International Journal of Research
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 07 Issue 12
December 2020

The Role And Importance Of Software Metrics In Quality And Reliability Of A Product

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 and reliability of a product has been analysed and discussed

What is Quality?

Quality is defined as fitness for purpose, or grade. It can also be defined as degree of preference. The manufactured products are as per the customer's specification.

What is Quality Control?

Quality Control (QC) can be defined as the measurements of components, at various manufacturing stages in order to control quality of the products being manufactured. QC will ensure that the manufactured components are within the acceptable tolerances.

Statistical Quality Control (SQC):

A statistical method for determining the extent to which the quality goals are being met without checking produced every item. It is also used for indicating whether or not the variations which occur are exceeding normal expectations

General activities of Statistical Quality Control:

1. Systematic collection & graphic recording of Data
2. Analyzing the data
3. Practical Engineering

Types of causes:

There are two types of causes

I) Chance causes:

The causes in which the variations are beyond the control of human hand & cannot be prevented are called as chance causes.

II) Assignable Causes:

The causes in which the variations are preventable are called as Assignable causes.

Factors deciding the Quality of a product

- Quality of materials
- Quality of manpower
- Quality of Machines
- Quality of Management

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.

Aesthetic Software quality: The Software quality with elegant and easy to use commands and interfaces attractive screens and well formatted outputs.

Software Quality Control: The function of software quality control is to measure /check the standards, processes, and procedures of the manufactured product with the aid of software in order to ensure that the project produces the required internal and external (deliverable) products.
Software Quality Assurance: Software Quality Assurance is assuring & correct implementation of the appropriate standards, processes and procedures for the product.

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

International Journal of Research
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 07 Issue 12
December 2020

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.

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

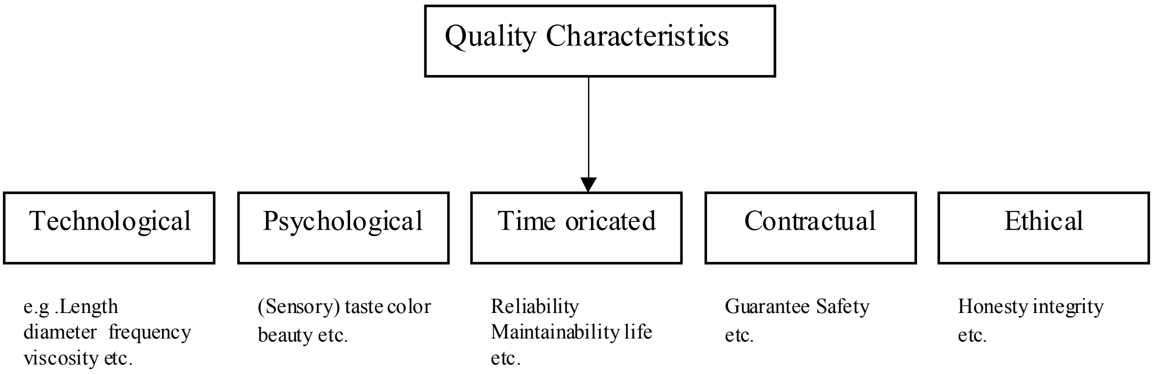
COMPARISON OF SOFTWARE METRICS STRENGTHS AND WEAKNESSES

A. *Source Code Metrics*
B. Function Point Metrics
C. Object-Oriented Metrics

Quality Characteristics

A physical or chemical property, a dimension, a temperature, pressure, taste, small or any other requirements used to define the nature of the product or service (which contributes to fitness for use) is a quality characteristic. Thus, a metal cylinder may be defined by stating the quality characteristics such as, the type of metal, the length, the diameter etc. the quality characteristic contributes to fitness for use for the product.

Quality characteristics can be classified as:



Statistical Quality Control(S.Q.C)

A quality control system performs inspection, testing and analysis to ensure that the quality of the products produced is as per the laid down quality standard. It is called " Statistical Quality Control" when statistical techniques are employed to control, improve and maintain quality or to solve quality problems. Statistic is the collection, organization, analysis, interpretation and presentation of the data. It is based on law of large numbers and mathematical theory of probability. It is just one of the many tools necessary to solve quality problems it takes into account the existence of variation. It is in this sense that the adjective statistical is used in the expression statistical quality control. Building an information system to satisfy the concept of prevention and control and improving upon product quality, requires statistical thinking.

Statistical quality control is systematic as compared to guess-work of haphazard process inspection, and the mathematical statistical approach neutralizes personal bias and uncovers poor judgement. Statistical quality control consists of there general activities:

1. Systematic collection and graphic recording of accurate data.
2. Analyzing data.
3. Practical Engineering or management action, if the information obtained indicates significant deviations from the specified limits.

Modern techniques of statistical quality control and acceptance sampling have an important part to play in the improvement of quality, enhancement of productivity, creation of consumer confidence and development of industrial economy of the country.

The following are some of the statistical tools useful for Quality Control.

1.*Frequency Distribution:* Frequency distribution is a tabulation or tally of the number of times a given quality characteristic occurs within the samples.

