ASSIGNMENT QUESTIONS

Seminar Objectives

At the end of this seminar you should:
• understand the need for and the role of quality in systems development
• be able to define the terms quality, standards and reviews;
• be aware of the quality attributes which may be applied to system development products and process;
• be aware of standards and reviews and their importance to quality
Some definitions of Quality

- **Degree of excellence** (Oxford)
- **Fitness for purpose (AS1057)**
  - includes quality of design, the degree of conformance to design, it may include such factors as economic or perceived values
- **Ability to satisfy stated/implied needs** (ISO8402)
- **Conformance to requirements** (Crosby, Horch)

Determining Quality ...

- when having a meal in a restaurant
- when purchasing a car
- when buying a computer

The requirements vary immensely, and some of the success measures are very hard to quantify. Quality means different things to different people ... and it varies in different situations.
Why should it concern us?

- Customers' expectations and demands are increasing
- Competitors provide it
- Substantial savings demonstrated

<table>
<thead>
<tr>
<th>PRODUCTIVITY</th>
<th>COMPETITIVE POSITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUALITY</td>
<td>COST SAVINGS</td>
</tr>
<tr>
<td>EFFICIENCY</td>
<td></td>
</tr>
</tbody>
</table>

Information System Problems

- does not meet the client's business or processing needs
- does not support the client's working methods
- is unstable and unreliable
- does not improve productivity
- is difficult to use or requires excessive training to use
- is expensive to maintain
- is incomplete
- is expensive to operate
- has a short life span
- is delivered late
- costs more than budget
- cannot grow with the organisation
- does not produce a return on investment

Error Detection

- “Effort spent on software maintenance is greater than that spent on software development.”
- “An error is typically 100 times more expensive to correct in the maintenance phase on large projects, than in the requirements phase.”

Error Detection

The cost of detecting and correcting errors rises greatly during the systems development cycle.

$\quad$ Initiation $\quad$ Analysis $\quad$ Design $\quad$ Implementation

In addition to this is the cost to the organisation of having an incorrect system.

Quality Costs

The tip of the iceberg

Obvious upfront costs to the organisation

Review, Inspection, Re-do

User complaints, Downtime, Loss of sales, Re-testing, Re-documenting, Re-training, Overtime, Customer complaints, Financial losses, Employee turnover

Quality Requirements

- Correctness - Does it accurately do what is intended?
- Reliability - Does it do it right every time?
- Efficiency - Does it run as well as it could?
- Integrity - Is it precise and unambiguous?
- Usability - Is it easy to use?
- Maintainability - Is it easy to fix?
- Testability - Is correctness easy to check and verify?
- Flexibility - Is it easy to adapt and extend?
- Portability - Can it be easily converted?
- Reusability - Does it consist of general purpose modules?
- Interoperability - Will it integrate easily with other systems?
The Quality Process

The quality process involves the functions of:

**Quality Assurance** - A planned and systematic pattern of all actions necessary to provide adequate confidence that the product optimally fulfills customers' expectations, i.e. that it is problem-free and well able to perform the task it was designed for.

**Quality Control** - Quality control is concerned with ensuring that the required qualities are built into all of the products throughout their development life cycles. Quality control utilizes measurable quality criteria and is exercised via change control, quality reviews, project reviews and by the testing of products.


Implementing a Quality System

- Quality must start at the top - Executive sponsorship is vital.
- Everyone must be involved and motivated to realize that they have a responsibility towards the final product, its use, and its quality.
- Improve job processes by using standards, and preparing better documentation (using project control methodologies).
- Use a QA group.
- Use reviews.

Standards

- Two levels of standards
  - Industry / National / International
  - Organisational
- Capability Maturity Model - A scale for assessing the degree of built-in documentation and discipline in a process, in which the scale goes from Level 1, with no formal process, to Level 5, with a continuous, rigorous and self-improving process. Developed by the Software Engineering Institute of Carnegie Mellon University - 1984, and now being extended to a broader range of applications in management – over 5000 assessments since then.
- National / International
  - Standards Australia (AS 3563)
  - International Standards Organisation (ISO 9000 – IMS9004)
- Organisational
  - The organisation may adopt or tailor industry, national or international standards.
Standards

- Standards can be of varying levels of enforcement and type which will depend on the organisation and project variables.
  For example:
  - mandatory practice must be adhered to
  - advisable practice can be breached with good reason
  - form is a checklist, template, or form.

Standards - Examples

- Document template (form - eg template for these slides)
- Acceptance test sign off form (form)
- Screen standards (standard - mandatory practice)
- Unit test process (standard - mandatory practice)
- VB.Net standards (standard - mandatory practice)
- Post implementation review procedure (advisory practice)
  Note: different organisations and projects will have different opinions on whether a standard is mandatory or advisable.

Reviews

- Reviews are used in the quality control and quality assurance functions. There are two main forms of review:
  - management reviews
  - technical reviews
Management or Project Review

- Management must check the baseline for a deliverable to see it meets the quality assurance requirements.
- This may involve simply noting that a technical review has passed a particular deliverable. The manager can then be assured of quality (given that the manager has actively taken part in the development of the quality system).
- The manager can then alter the project plan if necessary to allow for delays or early completion.

