Chapter 3
Quality Management & Assurance

Google’s Nest’s Smart Thermostat Software Bug, January 2016

A December software update contained a bug which drained the device’s battery and deactivated the thermostat. Even when developers fixed the problem, they were left with annoyed customers who were less than happy to undertake the 9-step process to reboot the device ...

Knight Capital’s Faulty Trading Software, August 2012

Knight Capital had to give a big payout for its buggy trading software: $440 million. The software’s algorithm was designed to enact automatic orders over several days but a bug caused the software to make all the orders in 1 hour, buying and selling stock with a loss of up to 15 cents per share. Ultimately, the company lost 4x its 2011 profit and was acquired by Getco LLC, a rival company, in Dec 2012.

CNN BUSINESS

PayPal accidentally credits man $92 quadrillion

By Sho Wills, CNN

Updated 1355 GMT (2155 HKT) July 17, 2013

Account Statement | June 2013

PayPal

Reynolds, Christopher
Email (PayPal Account ID): [redacted]

Statement period:
June 1, 2013 - June 30, 2013

Balance Summary*

	USD
Beginning Balance	140.25
Ending Balance	-92,233,720,368,547,800.00

Account Activity

Table of Content

3.1 Quality Concepts & Principal Activities

- The Importance of Software Quality
- Defining Software Quality
- Principal Quality Activities

The Importance of Software Quality

- Increasing criticality of software. Organizations are relying more on their computer systems and software is used in more safety-critical applications.
- The intangibility of software makes it difficult to know that a project task was completed satisfactorily.
- Accumulating errors during software development. As the output from one step in software development is the input to the next, this leads to an accumulating detrimental effect.

Defining Software Quality

- Some qualities of a software product reflect the external view of software, e.g. *usability*.
- These *external qualities* have to be mapped to internal factors of which the developers would be aware.
  - E.g., well-structured code is likely to have fewer errors and thus improve reliability.
- Defining quality is not enough; we need to judge whether a system meets our requirements and therefore, need to be able to measure its qualities.

Software Quality Attributes

- Quality, simplistically, means that a product should meet its specification.
- The subjective quality of a software system is largely based on its non-functional characteristics.
  - This reflects practical user experience
- Refer to Appendix 3.1 ISO/IEC 25010 Quality Model & Attributes

Conflicting Software Quality Attributes

- There is a tension between customer quality requirements (efficiency, reliability, etc.) and developer quality requirements (maintainability, reusability, etc.);
- It is not possible for any system to be optimized for all of these attributes – e.g., improving robustness may lead to loss of performance.
- The quality plan should therefore define the most important quality attributes for the software that is being developed.

Software Quality Specification

Definition.	:	Definition of the quality characteristic.
Scale	:	The unit of measurement
Test	:	The practical test of the extent to which the attribute quality exists.
Minimally acceptable	:	The worst value which might be acceptable if other characteristics compensated for it, and below which the product would have to be rejected.
Target range	:	The range of values within which it is planned the quality measurement value should lie.
Now	:	The value that applies currently.

Principal Quality Activities

- Software Quality Management (SQM)'s 3 main activities:
  1. Quality Assurance (QA): the develop an organization framework (procedures & standards) that will lead to high quality software.
  2. Quality Planning (QP): the selection of appropriate procedures and standards from this framework and adapt for a specific software project.
  3. Quality Control (QC): the execution of processes to ensure that software development follows the quality procedures and standards.
- The QA team should be independent from the development team so that they can take an objective view of the work products.

Principal Quality Activities

It is process oriented

It is product oriented

Quality Planning	Quality Assurance	Quality Control
Selection of appropriate procedures and standards, and adapt to specific software product.	Definition of processes and standards that should lead to high quality products and the introduction of quality processes into the development process.	Application of the quality processes defined under quality assurance to weed out products that are not of the required level of quality.
The goal of this activity is to produce a quality plan (includes objectives, activities, standards) for all team members to be aware of and to follow.	During this activity, the organization identifies the procedures to set and follow, as well as the standards to adopt. This leads to the establishment of organizational procedures and standards to be adhered to to ensure high quality software.	This activity concerned with the enactment of processes which ensure that the project quality procedures and standards are followed and adhered to by software development team.

