p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 07 Issue 12
December 2020

The Importance Software Metrics In Statistical Tools For Any Application

Dr. L. Maria Arulraj Assistant Professor Statistics, Periyar University College of Arts & Science, Mettur Dam, Salem Dt.

ABSTRACT:

In this paper various aspects role and importance of software metrics in quality of a products has been analysed and discussed

Quality Assurance (QA):

Assuring the quality of the products through audits of the manufacturing processes is known as Quality Assurance (QA). It also assures that the products are manufactured as per established guidelines and gives inputs (Metrics / measures) to the processes for the continuous improvement.

Quality Assurance three stages:

- Design stage
- Manufacturing stage
- Field observation

Total Quality assurance = Quality of design + Quality of manufacture conforming to design + Quality of performance.

The two basic concepts in quality assurance:

i) Process control: To ensure that the proportion of defective item in manufacture product is not too large .it is achieved through Control charts.
ii) Product control: Controlling the quality of product by critical examination at strategic points and it is achieved through Sampling Inspection plan

Software quality: It can be defined as the totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs. Based on requirements it is classified as follows,

Functional software quality: The quality of Software that combines Low defect rates and high level of user satisfaction. This software should meet all functional requirements of user and adhere to international standards.

Structural Software quality: The Software quality that exhibits a robust architecture and comparable in a multi-tier environment without failures or degradation. It has low cyclamatic complexity levels.

Metrics: The measurements which satisfies the validity, Reliability & Robustness of any product.

Software Quality metrics: A measure of some property of a piece of software or its specifications. When applied to the software product, a software metric measures (or quantifies) a characteristic of the software.

Some common software metrics:

1. Sourcelines of code.
2. Cyclomatic complexity is used to measure code complexity.
3. Function point analysis (FPA), is used to measure the size (functions) of software.
4. Bugs per lines of code.
5. Code coverage, measures the code lines that are executed for a given set of software tests.
6. Cohesion, measures how well the source code in a given module work together to provide a single function.
7. Coupling, measures how well two software components are *data* related.

Few important properties of the software metrics:

1. They are all measurable,
2. They can be quantified.
3. They are all related to one or more software quality Characteristics.

Significance of SQC in research

when statistical techniques are employed to control, improve and maintain quality or to solve quality problems.. It is called “Statistical Quality Control”.

Metrics should be

1. Simple
2. Objective
3. Easily obtainable
4. Valid
5. Robust

- Statement of the problem

Assuring Good quality of the software is more vital. The study deals to ensure whether the software has attained the desirable characteristics, various metrics. In this paper an attempt has been made on different metrics associated with software quality assurance.

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 07 Issue 12
December 2020

The present Research mainly discusses the usages and its significance of metrics in maintaining the software quality and standards. In general, this research made an attempt to analyse and explores the metrics relevant to Software quality assurance.

OBJECTIVES

- To deal the usage of metrics in software Quality assurance
- To study about the factors, determines the extent of Quality Assurance.
- To focus the major areas in using metrics in managing Good Quality Assurance.
- To deal the usage of metrics in I.T industry in ensure Good Quality Assurance

METHODOLOGY

1. Collection of primary Data from Software Developing Companies
2. Collecting of Secondary Data from journals & http Sources
3. Segregation of Data as per the objective/requirement
4. Tabulation of segregated Data
5. Analysis of Data with Statistical Tools
6. Tabulation of summary of Findings.
7. Conclusion.

Statistical Tools used

1. Checklist
2. Pareto
3. Run Charts
4. Scatter Diagram
5. Control Chart

Limitations

1. The primary data pertaining to study are collected from the software developing companies. Due to regulation of the companies full names of the tested software could not be listed out.
2. The secondary data are collected from the sources like journals/ books/ periodicals, dissertations. Hence, there may be exclusion of certain vast items of metrics used in maintaining software quality.
3. The study covers only metrics related to Quality Assurance of Application/Utility software which are associated in deciding the quality of the product.