Graphic representation of frequency distribution will show:

(a) Average quality.
(b) Spread of quality
(c) Comparison with specific requirements
(d) Process capability

2. *Control chart:* Control chart is graphical representation of quality characteristics, which indicates whether the process is under control or not.

3. *Acceptance sampling:* Acceptance sampling is the process of evaluating a portion of the product/material in a lot for the purpose of accepting or rejecting the lot on the basis of conforming or not conforming to a quality specification. It reduces the time and cost of inspection and exerts more effective pressure on quality improvement than it is possible by 100 percent inspection. It is used when assurance is desired for the quality of materials/products either produced or received.

4. *Analysis of the data:* It includes special methods, which include such techniques as the analysis of tolerances, correlation, analysis of variance, analysis for engineering design, problem solving technique to eliminate cause of troubles.

Statistical methods can be used in arriving at proper specification limits of product, in designing the product, in the purchase of raw material, semifinised and finished products, manufacturing process, inspection, packaging, sales and also after sales service.

Industrial Quality Assurance

The main objects in any production process is to control and maintain the quality of the manufactured product so that it conforms to specified quality standards. In other words, one wants to ensure that the proportion of defective items in the manufactured product is not too large. This is called 'Process control' and is achieved through the techniques of control charts pioneered by W.A. Shewhart.

On the other hand, by product control one can mean controlling the quality of the product by critical examination at strategic points and it is achieved through 'Sampling Inspection Planes' pioneered by Dodge and Roming product control aims at guaranteeing a certain quality level to the producer. In other words, it attempts to ensure that the product marketed by sale department does not contain a large number of defective (unsatisfactory items).

So far the quality assurance concepts are mainly concerned with manufactured product. But for the past one decade IT sector plays vital role in the Indian export scenario.

Uses of Information Technology (hereafter called as IT) is like talking about the uses of electricity in modern life. IT as such is useful to human race in all into form.

First and the foremost use of IT and its development is employment. IT field need a lot of sharp and working brains. So the grave problem of unemployment is solved by IT to a great extent. Its said that about75% of the employee of Microsoft company are Indians. And even people like sabir Bhatia the inventor of Hotmail earn a lot of foreign currency, in India.

SOFTWARE QUALITY ASSURANCE

The nature of standards and the way standards are being developed is changing. In the past, standard development was an evolutionary process governed by the market dominance. Well established methods and practices adopted by some market-dominant companies were recognized as industry standards. These industry standards in turn became national standards, which in turn, resulted in international standards.

Although market dominance will continue to play an important role in future, international software standardization has become both technology driven and business oriented. Technology issues concerning engineers and designers and business issues concerning executives.

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.

Another example is IEEE standard 1061 for a Software Quality Metrics Methodology which deals with quality metrics.

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).

ISO-9000 series for Quality Management and Quality Assurance represent a concise and generic description of essential elements of management system for assuring quality in development, production and qualification. However, specific methods are left to the user.

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.

In software development, quality of design encompasses requirements, specifications, and the design of the system. Quality of conformance is an issue focused primarily on implementation. If the implementation follows the design and the resulting system meets its requirements and performance quality is high.

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.

Quality control activities may be fully automated, entirely manual, or a combination of automated tools and human interaction. A key concept of quality control is that all work products have defined, measurable specifications to which one may compare the output of each process. The feedback loop is essential to minimize the defects produced.

Quality Assurance

Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance 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. Of course, if the data provided through quality assurance identify problems, it is management`s responsibility to address the problems and apply the necessary resources to resolve quality issues.

Cost of quality

The cost of quality includes all costs incurred in the pursuit of quality or in performing quality-related activities. Cost of quality studies are conducted to provide a base line for the current cost of quality, identify opportunities for reducing the cost of quality, and provide a normalized basis of comparison. Furthermore, one can evaluate the effect of changes in dollar-based terms. Quality costs may be divided in to costs associated with prevention, appraisal, and failure prevention costs include.

- Quality planning
- Formal technical reviews
- Test equipment
- Training

Appraisal costs include activities to gain insight into product condition the " first time through" each process. Examples of appraisal costs include

- In-process and interprocess inspection
- Equipment calibration and maintenance
- Testing

SOFTWARE RELIABILITY

Software reliability, unlike many other quality factors, can be measured directed and estimated using historical and development data. Software reliability is defined in statistical terms as " the probability of failure-free operation of a computer program in a specified environment for a specified time" [MUS87]. To illustrates, program X is estimated to have a reliability of 0.96 over eight elapsed processing hours. In other words, if program X were to be executed 100 times and require eight hours of elapsed processing time (execution time), it is likely to operate correctly (without failure) 96 times out of 100.

The MTBF reliability measure is equality sensitive to MTTF and MTTR. The availability measure is somewhat more sensitive to MTTR, an indirect measure of the maintainability of software.

Software Safety

Software Safety is a software quality assurance activity that focuses on the identification and assessment of potential hazards that may affect software negatively and cause an entire system to fail. If hazards can be identified early in the software engineering process, software design features can be specified that will either eliminate or control potential hazards.