	Quality Assurance	Quality Control
Definition	QA is the implementation of processes, methodologies and standards that ensure that the software developed will be up to the required quality standards.	QC is the set of activities that are carried out to verify the developed product meets the required standards.
Target	QA focuses on the improvement of process and methodologies used to develop product.	QC focuses on the improvement of the product by identifying the bugs and issues.
Orientation	It is process oriented.	It is product oriented.
Nature of process	QA is preventive process as it establishes the methods which prevent the bugs.	QC is corrective process as it focuses on identifying the bugs and getting them fixed.
Verification vs Validation	Quality Assurance is a verification activity that verifies you are doing the right thing in the right manner.	Quality assurances is a validation activity that validates the product against the requirements.
Who	All the persons involved in the project starting from the requirement.	It is the responsibility of Quality Control inspector or the testing team that finds the issues.
Tools and Techniques	Defining Processes, Quality Audit, Selection of Tools, Training.	Defining Processes, Quality Audit, Selection of Tools, Training.
Examples	Examples of quality assurance activities include process checklists, process standards, process documentation and project audit.	Examples of quality control activities include inspection, deliverable peer reviews and the software testing process.

3.2 Quality Planning & Quality Control

- Introduction
- Quality Plans
- Quality Control

Introduction

- QA establishes the infrastructure that supports solid software engineering methods, rational project management and quality control actions.
- QC encompasses a set of software engineering actions that help to ensure that each work product meets its quality goals.
- Some organizations produce quality plans for each project.

Quality Plans

A quality plan

- Specifies the desired product qualities and how these are assessed and defines the most significant quality attributes.
- Should defined the quality assessment process
- Should indicate which organizational standards should be applied and where necessary, define new standards to be used.

Quality Plans Structure

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

Note:

- quality plans should be short, succinct documents
- if they are too long, no one will read them.

Quality Control (QC)

- QC involves monitoring the software development process to ensure that QA procedures and standards are being followed.
- The deliverables from the software development process are checked against the defined project standards during the QC process, either using quality reviews and/or automated software assessment.

Quality Reviews

- Quality reviews involve a group of people examining part or all of a software process, system, or its documentation to discover potential problems.
- Outcome of the review

  - List of problems found
  - author to do corrections

Quality Reviews

Review Type	Principal Purpose
Design or program inspections	To detect detailed errors in the requirements, design or code.
Progress reviews	To provide information for management about the overall progress of the project.
Quality reviews	To carry out a technical analysis or product components or documentation to find mismatches between the specification and the component design, code or documentation and to ensure that defined quality standards of the organization have been followed.

3.3 Process & Product Quality Relationship

3.3 Process & Product Quality Relationship

- In general, the quality of the development process directly affects the quality of delivered products. The quality of the product can be measured and the process is improved until the proper quality level is achieved.
- In manufacturing systems, there is a clear relationship between the production process and product quality. However, quality of software is highly influenced by the experience of software engineers. In addition, it is difficult to measure certain software quality attributes and to tell how process characteristics influence these attributes.

3.4 Techniques to Help Enhance Software Quality

- Inspections
- Cleanroom software development
- Software quality circles
- Lessons learnt report

Inspections

- Inspections can be applied to documents produced at any development stage.
- When a piece of work is completed, copies are distributed to co-workers who examine the work, noting defects/problems. A meeting then discusses the work and a list of defects requiring rework is produced.
- E.g. of works that may be examine:

  - Table design (e.g. fields in a table – too little?, too many?)

  - Program codes (e.g. length of variable names, etc)

- Inspections is a.k.a. Fagan method named after an IBM employee who pioneered the technique

Benefits of Inspections

- It is a very effective way of removing superficial errors.
- It helps developers to produce better structured and self-explanatory software.
- It helps spread good programming practices as the participants discuss specific pieces of code.
- It can enhance team spirit.

Principles of the Fagan method

- Inspections are carried out on all major deliverables.
- All types of defect are noted - not just logic or function errors.
- Inspections can be carried out by colleagues at all levels except the very top.
- Inspection meetings do not last for more than 2 hours.
- Inspections are carried out using a predefined set of steps.
- The inspection is led by a *moderator* who has been trained in the technique.
- The other participants have defined roles (e.g., one person will act as a recorder and note all defects found, and another will act as reader and take the other participants through the document under inspection).
- Checklists are used to assist the fault-finding process.
- Material is inspected at an optimal rate of about 100 lines an hour.
- Statistics are maintained so that the effectiveness of the process can be monitored.

