Name: A. Shiva Surya Saran
Roll No.: BE-B 20

Software Testing and Quality Assurance

Assignment 6

Unit 6: Software Quality Tools

1. Describe key elements of Total Quality Management.

Some of the important elements of total quality management are:

1. Management's commitment to quality
   If an organization is serious about implementing TQM, the lead must be taken by the top management with full commitment. It must initiate quality improvement programs. The top management should continue all the efforts and provide the resources to continue quality improvement programs. This is provided by collecting, reporting and use of quality related cost information.

2. Customer satisfaction
   TQM is designed in such a manner to meet the expectations of customers. In the present era, customer is the king. It must be recognized that customers are the most important persons for any business. The very existence of an organization depends on them. They are the life blood of a business and deserve the most courteous and affectionate treatment.

3. Preventing rather than detecting defects
   TQM checks the poor-quality products or services rather than simply to detect and sort out defects. "Prevention rather than detection" is the main characteristic of TQM. Some of the important techniques of TQM which aim at the prevention of defects rather than the detection of the defects are statistical process control, continuous process improvement and problem solving and system failure analysis etc.

4. Measurement of Quality
   Quality is a measurable entity, and we must know what current quality levels are i.e. Where we are or where we stand in respect of the quality and what quality levels we are aspiring for or where we are going.

5. Continuous improvement
   TQM comprises of a continuous process of improvement covering people, equipment, suppliers, materials, and procedures. It includes every aspect of an operation in an organization. In Japan, the word "Kaizen" is used to describe the continuous process of improvement.

6. Corrective action for root cause
   TQM aims at preventing repetition of problems by identifying the root causes for their

occurrence and developing means and corrective actions to solve the problems of the root level. Failure analysis and problem-solving skills are very useful techniques in this regard.

7. Training
   Proper training programmed must be undertaken to train the employees for the use of TQM concepts and techniques. Employees must be provided regular training for continuous improvement.

8. Recognition of high quality
   TQM aims at developing long term relationships with a few high-quality suppliers rather than those suppliers who supply the inferior goods at the low cost.

9. Involvement of Employees
   Involvement of employees means that every employee is completely involved at every step of production process which plays an active role in helping the organization to meet its targets. Employee involvement and empowerment can be assured by enlarging the employee's job so that responsibility and authority is moved to the lowest level possible in the organization.

10. Benchmarking
    Benchmarking is a systematic method by which organizations can measure themselves against the best industry practices. Benchmarking aims at developing best practices that will lead to better performance. It helps a company to learn and incorporate the best practices into its own operations. Benchmarking is a technique of distinguishing an organization's efforts with the best performance in the field and to suggest how the gap between the two performances can be removed. Thus, benchmarking is a technique of continuous improvement.

2. Explain Product Quality Metric and Process Quality Metrics.

Product Quality Metrics

Describes the characteristics of the product such as size, complexity, design features, performance, and quality level.

This metrics include the following −

- Mean Time to Failure
- Defect Density
- Customer Problems
- Customer Satisfaction

Mean Time to Failure

It is the time between failures. This metric is mostly used with safety critical systems such as the airline traffic control systems, avionics, and weapons.

Defect Density
It measures the defects relative to the software size expressed as lines of code or function point, etc. i.e., it measures code quality per unit. This metric is used in many commercial software systems.

Customer Problems
It measures the problems that customers encounter when using the product. It contains the customer's perspective towards the problem space of the software, which includes the non-defect-oriented problems together with the defect problems. The problems metric is usually expressed in terms of Problems per User-Month (PUM).

Customer Satisfaction
Customer satisfaction is often measured by customer survey data through the five-point scale −

- Very satisfied
- Satisfied
- Neutral
- Dissatisfied
- Very dissatisfied

Satisfaction with the overall quality of the product and its specific dimensions is usually obtained through various methods of customer surveys. Based on the five-point-scale data, several metrics with slight variations can be constructed and used.

Process Quality Metrics

Process quality metrics deals with the tracking of defect arrival during formal machine testing for some organizations. This metric includes −

- Defect density during machine testing
- Defect arrival pattern during machine testing
- Phase-based defect removal pattern
- Defect removal effectiveness

Defect rate during formal machine testing
It (testing after code is integrated into the system library) is correlated with the defect rate in the field. Higher defect rates found during testing is an indicator that the software has experienced higher error injection during its development process, unless the higher testing defect rate is due to an extraordinary testing effort.

Defect arrival pattern during machine testing
The overall defect density during testing will provide only the summary of the defects. The pattern of defect arrivals gives more information about different quality levels in the field.

