PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

Solution set

1 A. Software Quality?

_ *Conformance to explicitly stated functional, performance requirements & development standards and implicit characteristics that are expected by all softwares.*

_ Department of Defense defined a SQ as *"the degree to which the attribute of the softwares enable it to perform its intended end use"*

Views of Quality

_ Based on different characteristics of a product, it is been classified into number of views
or perspectives

_ These views may be considered based on external observers viewing towards a product
ex: user, supplier or client etc.,

_ These views are used to vary & conflict based on different types of peoples who is viewing the product

_ Ex: viewing persons

_ Project Manager

_ Business Analyst

_ Programmer

_ Quality Auditor

_ End user

_ Line Manager and Project Sponsor

1B. conflicting views are classified into 5 types

_ Transcendent view

_ Product based view

_ Value based view

_ Manufacturing view

_ User based view
7
_ Transcendent view
_ This view relates quality to excellence or elegance.
_ In large scale projects, it is expensive to innate high degree of excellence
_ Product based view
_ Quality ı Cost i.e., to build high quality in a product it costs high
_ This type of quality can be added to a product in 2 ways
_ Greater Functionality
_ High Quality Solution thro the use of QMS
_ Value based view
_ This view depends on what price that the customer can afford
_ Within the software development, this view can be used to add the quality to a product with the help of needed people, time & tools
_ Manufacturing view
_ This view measures the quality in terms of conformance to requirements
_ User based view
_ This view can be summarized as "Fitness for purpose"
_ i.e., the intended functionality for the product must be facilitated according to the User

2A. The Boehm model (1978)

It is to provide a set of well-defined, well-differentiated characteristics of software quality.

It is hierarchical in nature but the hierarchy is extended, so that quality criteria are subdivided.

According to the uses made of the system and they are classed into ‗ general' or ‗ as is' and the utilities are a subtype of the general utilities, to the product operation.

There are two levels of actual quality criteria, the intermediate level being further split into primitive characteristics which are amenable to measurement.
This model is based upon a much larger set of criteria than McCall' s model, but retains the same emphasis on technical criteria.

The two models share a number of common characteristics are,

The quality criteria are supposedly based upon the user‘s view.
The models focus on the parts that designers can more readily analyze.

Hierarchical models cannot be tested or validated. It cannot be shown that the metrics accurately reflect the criteria.
The measurement of overall quality is achieved by a weighted summation of the characteristics.
Boehm talks of modifiability where McCall distinguishes expandability from adaptability and documentation, understandability and clarity.

2B. THE GE MODEL (MCCALL, 1977 AND 1980) / (McCall Model)
-> This model was first proposed by McCall in 1977.
-> It was later revised as the MQ model, and it is aimed by system developers to be used during the development process.
-> In early attempt to bridge the gap between users and developers, the criteria were chosen in an attempt to reflect user‘ s views as well as developer‘ s priorities.
-> The criteria appear to be technically oriented, but they are described by a series of questions which define them in terms to non specialist managers.

The three areas addressed by McCall’ s model (1977):
Product operation: requires that it can be learnt easily, operated efficiently And it results are those required by the users.
Product revision: it is concerned with error correction and Adaptation Of the system and it is most costly part of software development.
Product transition: it is an important application and it is distributed processing and the rapid rate of change in hardware is Likely to increase.

McCall’ s criteria of quality defined
Efficiency is concerned with the use of resources e.g. processor time, storage. It falls into two categories: execution efficiency and storage efficiency.

Usability is the ease of use of the software.

Integrity is the protection of the program from unauthorized access.

Correctness is the extent to which a program fulfils its specification.
Reliability is its ability not to fail.
Maintainability is the effort required to locate and fix a fault in the program within its operating environment.

Flexibility is the ease of making changes required by changes in the operating environment.

Testability is the ease of testing the programs, to ensure that it is error-free and meet its specification.

Portability is the effort required to transfer a program from one environment to another.

Reusability is the ease of refusing software in a different context.

Interoperability is the effort required to couple the system to another system.

3A. The metrics cited depends to a very large extent upon just seven distinct measurable properties: readability, error prediction, error detection, complexity, and mean time to failure (MTTF), modularity, testability.

1. Readability as a measure of usability may be applied to documentation in order to assess how such documentation may assist in the usability of a piece of software.
2. Error prediction as a measure of correctness this measure is depends upon the stable software development environment.
3. Error detection as measure of correctness
4. Mean time to failure (MTTF) as a measure of reliability
5. Complexity as a measure of reliability the assumption underpinning these measures is that as complexity increases, so reliability decrease.
6. Complexity as a measure of maintainability is also indicative of maintainability.
7. Readability of code as a measure of maintainability has also been suggested as a measure of maintainability.