Cleanroom Software Development

- Improved from structured programming: de-component complex system so that each component has only one entry and exit point which make testing easier and more precise.
- Aims to *avoid defect* rather than detect and repair them:
  - Use incremental development approach
  - Use statistical testing/verification
- Rationale for incremental approach:
  - User requirement changes are inevitable
  - New requirements are added later; Main modules given first (gradually increase the implemented functionality)
  - Each new function will be tested and measured against pre-defined standards. If fail to meet the standards, go back to design phase.

Cleanroom Software Development

There are 3 separate teams:

- Specification team
  - obtains the user requirements and a *usage profile* estimating the volume of use for each feature in the system
- Development team
  - develops the code in increments but which does no testing of the program code produced
- Certification team
  - carries out testing which is continued until a statistical model shows that the failure intensity has been reduced to an acceptable rate

Cleanroom Benefits & Weaknesses

- Benefits
  - Lower number of errors
    (Codes are develop by dev-team but testing are carried out by certification-team)
  - Lower cost
    (Lower no of errors means there will be less rework, less rework is a cost saving)
  - Project on schedule
    (Incremental approach able to cope with requirement changes)
  - Software of higher quality
    (Lower no of errors means higher software quality)
- Weaknesses
  - Works well with skilled and committed engineers only.
  - The usage of this approach is confined to a few, technologically advanced organizations only.

Software Quality Circle

- The aim of Japanese quality circle approach is to examine and modify the activities in the development process in order to reduce the number of errors that they have in their end-products.
- Testing and Fagan inspections can assist the removal of errors, but the same types of error could occur repeatedly in successive products created by a faulty process. By uncovering the source of errors, this repetition can be eliminated.

Quality Circle

- A quality circle is a group of 4 to 10 volunteers working in the same area who meet for about an hour a week to identify, analyze and solve their work-related problems.
  - One volunteer is the group leader.
  - There could be an outsider, a facilitator, who can advise on procedural matters.
- In order to make the quality circle work effectively, training needs to be given.
- Together, the group
  - selects a pressing problem that affects their work;
  - Identify the cause of the problem, and
  - Decide on a course of action to remove these causes.

- A faulty process will repeatedly produce faulty products.
- Quality Circle is to fix the process so that the same types of error will not occur again

Lessons Learnt Reports

- Written by the project manager as soon as possible after the completion of the project.
- Includes
  - reflection on the performance of a project at its immediate end when the experience is still fresh, and
  - identifying lessons to be applied to future projects.
- However, one frequent problem is there is often very little follow-up on the recommendations of such reports, as there is often nobody within the organization with the responsibility and authority to do so.

