STQA

Links

- https://jagdishkapadnis.files.wordpress.com/2018/08/stqa-unit-1-handwritten-notes.pdf
- https://books.google.co.in/books?id=zUm8My7SiakC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- https://www.slideshare.net/VinothkumarRJ/unit-i-software-testing-and-quality-assurance

MCQ's

- http://nktdegreecollege.org/uploads/students/SQA-MCQ.pdf

Unit 1

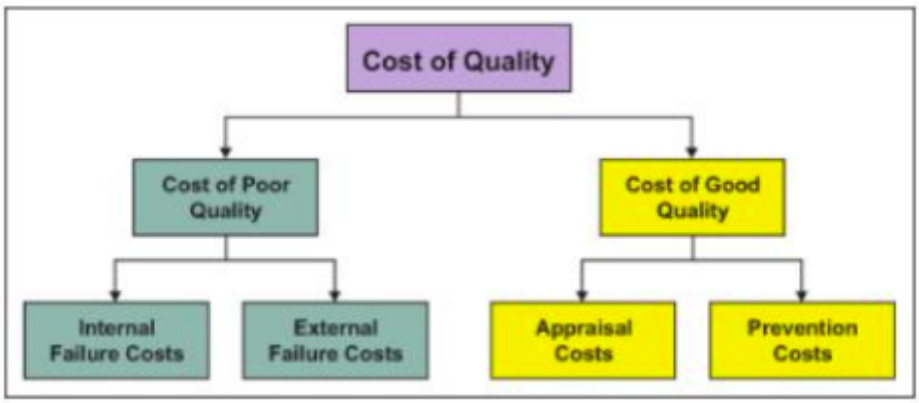
Definitions of Quality:

- customer definition
- manufacturing definition
- product definition
- value
- transcendent quality

Sr. No	Continuous Improvements	Continual Improvements
1	Continuous improvements is dynamic in nature.	Continual Improvements is dynamic & static change management.
2	The changes are done at every stage & every time.	The changes are done, absorbed baselined & sustained before taking next step of improvement.
3	Continuously striving for the excelence gives a continuously improvement.	Periodic improvements followed by stabilization process.
4	It has high dependence on people having innovative skills tending towards inventions.	Less dependence on people and more dependence on innovation processes.
5	Environment is continuously changed.	Changed in environment are followed by stabilization.

- sales price = cost of manufacturing + cost of quality + profit
- on the basis of quality focus organizations may be place in 2 categories
  - orgnizations which are less quality conscious termed as q
  - organizations which are more quality conscious termed as Q

Sr. No.	Quality Culture is 'Q'	Quality Culture is not 'q'
1	These organizations believe in listening to customers and determining their customers requirements.	These organizations assume that they know customer requirements
2	These organizations concentrate on identifying cost of quality & focusing on it reduce cost of failure which will reduce cost & price.	These organizations overtook cost of poor quality & hidden factor effect. They believe in more testing to improve product quality.
3	Doing things right for the first time & every time is the motto of success.	Doing things again & again to make them right is their way of working. Inspection, rework, scrap etc and essential.
4	These organizations believe in taking ownership of processes and defects at all levels.	These organizations try to assign responsibility to defects to someone else.
5	They demonstrate leadership & commitment to quality & customer	They believe in assigning responsibility for quality to others.



- Internal failure costs are costs associated with defects found before the customer receives the product or service. External failure costs are costs associated with defects found after the customer receives the product or service.

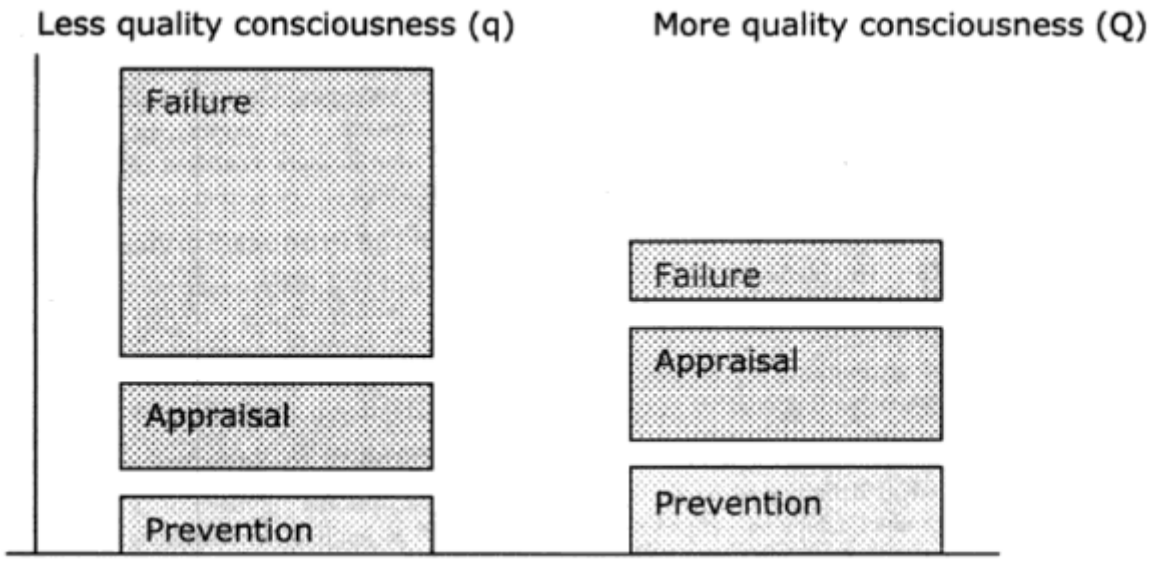


Fig. 1.2 Cost of Quality

Green Money/Cost of Prevention Green money is considered as an investment by the organisation in doing quality work. It is a cost spent in definition of processes, training people, developing foundation for quality, etc. It gives return on investment, and includes all prevention-based costs. If an organisation has defined and optimised processes, trained people, and fixed guidelines and standards for doing work which are followed, then the return on such investment can be seen in terms of less inspection and testing, and higher customer satisfaction with repeat orders. This improves profitability of an organisation.