A modeling and analysis process is conducted as part of software safety. Initially, hazards are identified and categorized by critically and risk. For example, some of the hazards associated with a computer-based control for an automobile might be

- ✯ Cause uncontrolled acceleration that cannot be stopped.
- ✯ does not respond to depression of brake pedal (by turning off)
- ✯ does not engage when switch is activated
- ✯ slowly loses or gains speed

Measurement Theory

Managing any development of a project requires adequate measurements to be taken and used as feedback to construed better strategies and techniques for future projects,

As a consequence of source of these concerns the last few years have seen a rapid growth with set that any metric should be developed and tested in the context of measurement theory and that application of such theory will assist in clarifying, whether any specific measure is appropriate in such individual situation.

In addition to the focus on measurement theory, Zuse also introduces the view of a view point. A view point is simply a way to look at something that is an empirical relation he feels that measures should be based on specific view points.

1. A measure must be part of overall strategy for software development process improvement.
2. A measure must be relevant to the grabs of organization.
3. A measure must be accurate, that is; they must possess some form of internal validity.
4. A measure must address both structure of process and data.
5. A useful measure must be consistent, that is reliable,

Validity

It can be classified in three categorize. 1.face, 2.internal, 3.external,

- ✯ All face validity implies that a measured seems to be valid " on the face of it" This is mainly useful in practical perspective.

International Journal of Research
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 07 Issue 12
December 2020

- ✯ Internal validity deals with how measure captures real differences in the values of an attribute. For example using the definition of SLOC, in comparing in the C programming based on length/size, if the number of semicolon is greater in one programme then one programme is larger than other.
- ✯ External validity deals with complexity of the programme corrections can be used to find complexity.

It is observed that many metrics are validated against only one data set. The first method used to validated metrics is the multiple regression techniques. Second validation method is correlation a good influential statistical approach demands a priori statement of null hypotheses, then it becomes clear what entities should be measure for example the hypothesis may be differ for the same system design implemented in two different programming languages. So that the validation process can be summarized as follows.

1. The collection large amount of data.
2. A correlation each measure with every other measure.

Identification of good correlation.

Reliability

This deals with stability, with the idea but the measure should produce same results given the same entity in the same environments depending upon the type of the measure being tested.

Robustness

Robustness is the ability to recover from failure and to be able to handle incorrect input.

SCALE TYPES

A Nominal Scale

A nominal scale is a simple labeling

- A number desired from product process, or resources (e. g, Loc, Effort, personal experience).
- A scale of measurement software failure:- Yes or No.
- An identifiable attribute (e.g. “probability” or “coupling”).

The only statistics possible are non parametric ones. In other words we can use only models, frequencies medians and test such as spearman and Kendall’s we can not use means, standard deviations or the common pearson product moment correlation.

Ordinal scales are stronger than nominal scales that they permit ordering. An example of an ordinal scale is a typical Likert scale such as "Rank order statistics are possible in addition to useful for nominal data.

Internal scales are the next in order of "Strength". They are measurable quantities like Lines Of Code(LOC).

Ratio scales permit further calculation based on rations and percentages. Examples probability, index number and temperature in degrees Kelvin. These are all scales for which a zero value has a meaning.

Measurement theory is being increasingly recognized as having a major contribution to make to the maturity of software engineering metrics.

