
- Allocate additional resources to the SQM system.

8.7 Department management responsibilities for quality assurance

The quality system-related responsibilities

- Preparation of the departments annual SQA activities program and budget, based on recommended SQA unit program.
- Preparation of the departments SQA systems development plans, based on recommended SQA unit plan.
- Control of performance of the departments annual SQA activities program and development projects
- Presentation of the department's SQA issues to the executive in charge of software quality

Project-related responsibilities

- Control of compliance to QA procedures in the department's units
- Detailed follow up of contract review results and proposal approvals

- Review of unit performance of planned review activities; approval of project documents and project phase completion
- Follow up of software tests; approval of projects software products
- Follow up of progress of software development project schedules and budget deviations. Advise and support project mangers in resolving difficulties
- Follow up of quality of maintenance services
- Detailed follow up of project risks and their solutions
- Follow up of project's compliance with customer requirements and customers satisfaction
- Approval of large software change orders and significant deviations from project specifications

8.8 Project management responsibilities for quality assurance

Professional hands on tasks:

- Preparation of project and quality plans and their updates.
- Participation in joint customer-supplier committee
- Close follow up of project team staffing, including recruitment, training and instruction.

Management tasks - The follow up issues:

- Performance of review activities and the consequent corrections, including participating in some reviews.
- Software development and maintenance units performance with respect to development, integration and system test activities, corrections and regression tests and acceptance tests
- Software installation in customer sites and the running-in of the software system by the customer
- SQA training and instruction of project team members
- Schedules and resources allocated to project activities.
- Customer requests and satisfaction
- Evolving project development risks, application of solutions and control of results.

9 Quality control

Checking the software development process to ensure that procedures and standards are being followed. Two approaches to quality control: Quality reviews, Assessment via software metrics

9.1 Quality reviews

- The principal method of validating the quality of a process or of a product
- Group examines part or all of a process or system and its documentation to find potential problems
- There are different types of review with different objectives
  - Inspections for defect removal (product)
  - Reviews for progress assessment(product and process)
  - Quality reviews (product and standards)

9.1.1 Purpose of Reviews

- Serve as a **filter** for the software process
- Uncover errors that can then be removed
- Purify the software analysis, design, coding, and testing activities
- Catch large classes of errors that escape the originator more than other practitioners
- Include the formal technical review (also called a walkthrough or inspection)
  - Acts as the most effective SQA filter
  - Conducted by software engineers for software engineers
  - Effectively uncovers errors and improves software quality
  - Has been shown to be up to 75% effective in uncovering design flaws (which constitute 50-65% of all errors in software)
- Require the software engineers to expend time and effort, and the organization to cover the costs

10 Formal Technical Review (FTR)

- **Objectives:**
  - Uncover errors in function, logic, or implementation for any representation of the software
  - Verify that the software under review meets its requirements
  - Ensure that the software has been represented according to predefined standards
  - Achieve software that is developed in a uniform manner
  - Make projects more manageable
- Serves as a **training ground** for junior software engineers to observe different approaches to software analysis, design, and construction
- **Promotes backup and continuity** because a number of people become familiar with other parts of the software
- May sometimes be a sample-driven review
  - Project managers must quantify those work products that are the primary targets for formal technical reviews
  - The sample of products that are reviewed must be representative of the products as a whole

10.1 The FTR meeting

- Has the following constraints: From 3-5 people should be involved, advance preparation (i.e., reading) should occur for each participant **but should require no more than two hours** a piece and involve only a small subset of components, the duration of the meeting should be less than two hours
- Focuses on a specific work product (a software requirements specification, a detailed design, a source code listing)

10.2 Before the FTR meeting

- The producer informs the project manager that a work product is complete and ready for review
- The project manager contacts a review leader, who evaluates the product for readiness, generates copies of product materials, and distributes them to the reviewers for advance preparation
- Each reviewer spends one to two hours reviewing the product and making notes before the actual review meeting
- The review leader establishes an agenda for the review meeting and schedules the time and location

10.3 During the FTR meeting

- The meeting is attended by the review leader, all reviewers, and the producer
- One of the reviewers also serves as the recorder for all issues and decisions concerning the product
- After a brief introduction by the review leader, the producer proceeds to ”walk through” the work product while reviewers ask questions and raise issues
- The recorder notes any valid problems or errors that are discovered; no time or effort is spent in this meeting to solve any of these problems or errors

10.4 At the conclusion of the FTR meeting

- All attendees must decide whether to
  - **Accept** the product without further modification
  - **Reject** the product due to severe errors (After these errors are corrected, another review will then occur)
  - **Accept** the product provisionally (Minor errors need to be corrected but no additional review is required)
- All attendees then complete a sign-off in which they indicate that they took part in the review and that they concur with the findings

10.5 Following the FTR meeting

- The **recorder** produces a list of review issues that: **Identifies problem areas** within the product, Serves as an action item checklist **to guide the producer** in making corrections.
- The recorder includes the list in an FTR summary report: This one to two-page report describes **what** was reviewed, **who** reviewed it, and **what** were the findings and conclusions
- The review leader **follows up** on the findings to ensure that the producer makes the requested corrections

10.6 FTR Guidelines

- Review the product, **not the producer**
- Set an agenda and maintain it
- **Limit** debate and rebuttal; conduct in-depth discussions off-line
- **Enunciate problem areas**, but don’t attempt to solve the problem noted
- **Take written notes**; utilize a wall board to capture comments
- Limit the number of participants and insist upon advance preparation
- **Develop a checklist** for each product in order to structure and focus the review
- Allocate resources and schedule time for FTRs
- Conduct meaningful training for all reviewers
- Review your earlier reviews to **improve the overall review process**

11 Quality Metrics

- Software measurement is concerned with **deriving a numeric value** for an attribute of a software product or process
- A software metric is any type of **measurement** which relates to a software system, process or related documentation
- This allows for **objective comparisons** between techniques and processes. There are few standards and no systematic use

11.1 Quality attributes and software metrics

- **Maintainability**: the effort needed to make specified modifications.
- **Reliability**: the capability of software to maintain its level of performance under stated conditions for a stated period of time.
- **Portability**: the ability of software to be transferred from one environment.
- **Usability**: the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.

11.2 Important software metric assumptions

- A software property **can be measured**
- A relationship exists between what we can measure and a quality attribute
- This relationship has been formalized and validated

11.3 The measurement process

- A software measurement process **may be part of** a quality control process
- Data collected during this process **should be maintained** as an organisational resource
- Once a measurement database has been established, **comparisons** across projects become possible

11.4 Product metrics

A quality metric should be a **predictor** of product quality. Classes of product metric:

- Dynamic metrics
  - Collected by a program in execution (response time, number of failures)
  - Help assess efficiency, effectiveness, availability and reliability
- Static metrics
  - collected by measurements made of the system representations (lines of code)
  - help assess complexity, understandability and maintainability

Reference section

quantative

relating to, measuring, or measured by the quantity of something rather than its quality.