4. Only few important metrics only are taken in this due to the time constraints.
Software Standards
Software standards can be split into two categories:

(a) Product Standards
(b) Process Standards

Software Standards

Software standards can be split into two categories:
(a) Product Standards
(b) Process Standards

PRODUCT STANDARDS

Product standards provide a frame work from which to build products and demand a rigorous approach for specification and design of the product which can then be tested and evaluated. Product standards include specifications for languages such as FORTRAN, COBOL, C,

Process Standards

Process standards are generally written as policy statements and recommendations on how to conduct certain process and do not dictate specific methodology. For example, ISO 9000 (Quality Management and Quality Assurance Standards) refers to a family of international standards that addresses software quality processes and define a process certification.

In view of above categories, a standard has been defined as an approved, documented and available set of criteria used to determine the adequacy of an action (process standard) or object (product standard).

SOFTWARE QUALITY ASSURANCE (SQA)

Software Quality properties include cyclomatic complexity, cohesion, number of function points, lines of code, and many others. When we examine an item based on its measurable characteristics, two kinds of quality may be encountered: quality of design and quality of conformance.

Quality of design refers to the characteristics that designers specify for an item. The grade of materials, tolerances, and performance specifications all contribute to the quality of design. As higher-grade materials are used, tighter tolerance and greater levels of performance are specified, the design quality of a product increases, if the product is manufactured according to specifications.

Quality of conformance is the degree to which the design specifications are followed during manufacturing. Again, the greater the degree of conformance, the higher is the level of quality of conformance.User satisfaction = compliant product + good quality + delivery within budget and schedule.

Quality Control

Quality control involves the series of inspections, reviews, and tests used thought the software process to ensure each work product meets the requirements placed upon it. Quality control includes a feedback loop to the process that created he work product. The combination of measurements and feed back allows one to tune the process when the work products created fail to meet their specifications. This approach views quality control as part of the manufacturing process.

