Singapore Institute of Management
BSc in Computing & Information Systems
CS 210 Tutorial 7

---

1. How do you achieve software Quality Assurance in a project? Briefly describe. What is the purpose of performing Quality Assurance in the project (especially as it involves effort and time to build QA into projects)?

Answer :

QMS has goals

- *Provision and management of resources (people and tools) for SQA activities*
- *Development of standards and procedures*
- *Auditing of projects to ensure standards are being followed*
- *Review of the QMS itself*

2 products produced in QMS

- *Quality Manual – lays down standards, procedures and guidelines*
- *Quality Plan – spells out on per project basis, which elements of the manual and which QCA are going to be used*

Achieve QA through

- *Quality management system*
  - *Quality Manual – lays down standards, procedures and guidelines*
  - *Quality Plan – spells out on per project basis, which elements of the manual and which QCA are going to be used*
- *Multi-tiered testing strategies*
- *Configuration management especially with respect to documentation*
- *Mechanisms to measure and report*
- *Through QCA to check compliance with standards*
  i. *Walkthroughs – to check that code and design are in agreement*
  ii. *Acceptance tests – check that a function has been correctly implemented*
  iii. *Profiling – check which statements are being covered by a test and that there are no undetected efficiency bottlenecks*
  iv. *Checkouts- to obtain the stamp of approval of a senior s/w engineer that the work of a junior member meets the desired standards*
  v. *Software Reviews – to point out possible improvements on a single person's (or team's) work, confirm improvements is not possible or needed in parts of the system, achieve more*

uniformity, predictability in the technical work so as ot make the technical process more manageable.

QA is essential to ensure the production of high-quality software. This is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confidence that product quality is meeting its goals. The data provided through QA identifies problems and it is management's responsibility to address problems and apply resources to resolve quality issues.

2. Can a program be correct and still not be reliable? Explain.

Answer :

Yes a program can be correct and still not be reliable. This is because reliability is defined as the "probability of failure-free operation of a computer program in a specified environment for a specified time".

Thus program X is estimated (for e.g.) to have a reliability of 0.96 over 8 elapsed hours of elapsed processing time (execution time), it is likely to operate correctly without failures 96 times out of 100.

Thus a program can be operating correctly sometime but is unreliable as it fails quite often.

3. Describe the different types of software technical metrics that are available for measuring software quality.

.

Answer :

*For the **analysis** model, metrics focuses on function, data and behaviour – the 3 components of the analysis model. The types of technical metrics available include :*

a) *Function based metric **-** use the function point as a normalizing factor or as a measure of the "size" of the specification*
b) *Bang metric - used to develop an indication of software "size" by measuring characteristics of the data, functional and behavioral models*
c) *Specification metric - used as an indication of quality by measuring number of requirements by type*

Each providing a quantitative means for evaluating the analysis model.

***Metrics for design**, considers the architecture, component-level design and interface design issues.*

***Architectural** design metrics consider the structural aspects of the design model.*

a) *HK metric: architectural complexity as a function of fan-in and fan-out*
b) *Morphology metrics: a function of the number of modules and the number of interfaces between modules*

***Component-level design** metrics provide an indication of module quality by establishing indirect measures for cohesion, coupling and complexity.*

a) *Cohesion metrics: a function of data objects and the locus of their definition*
b) *Coupling metrics: a function of input and output parameters, global variables, and modules called*
c) *Complexity metrics: hundreds have been proposed (e.g., cyclomatic complexity*

***Interface design** metrics provide an indication of layout appropriateness for a GUI. It is a function of layout entities, the geographic position and the "cost" of making transitions among entities*

***Halstead's Software Science provides a set of metrics at source code level :** a comprehensive collection of metrics all predicated on the number (count and occurrence) of operators and operands within a component or program*

4. You are the QA Manager and have been tasked to look at a project to determine for the project
   a) Are the modules well designed?
   b) Are the components well tested?
   What types of metrics will you use for each of the above? Justify.

   Answer :

   a) A mixture of metrics :
   - *Cohesion metrics – to check the cohesiveness of the module*
   - *Coupling metrics – to determine the connectedness of a module to other modules*
   - *Complexity metrics - To determine the complexity of the module and whether there is a need to simplify the module*

   b) Cyclomatic complexity – to determine the basis path testing and hence the exhaustiveness of the unit testing.

5. What is the Pareto principle? There are defects discovered in the payroll system. The defects are as classified as follows (with corresponding severity – 1 being most severe and 6 least severe):

Cause	No. of Errors	% of Total No of Errors	Severity
Incomplete or erroneous specification	22	24	2
Problems in communication with users	31	33	4
Problems with Human-computer interface	15	16	3
Incomplete or error in documentation	14	15	6
Data representation error	3	3	1
Miscellaneous	8	9	5
Totals	**93**	**100**	

Based on the severity of the defects, does the Pareto principle hold in this case? Support your answer.

Answer :

Pareto principle states that most (80%) of the defects can be traced back to a vital few (20%) causes.

No, the Pareto principle does not hold in this case. This is because 85% of the defects are caused by 83% of causes. The 85% of the errors are due to all causes except Incomplete or error in documentation.