

# Chapter 13: Software Quality

Project Management

Afnan Albahli

# Quality Concerns

- We need to define what qualities we require of a system.
- We also need to judge whether a system meets our quality requirements (this needs measurements).
- We may want to assess the likely quality of the final system while it is still under development.
- We also will need to make sure that the development methods would produce the quality needed.
  - A potential customer might check that the suppliers were using the best development methods.

# The Importance of Software Quality

Quality should be the concern of all the producers of goods and services, however the special characteristics of SW create special demands.

- Increasing criticality of SW.
  - The intangibility of SW make it difficult to know that a project task was completed satisfactory.
  - Accumulating errors during SW development.
- 
- For these reasons quality management is an essential part of effective overall project management.



# The Place of Software Quality in Project

- Step 1: Identify project scope and objectives: some objectives could relate to the qualities of the application to be delivered.
- Step 2: Identify project infrastructure, within this step, activity 2.2 identifies installation standards and procedures. Some of those will be about quality.
- Step 3: Analyze project characteristics, within this step activity 3.2 Analyze other project characteristics including quality based ones.
- Step 4: identify the products and activities of the project.
- Step 8: Review and publicize plan, at this stage the overall quality aspects of the project plan are reviewed.

# Defining Software Quality

- Every system has:
- Functional requirements: what is the system is to do.
- Resource requirements: allowable cost.
- Quality requirements: how well the system is to operate.

Example of quality requirements required by the users are:

Usability.

Reliability.

Defining quality is not enough, in order to judge whether a system meets our requirements, we need to be able to measure its qualities.

# Quality Measures

- In order to judge whether a system meets our requirements ,we need to be able to measure its qualities.
- A good measure must relate the number of units measured to the maximum possible.

Example:

- For the reliability, it could be argued that the less the errors in a program the more reliable it is.
- A measure of faults per thousand of lines of code is more useful than the total faults in a program.

# Quality Measures (cont'd)

- Quality measures are important because they are a way to know when we have been successful to meet a quality.

Measures can be:

- Direct measures: where we can measure the quality directly.
- Indirect measures: where the thing being measured is not the quality itself, but an indicator that the quality is present.
- e.g. the number of enquirers by users received by a help desk about how one operates a particular SW application might be an indirect measurement of its usability.



# Quality Specifications

- When there is concern about the need for a specific quality characteristic in a SW product then a quality specification with the following minimum details should be drafted:
  - Definition “description” 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, where the product would be rejected if had a lower value.
  - Target range: the range of values within which it is planned the quality measurement should lie.
  - Now: the value that applies currently.

# Reliability

- Software reliability is defined as: The ability of a system or component to perform its required functions under stated conditions for a specified period of time.
- The following are some measurements used to measure the reliability of a product:
  - Availability: the percentage of a particular time interval that a system is usable.
  - Mean time between failures: the total service time divided by the number of failures.

# Reliability

- Associated with reliability is maintainability, which is: how quickly a fault, once detected, can be corrected.  
*“this is from the users point of view”*
- *“the SW development managers” will be concerned about the effort involved.*
- *A key component of this is changeability, which is: the ease with which the software can be modified.*
- *However before an amendment can be made, the fault has to be diagnosed.*
- *Maintainability can therefore be seen as changeability plus a new quality, analysability, which is the ease with which causes of failure can be identified.*

# ISO 9126

- Stands for “International Organization for Standardization.”
- There was a lack of agreed on definitions for the qualities of a good software.
- This standard was introduced in 1991 to tackle the question of the definition of **software quality**.

# ISO 9126 (cont'd)

ISO 9126 defines six major software quality characteristics:

- **Functionality:** covers the functions that a software product provides to satisfy user needs.
- **Reliability:** refers to the capability of the software to maintain its level of performance.
- **Usability:** which relates to the effort needed to use the software.
- **Efficiency:** which relates to the physical resources used when the software is executed.
- **Maintainability:** relates to the effort needed to make changes to the software.
- **Portability:** relates to the ability of the software to be transferred to a different environment.