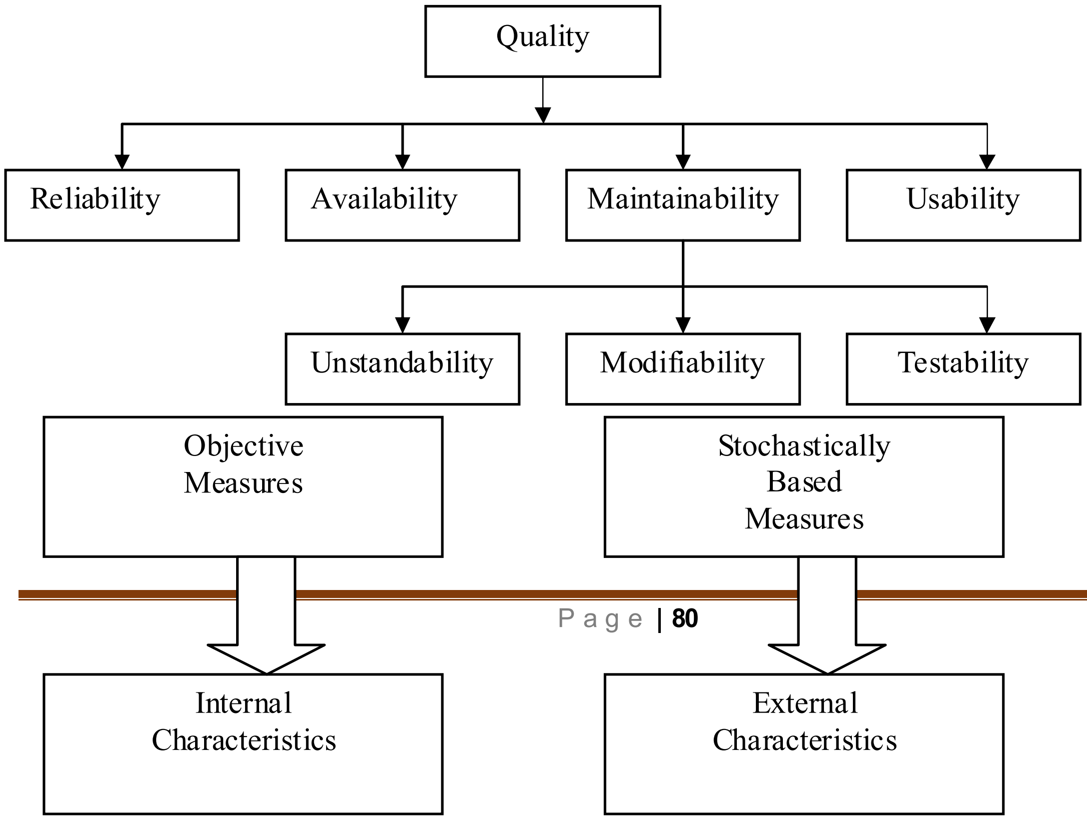
SOFTWARE QUALITY METRICS

Internal Characteristics

It describe structural complexity common measures of structural complexity are metrics for size, data structure, control flow complexity and inter module coupling. It is also influenced by the psychological characteristics of the analysis /designer/ programmer, the complexity of the problem, and complexities inherent in the selected representational medium.

The external characteristics are not clearly defined in software engineering. It focuses on a range of quality factors, from maintainability reliability, establish modifiability of the resultant software system.

Quality = reliability + availability + understandability + Modifiability + Stability + Usability.



Desirable properties of software metrics

The value must be computed in a resize manner

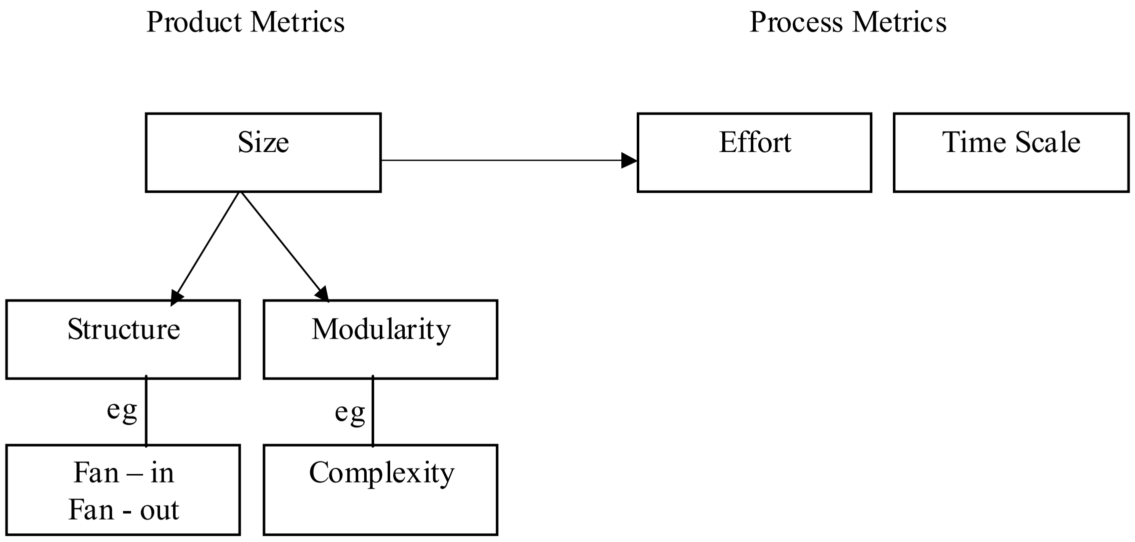
It must be intuitive and posses some form of internal validity.