SOFTWARE QUALITY METRICS

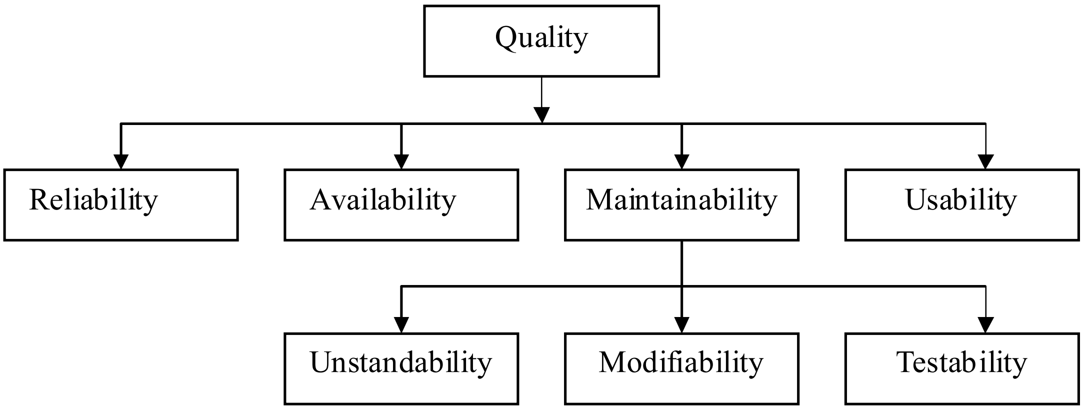
Internal Characteristics

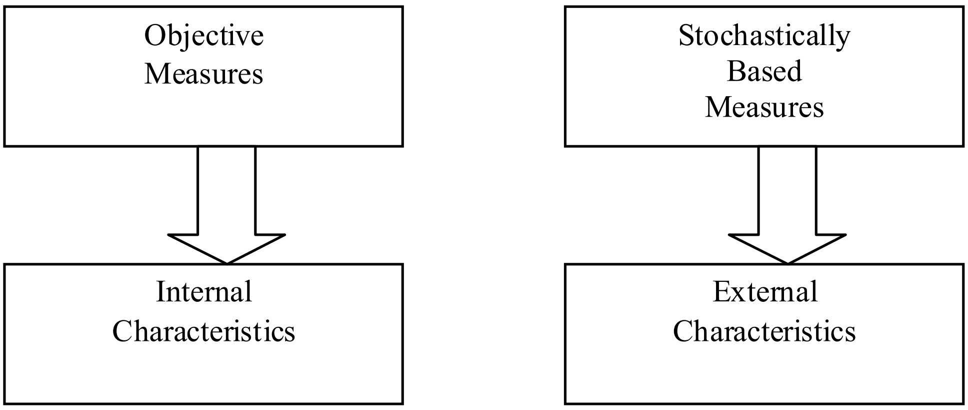
It describe structural complexity common measures of structural complexity are metrics for size, data structure, control flow complexity and inter module coupling. It is also influenced by the psychological characteristics of the analysis /designer/ programmer, the complexity of the problem, and complexities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

International Journal of Research
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 07 Issue 12
December 2020

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.





Typical Characteristics
* Size
* Modifiability
* Control flow complexity
* Modular Cohesion

Typical Characteristics
* Complexity
* Understand ability
* Testability
* Maintainability
* Quality

Desirable properties of software metrics

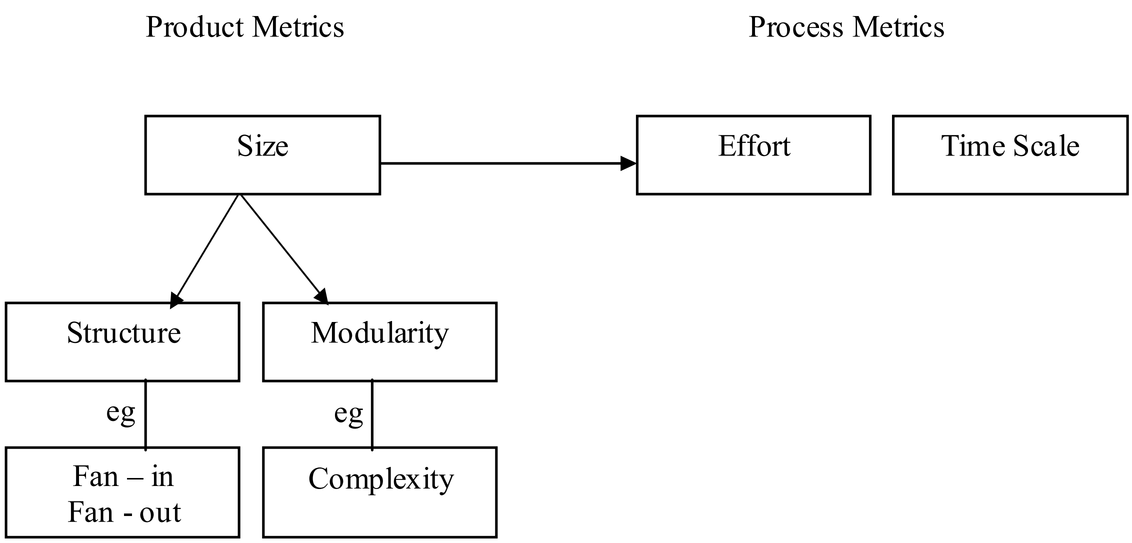
The value must be computed in a resize manner

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.

Typically as with Quality Assurance (QA) in manufacturing. The instantaneous inspection can give only a limited in sight not until process based management techniques, such as total quality management (TQM), replace QA can a system be said to be fully controlled. Manufacturing insights have some applicability to software development, although the major difference between a highly repeatable and repeated most manufacturing industry churning and identical over many years and a software industry developing that is never identical, even though there may be strong similarities between each accounting system, leads some with to question the extent to which manufacturing statistical control (which underlines TQM, for instance) can be of use to software development.

