

COURSE NAME

SOFTWARE QUALITY
AND TESTING

CSC 4133

(UNDERGRADUATE)

CHAPTER 2

SOFTWARE QUALITY

QUALITY PERSPECTIVES

❑ Subject of SQAT: People's perspectives

- External/Consumer: customers and users
- Internal/Producer: developer, testers, and managers
- Other: 3rd party, indirect users (email notification)

❑ Objects of SQAT:

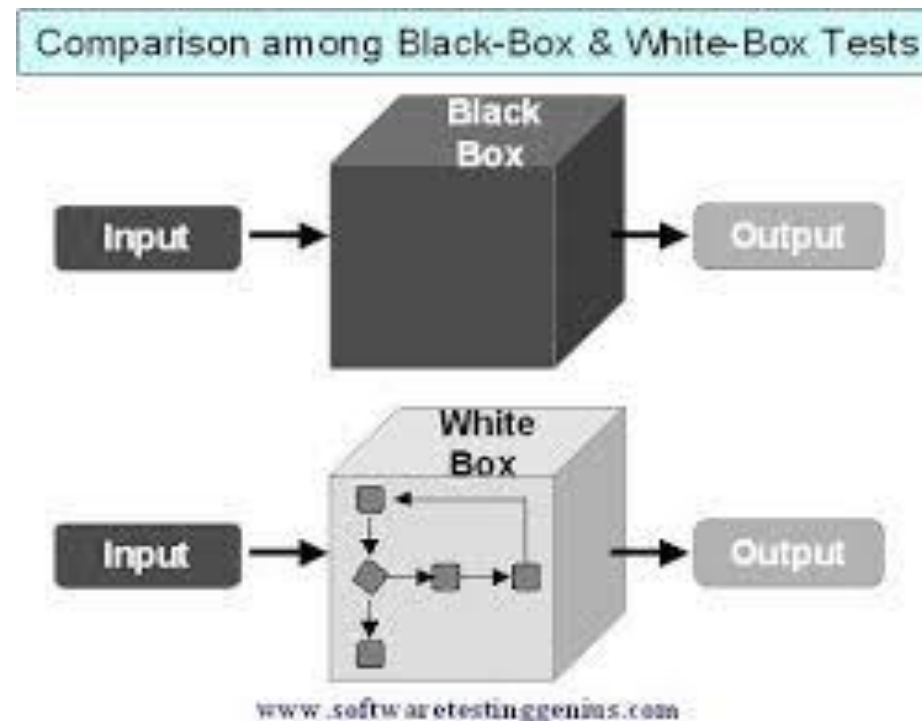
- Software products, systems, and services

❑ External View

- mostly sees a software system as a black box, where one can observe its behavior but not see through inside

❑ Internal View

- mostly sees a software system as a white box, or more appropriately a clear box, where one can see what is inside and how it works



VIEWS IN SOFTWARE QUALITY (KITCHENHAM AND PFLEEGER)

1. Mystical view (misconception)

- Quality is something that can be recognized through experience, but not defined in some tractable form
- A good quality object stands out, and it is easily recognized

User view – fitness for the purpose.



2. User view

- Quality concerns the extent to which a product meets user needs and expectations
- A product is of good quality if it satisfies a large number of users
- It is useful to identify the product attributes which the users consider to be important
- This view may encompass many subject elements, e.g. usability, reliability, efficiency, etc.

VIEWS IN SOFTWARE QUALITY

3. Manufacturing view

- Conformance to process standards/requirements
- Quality is seen as conforming to requirements leads to consistency in products
- Any deviation from the requirements is seen as reducing the quality of the product
- Products are manufactured “right the first time” so that the cost can be reduced (very low changing possibility and without re-work)
- Product quality can be incrementally improved by improving the process
- The CMM and ISO 9001 models are based on the manufacturing view



VIEWS IN SOFTWARE QUALITY

4. Product view

- If a product is manufactured with good internal properties, then it will have good external properties

5. Value-based view

- Customers' willingness to pay for a software
- Value-based view represents the merger of two concepts: excellence and worth
- Quality is a measure of excellence, and value is a measure of worth
- The value-based view makes a trade-off between cost and quality



WHY MEASURE QUALITY?