3.5 Quality Management & Costs

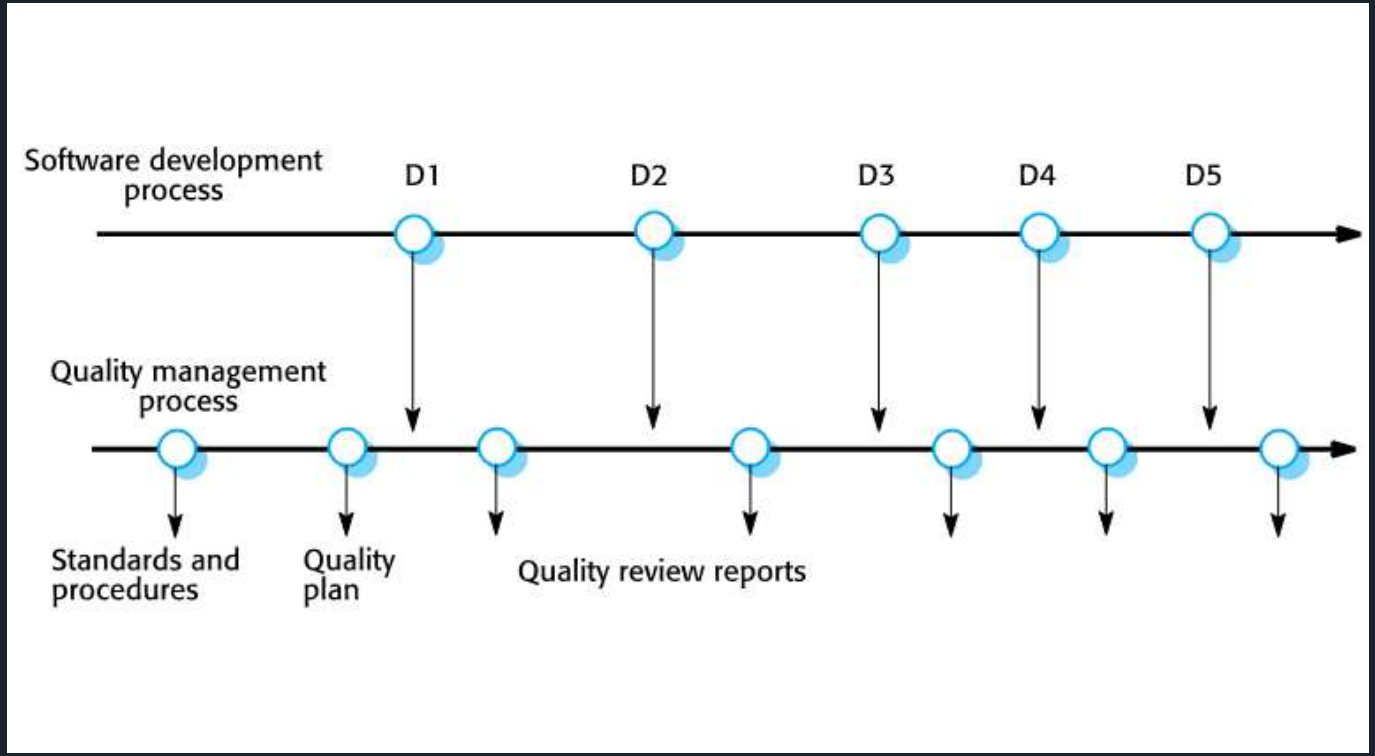
- Quality Management
- Cost of Quality
- Categories of Quality Costs
-

Quality Management

- is concerned with ensuring that the required level of quality is achieved in a software product.
  - At the organizational level, it is concerned with establishing a framework of organizational processes and standards that will lead to high-quality software.
  - At the project level, quality management
    - involves the application of specific quality processes and checking that these processes have been followed.
    - concerned with establishing a quality plan for a project. The quality plan should set out the quality goals for the project and define what processes and standards are to be used

Quality Management (cont'd)

Quality management provide an independent check on the software development process.



Cost of Quality (COQ)

Cost of quality	=	Cost of achieving quality*	+	Cost of low-quality software

*includes costs incurred in the pursuit of quality or in performing quality-related activities

Categories of Quality Cost

The cost of quality can be divided into

a. Prevention Costs
b. Appraisal Cost
c. Failure Costs

Which cost is the most costly to a software company?

a. Prevention Costs

The cost arises from efforts to prevent defects.

Includes the costs of

- management activities required to plan and coordinate all QC and QA activities,
- added technical activities to develop complete requirements and design models,
- test planning, and
- Conduct training associated with these activities

The goals of these activities are to ensure that a project is error-free or within an acceptable range

b. Appraisal Costs

The cost arises from efforts to detect defects.

Includes the costs of

- Conducting technical reviews
- Data collection and metrics evaluation
- Testing and debugging

These activities associated with checking processes and its output.

The goals are also to ensure that a project is error-free or within an acceptable range

c. Failure Costs

cost arises from defects identified by the client or end-users and efforts to correct

- Internal failure costs (i.e. means error detected in a product prior to shipment)
- Costs incurred include:
  - Perform rework to correct an error,
  - Costs of correcting design error
- External failure costs (i.e. when defects are found after the product has been shipped to the customer) which include the costs for
  - Complaint resolution
  - Product return and replacement
  - Help line support
  - Poor reputation and the resulting loss of business

3.6 Quality Assurance & Standards

- Quality Standards
- ISO/IEC 25010:2011
- ISO 9001 Standards Framework
- Quality Culture

Quality Standards

- Standards define the required attributes of a product or process.
- Standards play an important role in quality management.
- Standards may be international, national, organizational or project standards.