8. Modularity as a measure of maintainability increased modularity is generally assumed
to increase maintainability. Four measures have been suggested. Yau and Collofello

(1979) measured ― stability‖ as the number of modules affected by program modification. Kentger (1981) defined a four-level hierarchy of module types:
Control modules.
Problem-oriented modules.
Management modules for abstract data.
Realization modules for abstract data.
9. Testability as a measure of maintainability the ease and effectiveness of testing will
have an impact upon the maintainability of a product.

3B. Software Metrics
_ "Structuredness" is used to predict the maintainability, reliability & adaptability of the
software later in the lifecycle
_ Classified into 2 types:
_ Predictive Metrics
_ Descriptive Metrics
_ Predictive Metrics – It is used to make predictions about the software later in the life
cycle
_ Descriptive Metrics – It describes the state of the software at the time of the measurement
_ For Ex: reliability metric might be based upon the number of "system crashes" during the given period
_ Different authors have taken different approaches to metrics. Structuredness is measured
by questions such as:
o Have the rules for transfer of control between modules been
followed?(y/n)
o Are modules limited in size?(y/n)
o Do all modules have only one exit point ?(y/n)
o Do all modules have only one entry point?(y/n)
_ A well-structured program will produce positive answers to such questions.
_ McCall's approach is more quantities, using scores derived from equations such as

_ Where: n01 = no of modules containing one or zero exit points only ntot = total number
of modules
_ Generally, in this approach, scores are normalized to arrange between 0 and 1, to allow
for easier combination and comparison. This appears attractive, to give unjustified credibility to the results obtained. To validate this relationship and determine whether it is
a linear relationship or more complex in nature. It is also possible to validate whether
dependence of maintainability structured ness in identical to that of adaptability or reusability.
17

What makes a good metric?

_ Seven criteria for a good metric, after Watts (1987)
_ Objectivity the results should be free from subjective influences. It must not matter who
the measure is Reliability the results should be precise and repeatable. Validity the metric
must measure the correct characteristic.
_ Standardization the metric must be unambiguous and allow for comparison.
_ Comparability the metric must be comparable with other measures of the same Criterion.
_ Economy the simpler and therefore, the cheaper the measure is to use, the Better.
_ Usefulness the measure must address a need, not simply measure a property for its own
sake.
_ A further important feature is consistency.
_ Automation is also desirable.
_ Metrics available for each criterion (after Watts, 1987)

4A. An overall measure of quality

Much of the work in this area has been concerned with simple reduction of a set of scores
to a single _figure-of-merit‘.
Five such methods are detailed by Watts (1987) as part of the MQ approach.
1. Simple scoring: In this method, each criterion is allocated a score. The overall quality
is given by the mean of the individual scores.

2. Weighted scoring: This scheme allows the user to weight each criterion according to
how important they consider them to be. Each criterion is evaluated to produce a score
between 0 and 1. Each score is weighted before summation and the resulting figure reflects the relative importance if the different factors.

3. Phased weighting factor method: This is an extension of weighted scoring. A weighting is assigned to a group characteristics before each individual weighting is considered.

4. The Kepner- Tregoe method (1981): The criteria are divided into ‗ essential' and ‗ desirable' . A minimum value is specified for each essential criterion and any software failing to reach these scores is designated unsuitable. Suitable' software is then judged by use of the weighting factor method.

5. The Cologne combination method (Schmitz, 1975): This method is designed with comparative evaluation is mind. Using the chosen criteria, each product is ranked in
order.

4B. Gilb's Approach

_ *It is an iterative approach aiming to converge towards clear & measurable multidimensional objectives*

_ This approach makes use the concept of McCall & Boehm models

_ For each stage, a partial product can be viewed with user where a product will be evaluated to identify whether it meets the needs of user. If it does not satisfy errors has to
be identified & cleared out during the next iteration until a product gets satisfied by the
user.

_ 5 problem areas highlighted

_ Simple fact that the method is different

_ Need of training & re-training and associated costs

_ Need of effective management

_ Need to measure progress towards the ultimate goal

_ Picking up errors

_ With reference to Gilb's approach, product quality can be measured in terms of "Quality
Template"

_ It models quality in terms of Quality attributes & Resource attributes. This is because
quality of a product can be constrained by the available resources
20
_ "Quality Template" can be pictured as:
_ Quality Attributes
_ Workability
_ Availability
_ Adaptability
_ Usability

5A.THE GOAL QUESTION METRIC APPROACH

The Goal Question Metric (GQM) approach is based upon the assumption that for an organization to measure in a purposeful way it must first specify the goals for itself and its projects, then it must trace those goals to the data that are intended to define those goals operationally, and finally provide a framework for interpreting the data with respect to the stated goals. Thus it is important to make clear, at least in general terms, what informational needs the organization has, so that these needs for information can be quantified whenever possible, and the quantified information can be analyzed a to whether or not the goals are achieved. The approach was originally defined for evaluating defects for a set of projects in the NASA Goddard Space Flight Center environment.