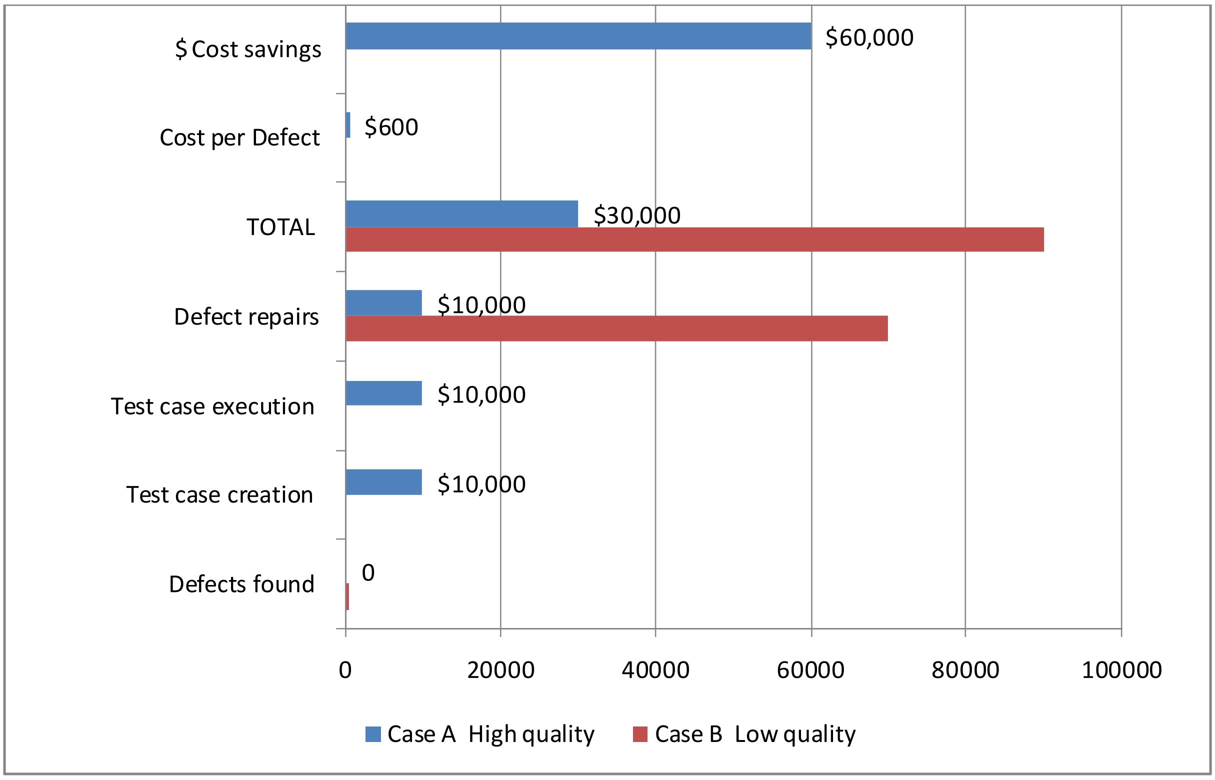


The software project manager has a variety of uses for software metrics both for product and process, Responsibility of a project Manager range from applying (largely product metrics) to monitoring, controlling innovating, planning, and predicating. An in depth study on software metrics is done

Table .COMPARISON OF COST PER DEFECT

(IT AFFECTS QUALITY)

Metrics	**Case A**	**Case B**
Software	**High quality**	**Low quality**
Defects found	50,	500
Test case creation	$10,000	,$10,000
Test case execution	$10,000	,$10,000
Defect repairs	$10,000	$70,000
TOTAL	$30,000	$90,000
Cost per Defect	$600	$180
$ Cost savings	$60,000	$0.00