Importance of Quality Standards

- Encapsulation of best practice - avoids repetition of past mistakes.
- They are a framework for defining what quality means in a particular setting, i.e. the organization's view of quality.
- They provide continuity - new staff can understand the organization by understanding the standards that are used.

Product & Process Standards

- Product standards - apply to the software product being developed.
  - Include document standards (e.g., the structure of requirements documents, documentation standards, coding standards)
- Process standards - define the processes that should be followed during software development
  - Include definitions of specification, design and validation processes, process support tools and a description of the documents that should be written during these processes.

Product & Process Standards Examples

Product Standards	**Process Standards**
Design review form	Design review conduct
Requirements document structure	Submission of new code
Method header format	Version release process
Java programming style	Project plan approval process
Project plan format	Change control process
Change request form	Test recording process

Problems with Standards

- They may not be seen as relevant and up-to-date by software engineers
- They often involve too much bureaucratic form filling
- If they are unsupported by software tools, tedious form filling work is often involved to maintain the documentation associated with the standards

Standards Development

- Involve practitioners in development - engineers should understand the rationale underlying a standards.
- Review standards and their usage regularly - standards that are outdated will have reduced credibility among practitioners
- Detailed standards should have specialized tool support - excessive clerical work is the most significant complaint against standards.

ISO/IEC 25010:2011

Appendix 3.1

The ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System & software quality models defines

- A quality in use model composed of 5 characteristics* that relate to the outcome of interaction when a product is used in a particular context of use.
- A product quality model composed of 8 characteristics that relate to static properties of software and dynamic properties of the computer system
  **some of which are further subdivided into sub-characteristics*

Note: this standard replaced the ISO/IEC 9126 Software engineering-Product quality standard.

Table 2 — Product Quality characteristics and subcharacteristics

Functional suitability	**Reliability**
Functional completeness	Maturity
Functional correctness	Availability
Functional appropriateness	Fault tolerance
Performance efficiency	Recoverability
Time behaviour	**Security**
Resource utilization	Confidentiality
Capacity	Integrity
Compatibility	Non-repudiation
Co-existence	Accountability
Interoperability	Authenticity
Usability	**Maintainability**
Appropriateness recognizability	Modularity
Learnability	Reusability
Operability	Analysability
User error protection	Modifiability
User interface aesthetics	Testability
Accessibility	**Portability**
	Adaptability
	Installability
	Replaceability

ISO/IEC 25010:2011 can benefit these activities:

- Identifying software & system requirements
- Validating the comprehensiveness of a requirements definition
- Identifying software and system design objectives
- Identifying software and system testing objectives
- Identifying quality control criteria as part of QA
- Identifying acceptance criteria for a software product
- Establishing measures of quality characteristics in support of these activities

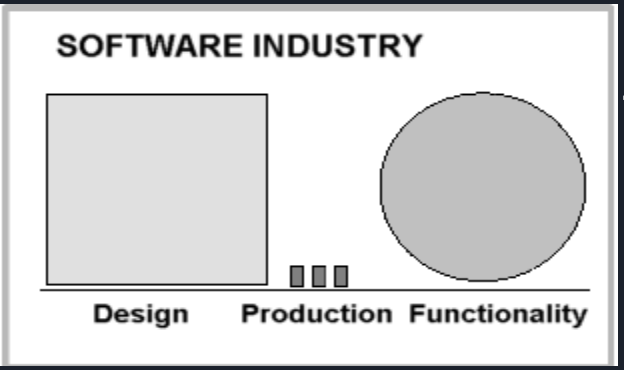
ISO/IEC 25010:2011 Quality in Use Model

- Effectiveness
- Efficiency
- Satisfaction
  - Usefulness
  - Trust
  - Pleasure
  - Comfort
- Freedom from risk
  - Economic risk mitigation
  - Health & safety risk mitigation
  - Environmental risk mitigation
- Context coverage
  - Context completeness
  - Flexibility

Refer to https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en for details.