The result of the application of the Goal Question Metric approach application is the specification of a measurement system targeting a particular set of issues and a set of rules for the interpretation of the measurement data. The resulting measurement model has three levels:

1. Conceptual level (GOAL): A goal is defined for an object, for a variety of reasons, with respect to various models of quality, from various points of view, relative to a particular environment. Objects of measurement are

 Products: Artifacts, deliverables and documents that are produced during the system life cycle; E.g., specifications, designs, programs, test suites.

- Processes: Software related activities normally associated with time; E.g., specifying, designing, testing, interviewing.
- Resources: Items used by processes in order to produce their outputs; E.g., personnel, hardware, software, office space.

2. Operational level (QUESTION): A set of questions is used to characterize the way the assessment/achievement of a specific goal is going to be performed based on some characterizing model. Questions try to characterize the object of measurement (product, process, resource) with respect to a selected quality issue and to determine its quality from the selected viewpoint.

3. Quantitative level (METRIC): A set of data is associated with every question in order to answer it in a quantitative way. The data can be

- Objective: If they depend only on the object that is being measured and not on the viewpoint from which they are taken; E.g., number of versions of a document, staff hours spent on a task, size of a program.
- Subjective: If they depend on both the object that is being measured and the viewpoint from which they are taken; E.g., readability of a text, level of user satisfaction.

THE GOAL QUESTION METRIC APPROACH

The Goal Question Metric (GQM) approach is based upon the assumption that for an organization to measure in a purposeful way it must first specify the goals for itself and its projects, then it must trace those goals to the data that are intended to define those goals operationally, and finally provide a framework for interpreting the data with respect to the stated goals. Thus it is important to make clear, at least in general terms, what informational needs the organization has, so that these needs for information can be quantified whenever possible, and the quantified information can be analyzed a to whether or not the goals are achieved. The approach was originally defined for evaluating defects for a set of projects in the NASA Goddard Space Flight Center environment.

The result of the application of the Goal Question Metric approach application is the
specification of a measurement system targeting a particular set of issues and a set of rules
for the interpretation of the measurement data. The resulting measurement model has three
levels:

1. Conceptual level (GOAL): A goal is defined for an object, for a variety of reasons,
with respect to various models of quality, from various points of view, relative to a particular environment. Objects of measurement are

- Products: Artifacts, deliverables and documents that are produced during the system life cycle; E.g., specifications, designs, programs, test suites.
- Processes: Software related activities normally associated with time; E.g., specifying, designing, testing, interviewing.
- Resources: Items used by processes in order to produce their outputs; E.g., personnel, hardware, software, office space.

2. Operational level (QUESTION): A set of questions is used to characterize the way
the assessment/achievement of a specific goal is going to be performed based on some characterizing model. Questions try to characterize the object of measurement (product, process, resource) with respect to a selected quality issue and to determine its quality from the selected viewpoint.

3. Quantitative level (METRIC): A set of data is associated with every question in
order to answer it in a quantitative way. The data can be

- Objective: If they depend only on the object that is being measured and not on the viewpoint from which they are taken; E.g., number of versions of a document, staff hours spent on a task, size of a program.
- Subjective: If they depend on both the object that is being measured and the viewpoint from which they are taken; E.g., readability of a text, level of user satisfaction.

5B. Quality Engineering

the activity consisting of the cohesive collection of all tasks that are primarily performed to ensure and help continually improve the quality of an endeavor‘s process and work products

Goals
The typical goals of quality engineering are to:
Ensure that the necessary levels of quality are achieved.
Make the achievement of quality predictable and repeatable.
Minimize endeavor, organizational, and personal risks due to poor quality.
Objectives
The typical objectives of quality engineering are to:
Define what quality means on the endeavor in terms of a quality model defining quality factors and quality sub factors.
Plan the quality tasks including helping the requirements team determine and specify the quality requirements and associated quality factors (attributes) and quality metrics.
Assure the quality of the **process** used by the endeavor.
Thus, quality assurance is concerned with fulfilling the quality requirements and achieving the quality factors of the endeavor‘s process.
―Are we building the products right?‖
Control the quality of the **work products** delivered during the endeavor.
Thus, quality control is concerned with fulfilling the quality requirements and achieving the quality factors of the endeavor‘s work products.
―Are we building the right products?