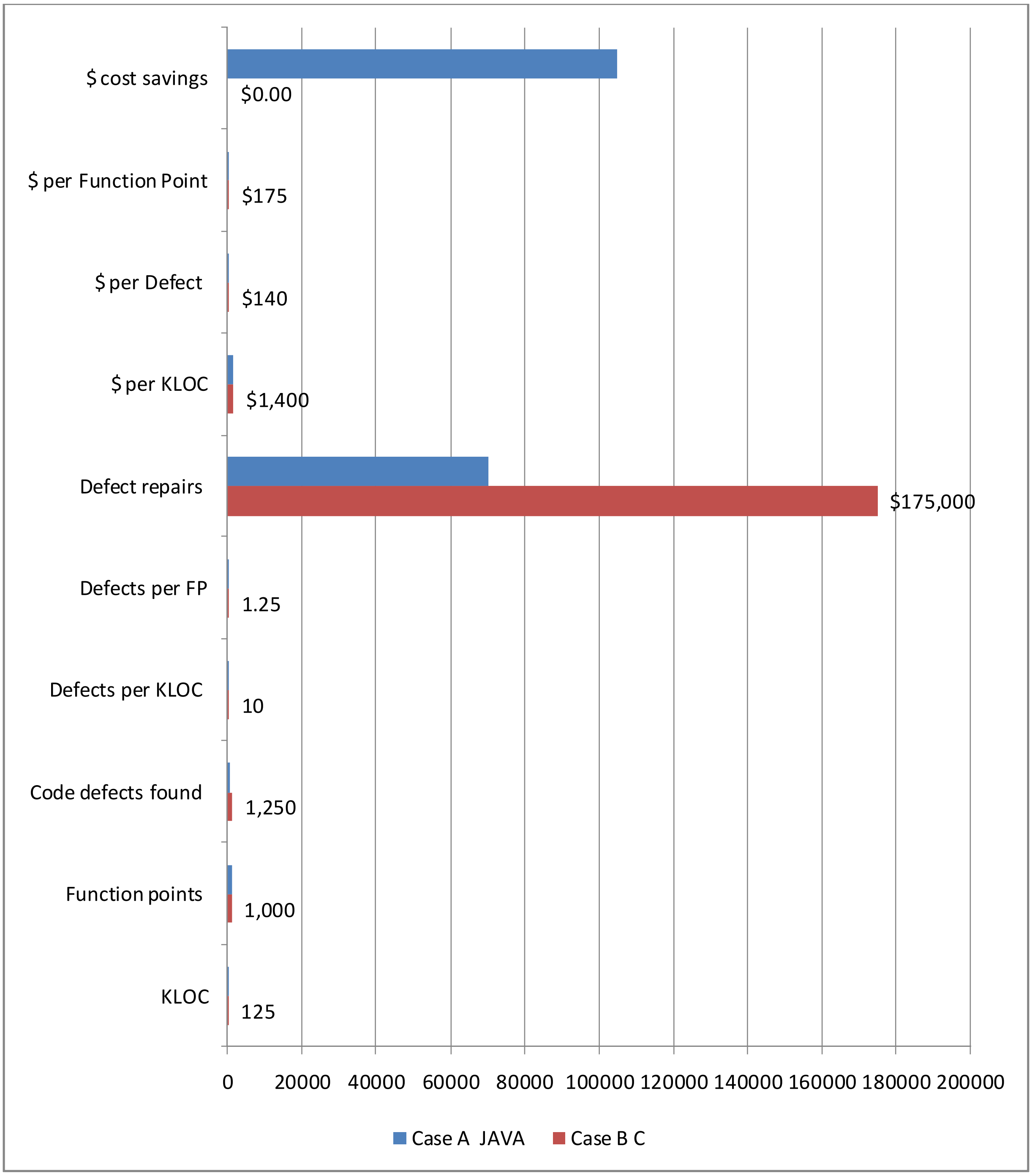
Graph 1. .COMPARISON OF COST PER DEFECT
(IT AFFECTS QUALITY)

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

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

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



Graph 2. LINES OF CODE HARM HIGH-LEVEL LANGUGES

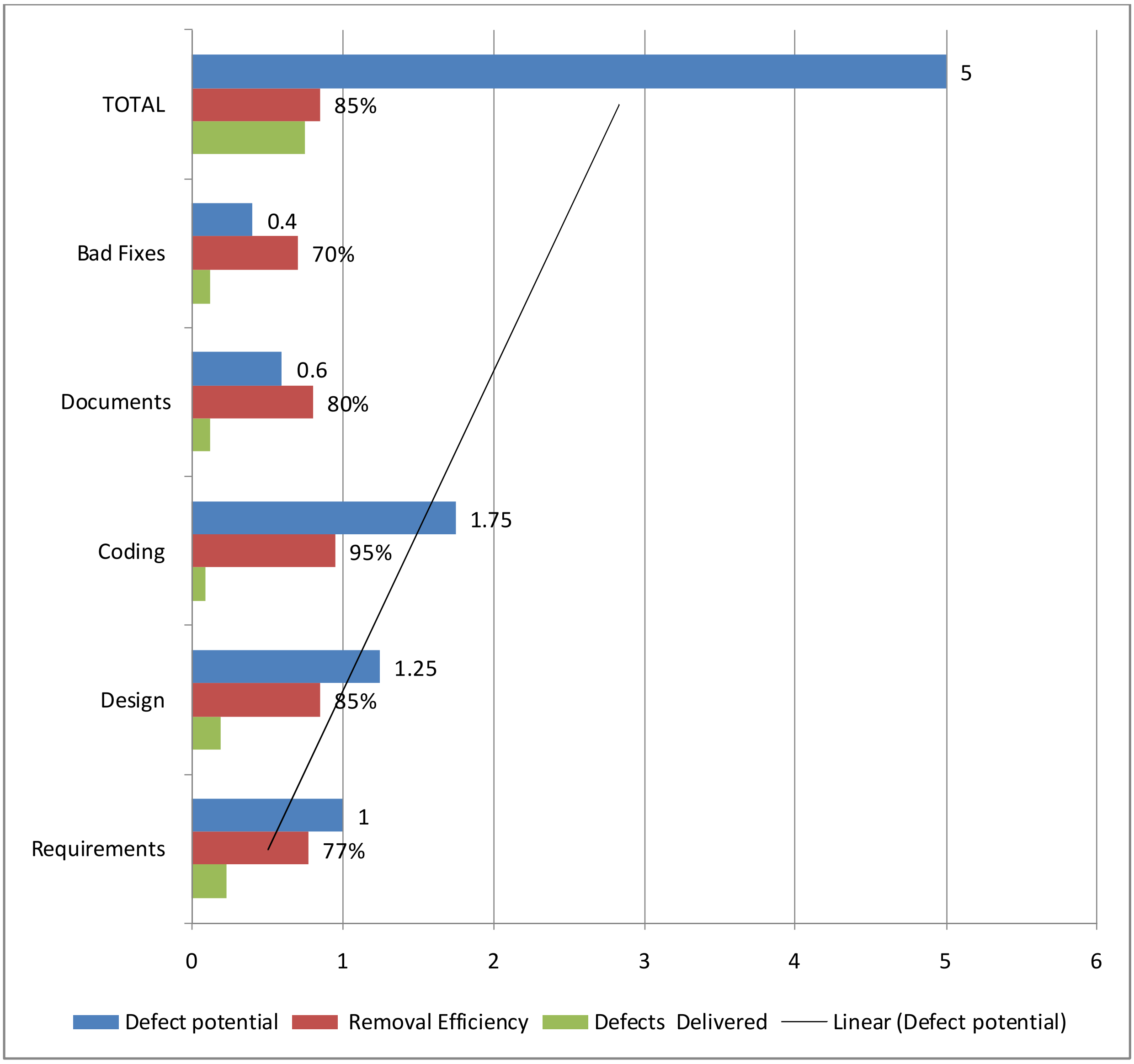
Table U.S. AVERAGES FOR SOFTWARE QUALITY

