SE 4367 Homework #1, Quality

Write a 1-2 page description of what “software quality” means.

- **Is there a simple definition for what quality software is?**
- **What attributes make up “quality”?**
- **Is quality truly important for the success of software products?**
- **How does testing contribute to a quality product?**

The article by Glass in the References folder on eLearning on “Defining Quality Intuitively” should be helpful.

Grading Rubric

Less than half-a-page, -50 points

More than 2 pages, -20 points per page

Proper cites and references, -50 points

- **see the handout**

Lack of thinking about, or understanding of, quality, -30 points

Missing class, assignment, name in filename or inside file, -5 points each

Formatting Submissions

In the file name, include:

- **class**
- **assignment identifier**
- **your name (or team's name)**
  - **e.g., se4367a01jdoe**

In the file (or hardcopy) submitted, include the class, assignment, and name information at the top.

Minus 5 points per violation. Potentially 30 points off for formatting mistakes!

Previous Observations

Some people did not cite the Glass article

- **you did not need to if you did not use it**
- **if you did read it (and quoted from it), you should have cited it as a reference!**

If you cite the Glass article in your discussion, you should reference it appropriately. Something like

- **R. Glass, “Defining Quality Intuitively,” IEEE Software, Vol. 15, No. 3, May/June 1998, pp. 103-104,107.**

Provide at least enough information to find the article!

Reference IEEE Software not just IEEE in your reference.

Do not reference IEEE Xplore with a URL for a reference for the Glass article!

- **URLs are not references unless referring to a website**

I assumed you did not read the Glass article as recommended if you did not cite it.

“Reinventing” definitions of quality like Barry Boehm’s formula from 40 years ago or discussions of Therac-25 without any citations is the kind of thing that can get you in talks with Judicial Affairs about plagiarism…

If you have a reference, you should cite it in the body of the report.

- should have both cite, e.g., [Glass 1998], and reference
- sometimes the reference was missing
- frequently the citation was missing

Actually submit the homework, not the assignment…

Use something like HW01 or A01, not “Software Quality” as the assignment identifier (both is fine too).

Do not submit files in an oddball format.

Please do NOT put your UTD id in the filename or inside your submissions.

Some people did not follow the submission format requirements (class, assignment, name)

Traditional Definitions of Quality

Quality is conformance to requirements

- **Phillip Crosby**

Quality is defined by the customer

- **W. Edwards Deming**

Quality is fitness for use.

- **Joseph Juran**

Nine dimensions of quality: performance, features, functionality, safety, conformance, reliability, durability, service, and aesthetics

- **David Garvin (1987)**

“The traditional definitions are wrong. Quality is not about user satisfaction, product conformance, or costs and schedules—nor is it solely about defects.

Instead, there is a well-defined, intuitive relationship between quality and those other product traits, one that clearly distinguishes among them.

Further, the detail-level definition of software quality concerns the attribute set that each product should have, a collection that must be prioritized differently for different project types.”

- **(Glass 1998)**

Glass on Quality

User satisfaction = compliant product + good quality + delivery within budget and schedule

- **it makes quality an attribute of the thing actually being defined: user satisfaction**
- **quality and user satisfaction are related but not equivalent**

Quality equals the totality of features and characteristics of a product or service that bear on its ability to satisfy stated and implied need.

- ISO 8402: 1986

Quality = portability + reliability + efficiency + human engineering + understandability + modifiability + testability (Boehm 1975)

- different software products call for different “ilities” mixes
- no universal definition of quality at the detail level, that is independent of the product in question, exists

Software Architecture in Practice
Quality Attributes

Availability

Interoperability

Modifiability

Performance

Security

Testability

Usability

ISO/IEC 25010 SQuaRE

Functional suitability

- functional completeness
- functional correctness
- functional appropriateness

Performance efficiency

- time behavior
- resource utilization
- capacity

Compatibility

- coexistence
- interoperability

Usability

- appropriateness recognizability
- learnability
- operability
- user error prediction
- user interface aesthetics
- accessibility

Reliability

- maturity
- availability
- fault tolerance
- recoverability

Security

- confidentiality
- integrity
- nonrepudiation
- accountability
- authenticity

Maintainability

- modularity
- reusability
- analyzability
- modifiability
- testability

Portability

- adaptability
- installability
- replaceability