Phase-based defect removal pattern
This is an extension of the defect density metric during testing. In addition to testing, it tracks the defects at all phases of the development cycle, including the design reviews, code inspections, and formal verifications before testing.

Defect removal effectiveness
It can be defined as follows −

$$DRE = \frac{Defect\ removed\ during\ a\ development\ phase}{Defects\ latent\ in\ the\ product} \times 100\%$$

This metric can be calculated for the entire development process, for the front-end before code integration and for each phase. It is called early defect removal when used for the front-end and phase effectiveness for specific phases.

3. Explain need of Software maintenance

Software Maintenance is the process of modifying a software product after it has been delivered to the customer. The main purpose of software maintenance is to modify and update software application after delivery to correct faults and to improve performance.

Need for Maintenance –
Software Maintenance must be performed to:

- Correct faults.
- Improve the design.
- Implement enhancements.
- Interface with other systems.
- Accommodate programs so that different hardware, software, system features, and telecommunications facilities can be used.
- Migrate legacy software.
- Retire software.

4. List ishikawa's 7 basic tools.

Ishikawa's 7 Basic Tools of Quality is a designation given to a fixed set of graphical techniques identified as being most helpful troubleshooting issues related to quality. They are called basic because they are used easily by people with little formal training in statistics.

The seven tools are:

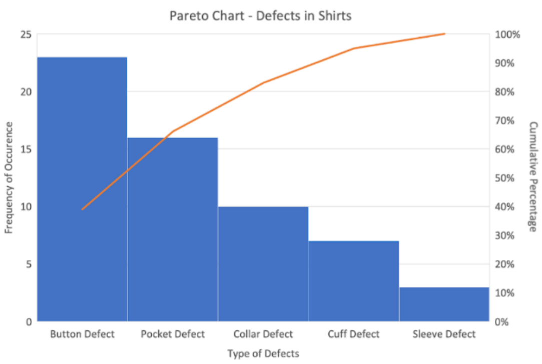
1. Check sheet
2. Histogram
3. Pareto chart
4. Control chart
5. Scatter diagram
6. Stratification (alternately, flow chart or run chart)
7. Cause-and-effect diagram (also known as the "fishbone" or Ishikawa diagram)

5. Explain following terms (any two) Pareto Chart, Scatter Diagrams, Cause and effect diagrams.

Pareto Chat

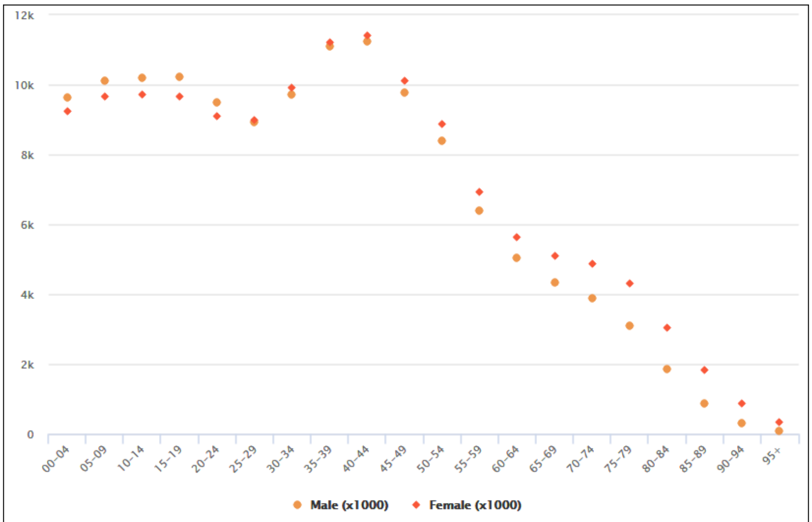
A Pareto Chart is a graph that indicates the frequency of defects, as well as their cumulative impact. Pareto Charts are useful to find the defects to prioritize in order to observe the greatest overall improvement.

- A Pareto Chart is a combination of a bar graph and a line graph.
- Each bar usually represents a type of defect or problem. The height of the bar represents any important unit of measure — often the frequency of occurrence or cost.
- The bars are presented in descending order (from tallest to shortest). Therefore, you can see which defects are more frequent at a glance.
- The line represents the cumulative percentage of defects.
- A Pareto Chart is a quality tool: it helps analyze and prioritize issue resolution.
- Pareto Charts can be analyzed with the Pareto Principle, also known as the 80/20 rule.