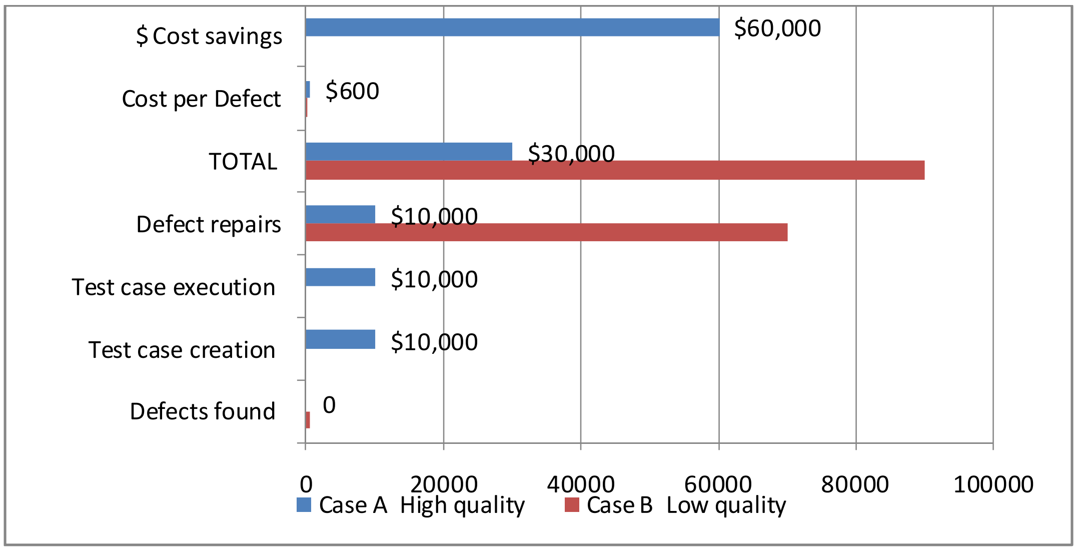
It must be robust that is reliable.

Must provide information to gain better understanding improving design.

Product and process metrics

This resource is concerned about measuring internal and external characteristics, but those characteristics may be two different natures either product or process. A product measure is a "Snap shot" at a particular point in times. It can be snap shot of the early design or of the code today, which one can them desire some code metrics.





Graph 1. .COMPARISON OF COST PER DEFECT

(IT AFFECTS QUALITY)

Table . LINES OF CODE HARM HIGH-LEVEL LANGUGES

Metrics	**Case A**	**Case B**
Software Language	**JAVA**	**C**
KLOC	50	125
Function points	1,000	1,000
Code defects found	500	1,250
Defects per KLOC	10.00	10.00
Defects per FP	0.5	1.25
Defect repairs	$70,000	$175,000
$ per KLOC	$1,400	$1,400
$ per Defect	$140	$140

$ per Function Point	$70	$175
$ cost savings	$105,000	$0.00

POOR QUALITY RESULTS: < 25% SUCCESS RATE

- Testing as only form of defect removal
- Informal Testing and uncertified test personnel
- Testing only by developers; no test specialists

SOFTWARE QUALITY OBSERVATIONS

• Individual programmers -- Less than 50% efficient in finding bugs in their own software.

• Normal test steps -- often less than 75% efficient (1 of 4 bugs remain)

• Design Reviews and Code Inspections – often more than 65% efficient; have topped 90%

SOFTWARE QUALITY AND PRODUCTIVITY

The most effective way of improving software productivity and shortening project schedules is to reduce defect levels.

Defect reduction can occur through:

1. Defect prevention technologies

✓ Structured design
✓ Structured code
✓ Use of inspections, static analysis
✓ Reuse of certified components

SUMMARY OF FINDINGS

✓ No single quality method is adequate by itself.
✓ Formal inspections, static analysis, models are effective
✓ Inspections + static analysis + testing > 97% efficient.
✓ Defect prevention + removal best overall
✓ QFD, models, inspections, & six-sigma prevent defects
✓ Higher CMMI levels, TSP, RUP, Agile, XP are effective
✓ Quality excellence has ROI > $15 for each $1 spent
✓ High quality benefits schedules, productivity,

- ✓ **<u>RESOURCES:</u>**
- ✓
- ✓ **Websites**
- ✓ http://www.asq.org/ American Society for Quality.
- ✓ http://www.asq-software.org/ Software Division of the American Society for Quality.
- ✓ http://www.sixsigmaforum.com/ Six Sigma Forum established by the American Society for Quality (ASQ).
- ✓ http://www.isixsigma.com/ A Web portal devoted to Six Sigma programs.
- ✓ http://www.qualitydigest.com/ Quality Digest. A magazine.
- ✓ http://www2.umassd.edu/SWPI/ Software Process Resource Collection.
- ✓ http://www.psmsc.com/ Practical Software and Systems Measurement.Sponsored by the U.S. Department of Defense and the U.S.