- ❑ Measurement allows us to have a quantitative view of the quality concept
- ❑ What are the reasons for developing a quantitative view of quality?
 - Baseline:
 - Measurement allows us to establish baselines for qualities to achieve (Usability- extract all the information within 20 minutes from a website)
 - Quality improvement based on cost:
 - Organizations make continuous improvements in their process models and an improvement has a cost associated with it. Measurement is key to process improvement
 - Know the present level for future planning:
 - The needs for improvements can be investigated after performing measurements

SOFTWARE QUALITY FACTOR

- ❑ Software Quality in terms of quality factors and quality criteria
- ❑ McCall's Quality Factors and Criteria

Quality Factors

- A quality factor represents the behavioral characteristic of a system
- Examples
 - Correctness
 - Reliability
 - Efficiency
 - Performance

SOFTWARE QUALITY FACTOR

❑ EFFICIENCY

- The amount of computing resources and code required by program to perform a function (e.g. multi-tasking)
- *At least 25 percent of the processor capacity and RAM available to the application shall be unused at the planned peak load conditions.*



❑ INTEGRITY / SECURITY

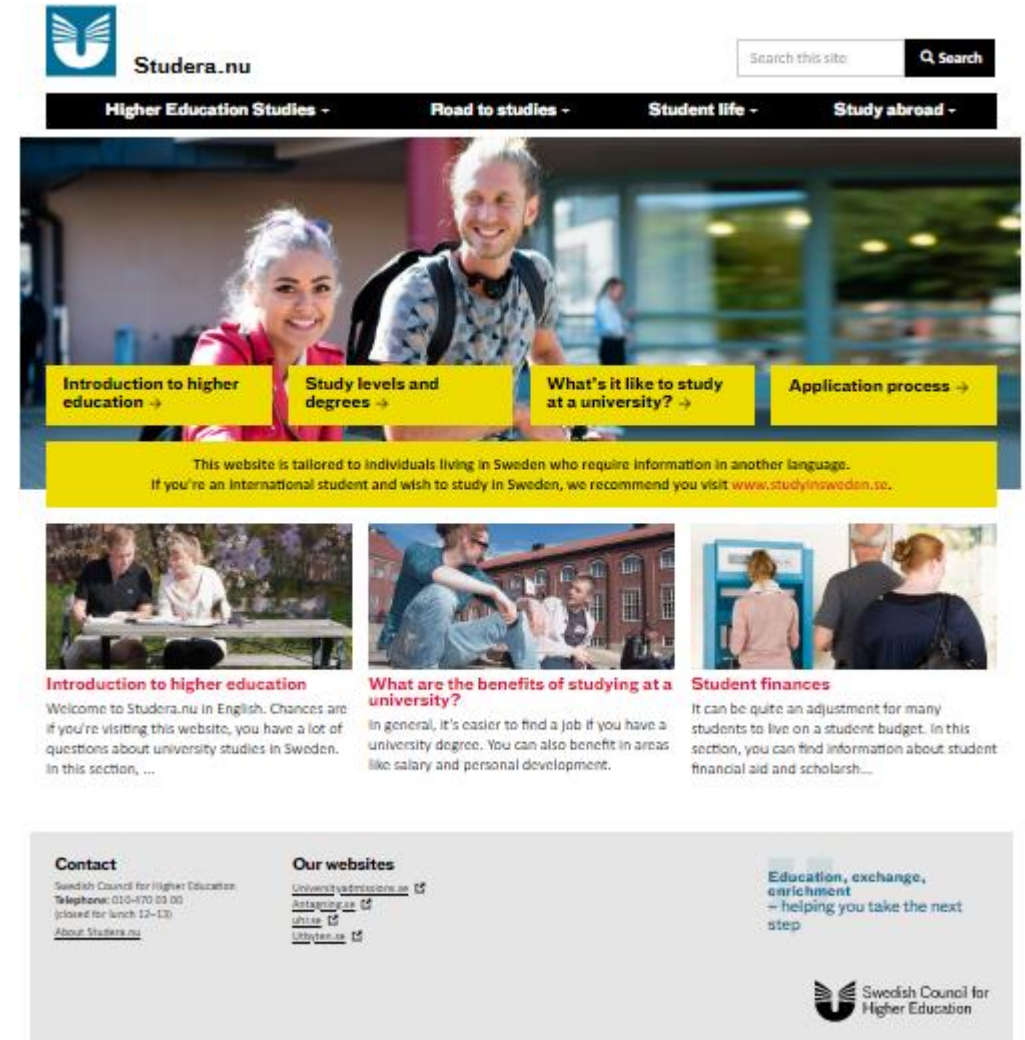
- The extent to which access to a software or data by unauthorized persons can be controlled.
- **CHOICE POINT** security issue of unprotected data on a tape
- *Only users who have Auditor access privileges shall be able to view customer transaction histories.*