The ISO 9001 Standards Framework

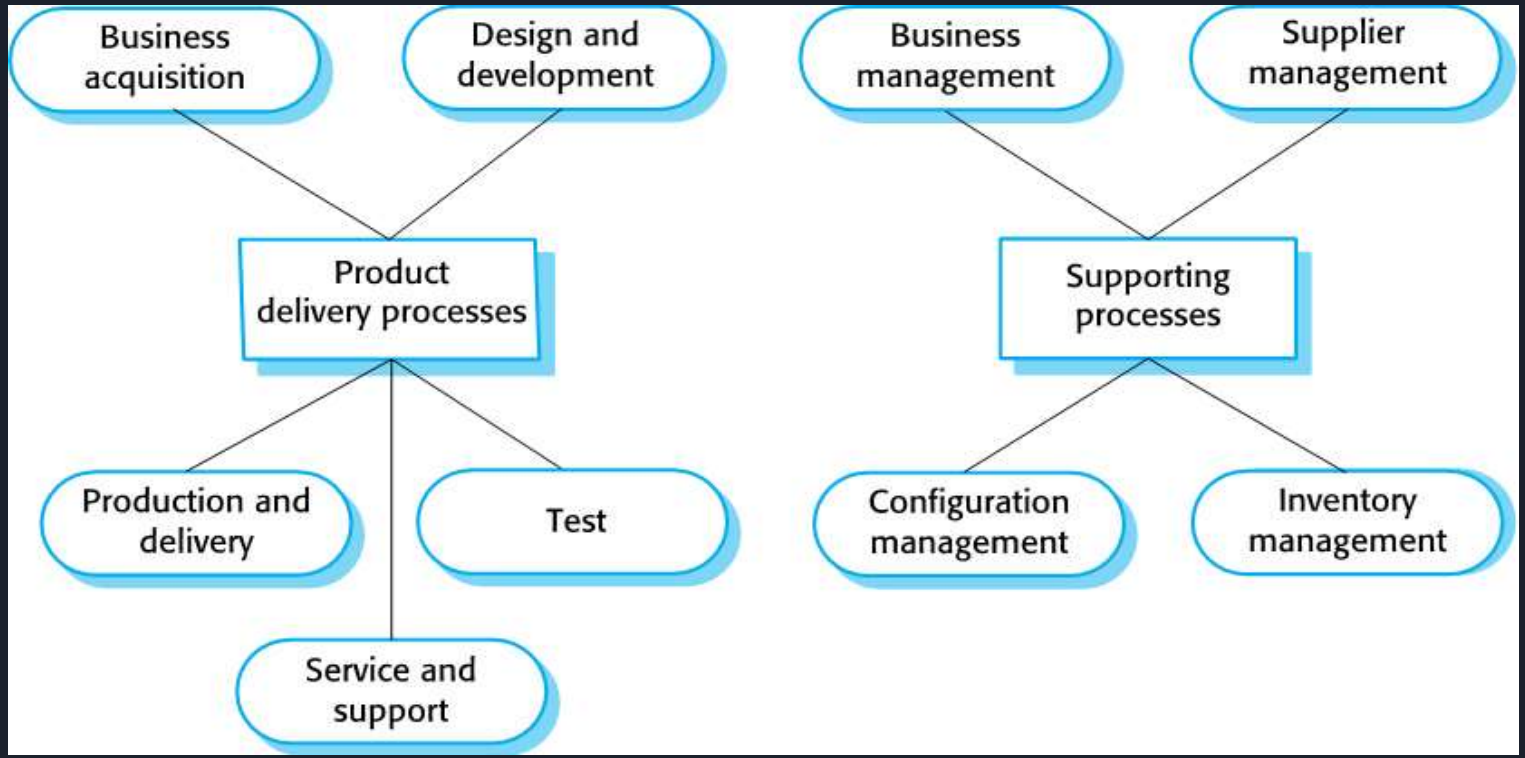
- An international set of standards that can be used as a basis for developing quality management systems.
- ISO 9001, the most general of these standards, applies to organizations that design, develop and maintain products, including software.
  - It sets out general quality principles, describes quality processes in general and lays out the organizational standards and procedures that should be defined. These should be documented in an organizational quality manual.