(Data expressed in terms of defects per function point)

Factor	Defect potential	Removal Efficiency	Defects Delivered
Requirements	1.00	77%	0.23
Design	1.25	85%	0.19
Coding	1.75	95%	0.09
Documents	0.60	80%	0.12
Bad Fixes	0.40	70%	0.12
TOTAL	5.00	85%	0.75

(Function points show all defect sources - not just coding defects)

(Code defects = 35% of total defects)



Graph 3.U.S. AVERAGES FOR SOFTWARE QUALITY

(Data expressed in terms of defects per function point)

International Journal of Research

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

Table. BEST IN CLASS SOFTWARE QUALITY

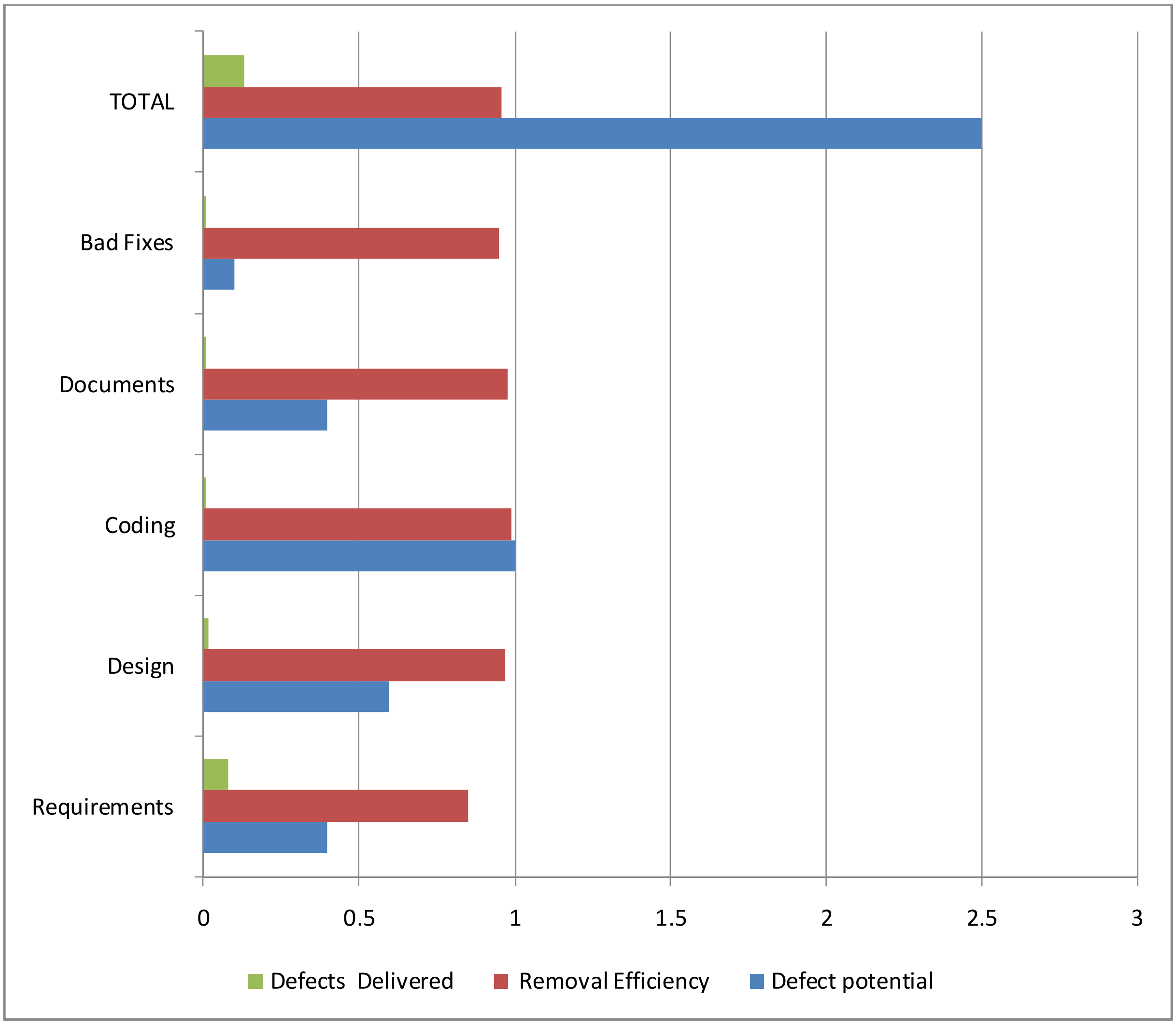
Factor	Defect potential	Removal Efficiency	Defects Delivered
Requirements	0.40	85%	0.08
Design	0.60	97%	0.02
Coding	1.00	99%	0.01
Documents	0.40	98%	0.01
Bad Fixes	0.10	95%	0.01
TOTAL	2.50	96%	0.13