Scatter Diagram

A scatter diagram (Also known as scatter plot, scatter graph, and correlation chart) is a tool for analysing relationships between two variables for determining how closely the two variables are related. One variable is plotted on the horizontal axis and the other is plotted on the vertical axis. The pattern of their intersecting points can graphically show relationship patterns.



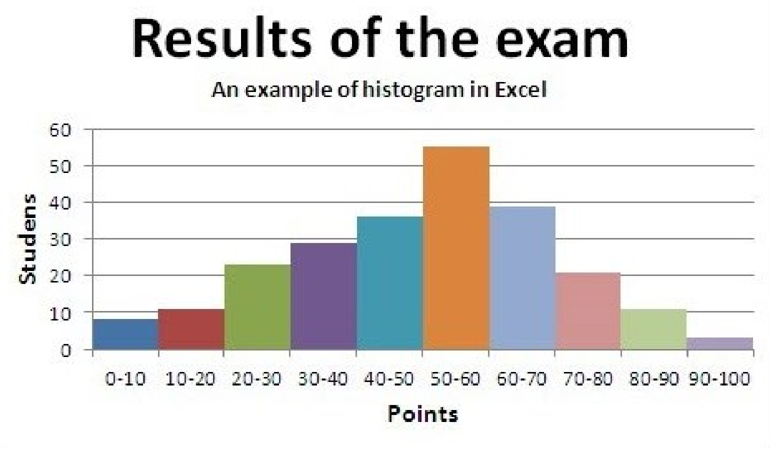
United States Population of Year 2000

Most often a scatter diagram is used to prove or disprove cause-and-effect relationships. While the diagram shows relationships, it does not by itself prove that one variable causes the other. Thus, we can use a scatter diagram to examine theories about cause-and-effect relationships and to search for root causes of an identified problem.

For example, we can analyse the pattern of motorcycle accidents on a highway. You select the two variables: motorcycle speed and number of accidents and draw the diagram. Once the diagram is completed, you notice that as the speed of vehicle increases, the number of accidents also goes up. This shows that there is a relationship between the speed of vehicles and accidents happening on the highway.

6. Use of histogram in test quality of product or project.

A histogram (or frequency distribution chart) is a bar graph that groups data by predetermined intervals to show the frequency of the data set. It provides a way to measure and analyse data collected about a process or problem and may provide a basis for what to work on first.



Histograms are also useful for displaying information such as defects by type or source, delivery rates or times, experience or skill levels, cycle times, or end user survey responses. When sufficient process data is available, a histogram displays the central point (average) of the process, variation (standard deviation and range), and shape of distribution.

Histograms can explain graphically whether a process is in control or out of control, but they are not a substitute for statistical process control charts. They can also provide insight on the process capability to meet user specifications.

7. Describe in detail Defect Removal Effectiveness.

Defect Removal Effectiveness (or efficiency as used by some writers) is calculated:

$$\text{DRE} = \frac{\text{Defects removed during a development phase}}{\text{Defects latent in the product at that phase}} \times 100\%$$

Since the latent defects in a software product is unknown at any point in time, it is approximated by adding the number of defects removed during the phase to the number of defects found later (but that existed during that phase).

For example, assume that the following table reflects the defects detected during the specified phases and the phase where those defects were introduced.

	Phase Introduced		
Phase Detected	Requirements	Design	Coding/Unit Test
Requirements	10	---	---
Design	3	18	---
Coding	0	4	26
Test	2	5	8
Field	1	2	7

The Defect Removal Effectiveness for each of the phases would be as follows:

Requirements DRE = 10 / (10+3+0+2+1) x 100% = 63%

Design DRE = (3+18) / (3+0+2+1+18+4+5+2) x 100% = 60%

Coding DRE = (0+4+26) / (0+2+1+4+5+2+26+8+7) x 100% = 55%

Testing DRE = (2+5+8) / (2+1+5+2+8+7) x 100% = 60%

Defect Removal Effectiveness can also be calculated for the entire development cycle to examine defect detection efforts before the product is released to the field. According to Capers Jones, world class organizations have Development DRE greater than 95%.[1]

Development DRE = (Pre-release Defect) / (Total Defects) x 100% =

(10+3+2+18+4+5+26+8) / (10+3+2+1+18+4+5+2+26+8+7) x 100 = 88%

The longer a defect exists in a product before it is detected, the more expensive it is to fix. Knowing the DRE for each phase can help an organization target its process improvement efforts to improve defect detection methods where they can be most effective. Future DRE measures can then be used to monitor the impact of those improvement efforts.