Software Quality Management

Concerned with ensuring that the required level of quality is achieved in a software product

Two typical concerns:

- At the organisational level, quality management is concerned with establishing a framework of organisational processes and standards that will lead to high quality software
- At the project level, quality management involves establishing a quality plan for a project which sets out the quality goals for the project and define what processes and standards are to be used; also checking that these planned processes have been followed

1 Quality management activities

Quality management provides an independent check on the software development process

The quality management process checks the project deliverables to ensure that they are consistent with organisational standards and goals

The quality team should be independent from the development team so that they can take an objective view of the software. This allows them to report on software quality without being influenced by software development issues

2 Quality planning

A quality plan sets out the desired product qualities and how these are assessed and defines the most significant quality attributes.

The quality plan should define the quality assessment process

It should set out which organisational standards should be applied and, where necessary, define new standards to be used

3 Software quality

Quality, simplistically means that the product should meet its specification

This is a problem for software systems:

- There is a tension between customer quality requirements and developer quality requirements
- Some quality requirements are difficult to specify in an unambiguous way
- Software specifications are usually incomplete and often inconsistent

3.1 Software fitness for purpose

- Have programming and documentation standards been followed in the development process?
- Has the software been properly tested
- Is the software sufficiently dependable to be put into use
- Is the performance of the software acceptable for normal use
- Is the software usable
- Is the software well structured and understandable

4 Reviews and inspections

Both are quality assurance activities to check the quality of project deliverables: documents 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)

Software or documents may be ”signed off” at a review which signifies that progress to the next development stage has been approved by management

4.1 Review process

Code, designs, specifications, test plans, standards etc can all be reviewed

Review should check consistency and completeness of the docs or code under review and make sure quality standards have been followed

4.2 Code inspection

Team members from different backgrounds examine source code line by line with the aim of discovering errors and defects

A defect is a block of code which does not properly implement its requirements or could be improved

Checklist of common programming errors is often used to focus the search for bugs

Each organisation should also develop its own checklist based on local standards and practices

4.3 Agile methods

Reviews are usually informal in agile development

Agile processes rarely use formal inspection or peer review processes

Rather, they rely on team members cooperating to check each others code

5 Estimation

Estimating project costs is one of the crucial aspects of project planning and management

Estimating cost has to be done as early as possible during the project life cycle

Types of costs:

- Facilities: hardware, space, furniture, telephone etc
- Software tools for designing software
- Staff (effort): the biggest component of cost

5.1 Estimation techniques

Organisations need to make software effort and cost estimates. There are a few techniques that can be used to do this

5.1.1 Expert judgement

The estimate of future effort requirements are based on the managers experience of past projects and the application domain. Essentially, the manager makes an informed judgement of what the effort requirements are likely to be

The disadvantage of this is that it is no better than the expertise and objectivity of the estimator, who may be biased. This is overcome by having a group consensus

5.1.2 Top down

- Estimate overall cost from global properties of the product
- Split up among various components
- Disadvantage: low level tech problems not identified

5.1.3 Bottom up

- Estimate made for each component by the developer
- Costs summmed
- Disadvantage: can look over many system level costs

5.1.4 Algorithmic cost modelling

In this approach, a formulaic approach is used to compute the project effort based on estimates of product attributes, such as size, and process characteristics, such as experience of staff involved, reuse and approach to software development

An example of this is CoCoMo II, this algorithmic model uses:

- Scale drivers (on a 5 point scale) describe your project and determine the exponent in the effort equation based primarily on the software project size
- Cost drivers (15 of these) assess the project development environment and team
- Scale drivers:
  - **Precedentedness** - is the project comparable to projects your team has done before
  - **Dev flexibility** - are your reqs flexible, or must you meet them all?
  - **Architecture/risk resolution** - to what degree have you already defined the architecture
  - **Team cohesion** - how would you describe the relationships among the stakeholders?
  - **Process maturity**

5.2 Causes of inaccurate estimates

- Frequent request for change by users
- Overlooked tasks
- User's lack of understanding of the requirements
- Insufficient analysis when developing estimates
- Lack of coordination of system development, technical services, operations, data administration and other functions during development
- Lack of an adequate method or guidelines for estimating
- Complexity of the proposed application system
- Capabilities of the project team members/number of project team members
- Project team's experience with the application, the programming language and hardware
- Extent of programming and documentation standards