(Data expressed in terms of defects per function point)

OBSERVATIONS

(Most often found in systems software > SEI CMM Level 3 or in TSP projects)

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



Graph . BEST IN CLASS SOFTWARE QUALITY

Table 25. POOR SOFTWARE QUALITY - MALPRACTICE

Defect Removal Delivered

	Defect Origins	Potential	Efficiency
Requirements	1.50	50%	0.75
Design	2.20	50%	1.10
Coding	2.50	80%	0.50
Documents	1.00	70%	0.30
Bad Fixes	0.80	50%	0.40
TOTAL	8.00	62%	3.05

(Data expressed in terms of defects per function point)

OBSERVATIONS

(Most often found in large water fall projects > 10,000 Function Points).

GOOD QUALITY RESULTS > 90% SUCCESS RATE

- Formal Inspections (Requirements, Design, and Code)
- Text static analysis
- Code static analysis (for about 25 languages out of 2,500 in all)
- Joint Application Design (JAD)
- Requirements modeling
- Functional quality metrics using function points
- Structural quality metrics such as cyclomatic complexity
- Defect Detection Efficiency (DDE) measurements
- Defect Removal Efficiency (DRE) measurements
- Automated defect tracking tools
- Active quality Assurance (> 3% SQA staff)
- Mathematical test case design based on design of experiments
- Quality estimation tools
- Testing specialists (certified)
- Root-Cause Analysis

MIXED QUALITY RESULTS: < 50% SUCCESS RATE

- CMMI level 3 or higher (some overlap among CMMI levels: Best CMMI 1 groups better than worst CMMI 3 groups)
- ISO and IEEE quality standards (Prevent low quality; Little benefit for high-quality teams)
- Six-Sigma methods (unless tailored for software projects)
- Quality function deployment (QFD)
- Independent Verification & Validation (IV & V)
- Quality circles in the United States (more success in Japan)
- Clean-room methods for rapidly changing requirements

- Kaizan (moving from Japan to U.S. and elsewhere)
- Cost of quality without software modifications
- Pair programming

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
- Passive Quality Assurance (< 3% QA staff)
- Token Quality Assurance (< 1% QA staff)
- LOC Metrics for quality (omits non-code defects)
- Cost per defect metric (penalizes quality)
- Failure to estimate quality or risks early
- Quality measurement “leakage” such as unit test bugs

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%

• Static analysis –often more than 65% efficient; has topped 95%

• Inspections, static analysis, and testing combined lower costs and schedules by > 20% ;

lower total cost of ownership (TCO) by > 45%. Quality Measurements Have Found:

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
2. Defect removal technologies
   - ✓ Design inspections
   - ✓ Code inspections, static analysis
   - ✓ Tabulation &Testing with Chi square Test

The Ranges of Defect Removal efficiency for different factors and Defects and software methodologies

- 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,

- **REFERENCES**

Rohatgi, V.K., Statistical Inference, New York: John Wiley & Sons, 1984.

Rudisill, d.,"QCASE'': A New Paradigm for Computer-Aided Software Engineering," International Software Quality Exchange 92 Conference Proceedings, San Francico, Calif., March 10-11. 1992. Sponsored by the Juran Institute, Wilton, Conn., pp.4A-19-4A-3A.

Snedecor, G. W., and W.G. Cochran, Statistical Methods, 7th ed., Ames, lowa: The lowa State University Press.

Snedecor,G.W., and W.G. Cochran, Statistical Methods, 7th ed., Ames, Jowa: The lwa State University Press, 1980.

Software Productivity Research, Quality and Productivity of the SEI CMM, Burlington,Mass,: Software Productivity Research, 1994.
Udpa S.R.: Quality Circles in India.
Xie, M., G.Y.Hong, and C.Wohlin "Software Reliability Prediction Incorporating Information from a similar project," The Journal of system and Software, Vol.49, 1999.
Xie,M., and G.Y.Hong, " Software Reliability Modeling and Analysis, "chapter in Handbook of statistics 20: Advances in Reliability N.Balakrishnan and C.R.Rao, eds. New York: Elsevier Science, 2002.