SOFTWARE QUALITY FACTOR

□ USABILITY

- The effort required to learn, operate, prepare input, and interpret output of a program
- Time required (e.g. 30 minutes maximum) to extract information regarding higher studies application in Sweden from the website <http://studera.nu>

- *A trained user shall be able to submit a complete request for a chemical selected from a vendor catalog in an average of four and a maximum of six minutes.*



SOFTWARE QUALITY FACTOR

□ RELIABILITY AND CORRECTNESS

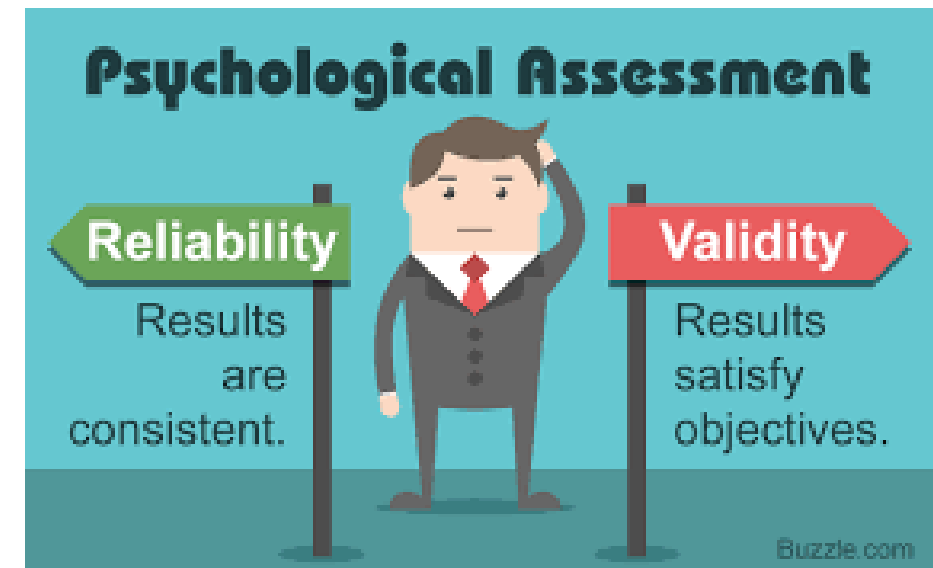
- The extent to which a program can be expected to perform its intended functions with required precision (user's objectives/satisfaction).
- The probability of the software executing without failure for a specific period of time is known as reliability

Example: *No more than five experimental runs out of 1000 can be lost because of software failures*

Reliability & Validity



Evaluating the validity and reliability of research data



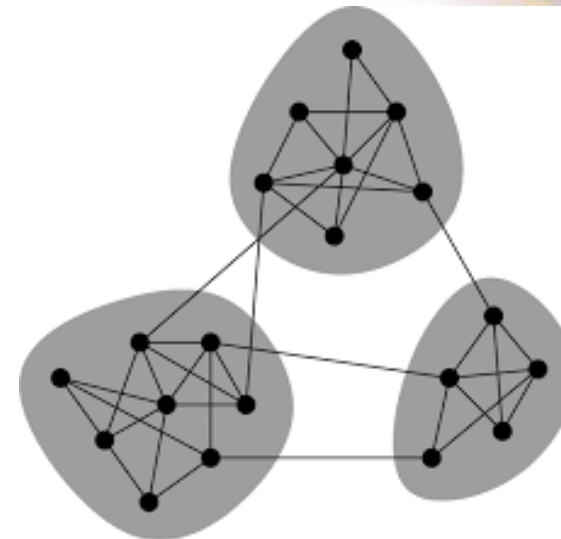
SOFTWARE QUALITY CRITERIA

Software Quality Criteria

- A quality criterion is an attribute of a quality factor that is related to software development
- Example: Modularity, testability, maintainability, reusability, etc.