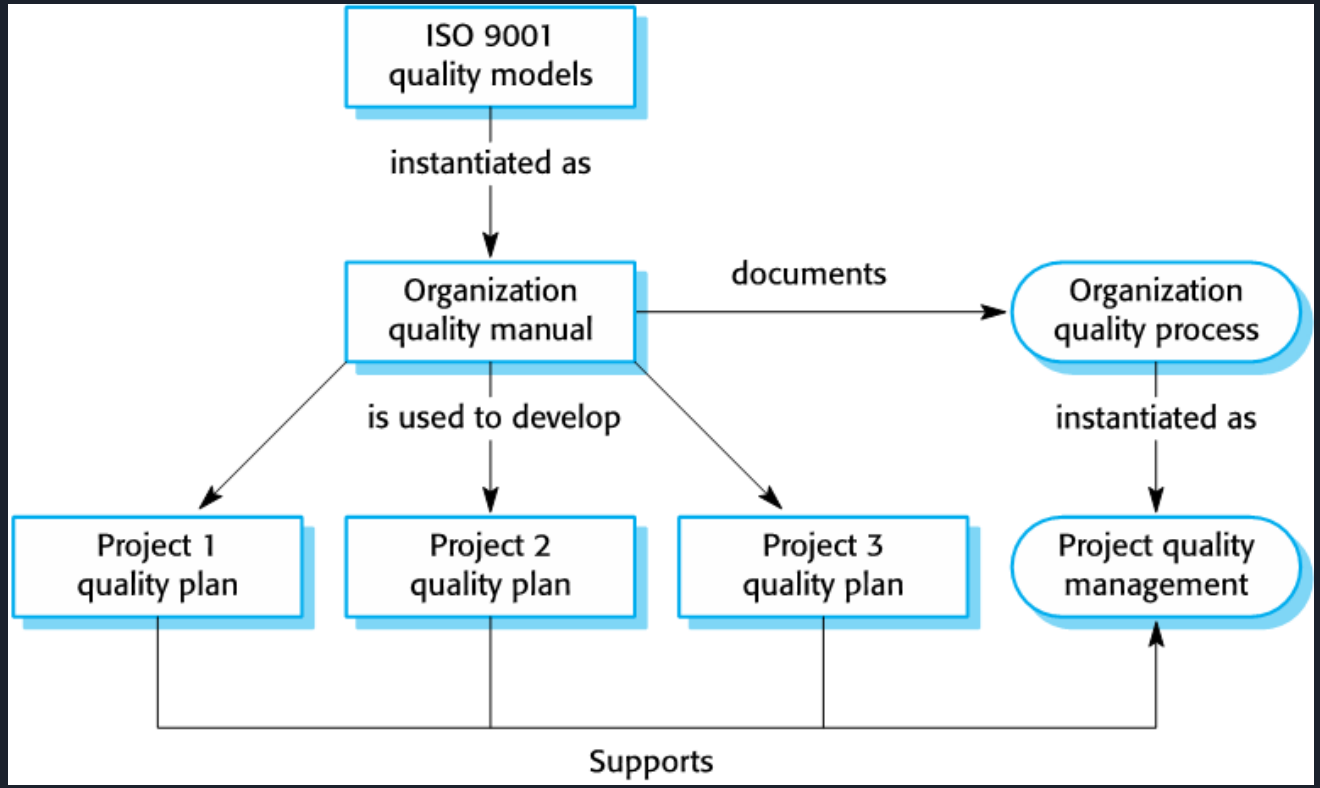
ISO 9001 Characteristics

- ISO 9001 is *not itself a standard for software development* but is a framework for developing software standards
- It sets *general quality principles*, describes quality processes in general, lays out the organizational standards and procedures that should be defined. These must be documented in an organizational quality manual.
- To be conformant with ISO 9001, a company must have defined the types of processes shown in the following figure and procedures that demonstrate that its quality processes are being followed.

ISO 9001 Core Processes



ISO 9001 & Quality Management



ISO 9001 Certification

- Quality standards and procedures should be documented in an organizational quality manual.
- An external body may certify that an organization’s quality manual conforms to ISO 9000 standards.
- Some customers require suppliers to be ISO 9000 certified.

Discussion in tutorial 3

Quality Culture

- Organizations should aim to develop a quality culture where everyone responsible for software development is committed to achieving a high level of product quality.
- Teams should be encouraged to take responsibility for the quality of their work and to develop new approaches to quality improvement.
- They should support people who are interested in the intangible aspects of quality and encourage professional behaviour in all team members.

Summary

3.1 Quality Concepts & Principal Activities
3.2 Quality Planning & Quality Control
3.3 Process and Product Quality Relationship
3.4 Techniques to Help Enhance Software Quality
3.5 Quality Management and Costs
3.6 Quality Assurance and Standards