Technical Reviews

- A technical review (from here on abbreviated to review) is a structured meeting where a piece of work, which has previously been distributed to participants, is checked for errors, omissions, and conformance to standards.
- All deliverables need review. Otherwise how do you control quality?
- The review is part of quality control and must produce a report so that the quality assurance function can be satisfied. The report may be a checklist which indicates that the deliverable passes/fails the quality requirements for that type of deliverable. This report is part of the baseline for the deliverable.

Technical Reviews

- A technical review:
  - is a formal meeting of a team who are guided by an agenda and standards
  - allows input from many people
  - produces a report which is made public
  - requires committed participants to be responsible and accountable for their work
  - is educational as it clarifies standards, and highlights strengths and weaknesses of the teams skills and knowledge
  - expects all participants to be responsible for the resulting quality of the artefact
References


Additional Notes

Quality Attributes – detailed

Correctness

Does it accurately do what is intended?
- meets the specification
- fulfils the user’s objectives
- Note that these may be contradictory requirements, given variations in:
  > the quality of the analysis process
  > the speed of environmental change in the systems domain of operation. For example, a good specification can produce the wrong system if development is slow and the environment changes quickly
Reliability

Does it do it right every time?
– The system doesn’t malfunction or fail (in normal use).
– The system performance is not diminished too much, during periods of heavy use.
– When the system fails (and it will), recovery is both possible and rapid, with no loss of data.

Efficiency

Does it run as well as it could?
– The system makes good use of:
  > machine resources
  > human resources.
– The amount of resources needed to perform a function

Integrity

Is it precise and unambiguous?
– Terminology is consistent
– The design is consistent.
– Programming practices are consistent.
Usability

- Is it easy to use? ....
  - learn
  - operate
  - prepare input for
  - interpret output from

Maintainability

- Is it easy to fix?
  - can areas requiring change be located easily?
  - can changes be made easily?
  - can documentation be updated easily?

The system must be structured so changes are limited in scope (have minimal impact beyond the area being changed)

Testability

- Is correctness easy to check and verify?
  - Test strategy part of design process.
  - System-specific test data generator available to developers and maintainers.
  - System structured to support module testing and integration testing
Flexibility

- **Is it easy to adapt and extend?**
  - The system is designed to be changed as the environment changes.
  - Performance is sacrificed for flexibility, e.g., small parameter tables are used rather than hardwired code.

Portability

- **Can it be easily converted?**
  - Limited (and explicitly detailed) use of hardware-specific features.
  - Limited (and explicitly detailed) use of proprietary software features.
  - Hardware and software performance tuning makes minimal use of such features.

Reusability

- **Does it consist of general purpose modules?**
  - Highly modular black box design.
  - Mechanisms in place to reward developers for writing for reuse and reusing existing modules.
  - Tools, techniques and standards necessary to describe, catalogue and retrieve modules from an organisation wide library.
Reusability

- Reuse should not be confined to program code - it can be reuse of analysis specifications or documentation standards...
- Mechanisms in place to reward developers for writing for reuse and reusing existing modules.
- Tools, techniques and standards necessary to describe, catalogue and retrieve modules from an organisation wide library. The reuse should not be confined to program code - it can be reuse of analysis specifications or documentation standards - why re-invent the wheel?

- Why re-invent the wheel?

Interoperability

• Will it integrate easily with other systems?
  - Accept from the beginning that the system will have to integrate with other systems (data, presentation, control, and platform integration)
  - Standard formats and protocols for integration built into system
  - For example, facilities for data exchange part of initial design. Such facilities must be included in the specification, even when not asked for in the initial user specification.

Notes for Technical Review Roles

• Participants take on different roles in a review:
  - review member
  - review leader
  - scribe
Review Member

- Pre-requisites for review member:
  - Know the review process
  - Be positive and supportive
  - Interested in improving the quality of the deliverable and the review process

- Preparation for review:
  - Read and annotate material before review (be prepared)

Review Leader

- Pre-requisites for review leader:
  - Know the review process
  - Know the objectives & standards for the review
  - Be objective & aware of the implications the review may have for certain participants

- Prepare:
  - Agenda
  - Organise venue & any ancillary materials necessary
  - Notify participants & ensure they have review deliverable, agenda, & standards in advance.

- During the review:
  - Were preparations suitable? - if not you may have to re-schedule
  - Secure agreement on objectives and standards
  - Encourage input from all participants, and politely silence overly talkative participants
  - Facilitate agreement to ensure action on the deliverable
  - Ensure participants understand he ramifications of these actions
  - Ensure scribe has accurately documented review
Review Leader

• After the review:
  – Was the review successful?
  – Is the outcome likely to produce a better quality product?
  – Can the review process be improved? Can the quality of the quality system be improved? - this is continuous improvement.
  – Do the objectives and standards for the type of deliverable need review
  – Is the presenter of the deliverable capable of improving the quality of the item
  – When will the deliverable be reviewed again? Who is responsible? (Are they aware of the responsibility?)

Scribe

• Pre-requisites for the scribe:
  – Know the review process
  – Know the objectives and standards for the current review
  – Be objective and aware of the implications the review may have for certain participants

Scribe

• During the review:
  – Record all issues clearly, accurately, and unambiguously.
    If not sure on a particular issue, seek clarification
  – Be sure to use clear reference pointers between the review action list (report) and the review deliverable.
  – Gather copies of any materials introduced by participants to support an issue.

• After the review:
  – Check the review action list is accurate and promptly distributed to relevant people (who may not have been in the review)