□ MODULARITY

- Modularity is an attribute of the architecture of a software system
- A highly modular software allows designers to put cohesive components in one module, thereby increasing the maintainability of the system
- Compatibility among the module integration



❑ MAINTAINABILITY SOFTWARE QUALITY CRITERIA

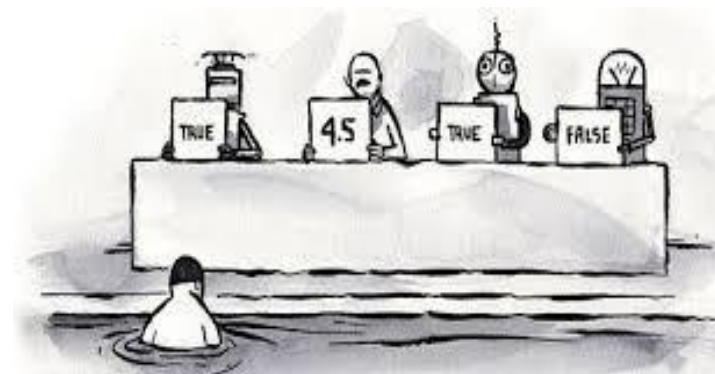
- The effort required to locate and fix a defect in an operational program
- *A maintenance programmer shall be able to modify existing reports to conform to revised chemical-reporting regulations from the federal government with 20 labor hours or less of development effort.*

❑ TESTABILITY

- The effort required to program to ensure that it performs its intended functions.
- *The maximum cyclomatic complexity* of a module shall not exceed 20*

❑ FLEXIBILITY

- The effort required to modify an operational program.
- *A maintenance programmer who has at least six months of experience supporting this product shall be able to make a new copy output available to the product, including code modifications and testing, with no more than one hour of labor.*



SOFTWARE QUALITY CRITERIA

□ PORTABILITY

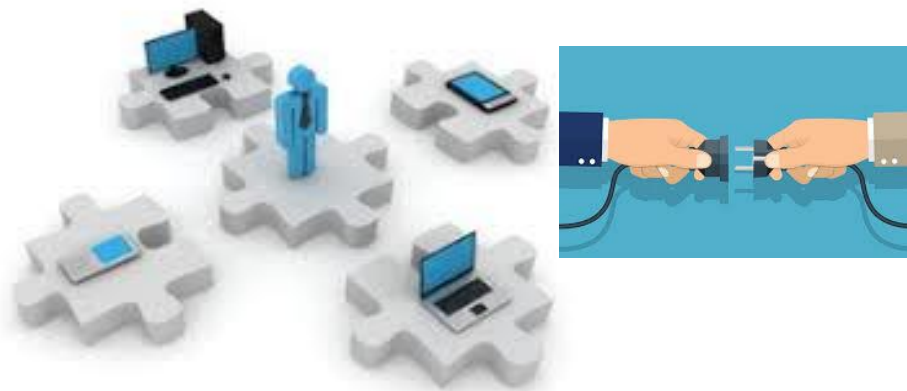
- The effort required to transfer a program from one hardware and/or software environment to another

□ REUSABILITY

- The extent to which parts of a software system can be reused in other application
- *The chemical structure input functions shall be designed to be reusable at the object code level in other applications that use the international standard chemical structure representations.*

□ INTEROPERABILITY

- The effort required to couple one system with another (Biometric SIM registration, Blockchain)
- *The Chemical Tracking System shall be able to import any valid chemical structure from the ChemiDraw (version 2.3 or*



ISO-9126 QUALITY FRAMEWORK

- ISO (International Organization for Standardization)

- **ISO-9126:**

- The mostly influential one in the software engineering community today
- Provides a hierarchical framework for quality definition, organized into quality characteristics and sub-characteristics
- Six top-level quality characteristics, each associated with its own exclusive (non-overlapping) sub-characteristics

Quality characteristics

- **Functionality:** what is needed
- **Reliability:** functions correctly
- **Usability:** effort to use
- **Efficiency:** resource needed
- **Maintainability:** correct/improve/adapt
- **Portability:** from one environment to another

QUALITY EXPECTATIONS

□ External/Consumer expectations

- “good enough” for the PRICE
 - Fit-for-use: doing the “right things”
 - Conformance: doing the “things right”

□ Expectations for different software

- General: functionality & reliability
- Usability: GUI/end-user/web/etc.
- Safety: safety-critical systems (autopilot)

□ Internal/Producer expectations

- “good enough” for the COST
 - Mirror consumer side
 - Functionality & correctness via V&V
- Service related: maintainability
- Interfacing units: interoperability
- 3rd party: modularity (outsource)

REFERENCES

- ❑ Software Testing And Quality Assurance – Theory and Practice - Kshirasagar Naik & Priyadarshi Tripathy
- ❑ Software Quality Engineering: Testing, Quality Assurance and Quantifiable Improvement - Jeff Tian