Blue Money/Cost of Appraisal Blue money is a cost incurred by the organisation during development, in the form of first-time review/testing which gets returned in future. It does not earn profit, but it is an essential part of development process to ensure the process capability. All appraisal techniques used during SDLC such as first-time verification or validation are considered as blue money. First-time testing helps in certifying that nothing has gone wrong and work product can go to the next stage. In initial phases, the cost of appraisal increases, but as the organisational process maturity increases, this cost must go down as fewer samples are required to prove the correctness of the process of making software.

Red Money/Cost of Failure Red money is a pure loss for the organisation. It involves money lost in scrap, rework, sorting, etc. It also represents loss due to various wastes produced during development life cycle, and directly reduces the profit for the organisation and the customer may not pay for it. As the investment or green money increases, failure cost must go down. All cost incurred in reinspection, retesting, and regression testing represent cost of failure.

- **Cost of Prevention** This is a cost spent by an organisation so that defect does not occur in the first place. This includes the cost in planning and training so that a good product is made in first go. This is also termed 'green money'.
- **Cost of Appraisal** This is a cost spent by an organisation to evaluate that the product is good. Any kind of first-time review/testing indicates a cost of appraisal. This is also termed 'blue money'.
- **Cost of Failure** This is a cost spent by an organisation in defect fixes, late payments, loss of goodwill, etc. This represents a loss for an organisation. This is termed 'red money'.

Innovation vs Invention

Invention	Innovation
Inventions may be accidental in nature. They are generally unplanned.	Innovation is a planned activity leading to changes.

Breakthrough changes are possible due to inventions.	Changes in small steps are possible by innovation.
Invention may lead to major changes in technology, way of doing work, etc.	Innovation generally leads to administrative improvements, whereas technological or breakthrough improvements are not possible.
Invention may lead to scraping of old technologies, old skills and sometimes, it meets with heavy resistance.	Innovation may lead to rearrangement of things but there may not be a fundamental change. It generally works on elimination of waste.

Total Quality Management (TQM)

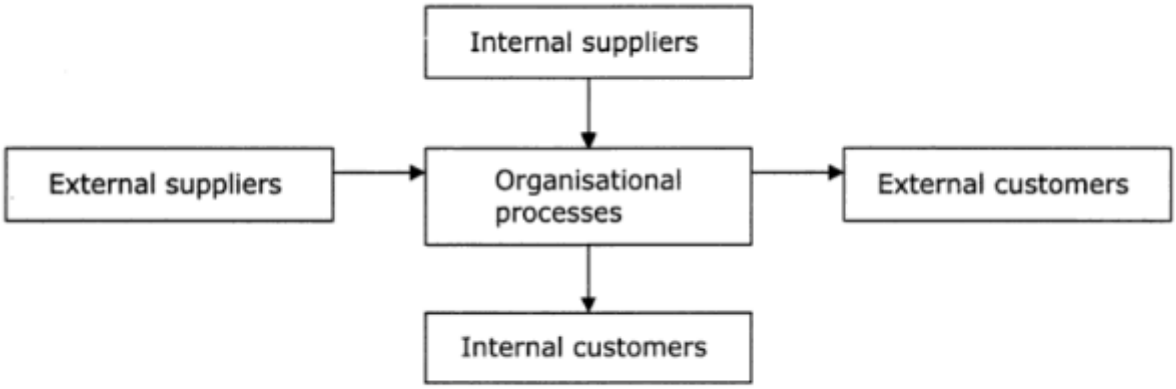


Fig. 1.3 Supply chain relationship between suppliers and customers

- Dr Edward Deming implemented quality management system drive by "TQM" and "Continual Improvemnet" in japnese environment.
- Internal customers and suppliers: guided by the principles of TQM
- Extrenal customers and suppliers: externa customer may be the final users purchasers of software

Quality Management throught statistical process control

- Dr Joseph Juran is a pioneer of statistical quality control
- Improvement cycle (DMMCI) - define, measure, monitor, control, improve

Continual Improvement Cycle

- PDCA cycle (Plan, Do, Check and Act)
- PDCA cycle is never ending cycle of improvements
- systematic sequemce of activities
- initially implmented in agriculture

Benchmarking and Metrices

- Benchmarking is the important concept used in QFD (Quality Function Deployment)
- It is the concept of creating qualitative/quantitative metrices or measure variables which can be used to asses the product quality on several scales against a benchmark.
- meric is the measurement of some parameter of a product.

Problem Solving Techniques

- Qualitative

    - It refers to understaning a problem solution using qualitative indexes suchas as high, medium, low, etc.
    - for low maturity organizations where the problems are much broader
- Quantitative
    - exact measures in numerical terms
    - for highly matured organizations

Requirements of a product

Categories of requirements:

- stated/implied: some requirements are stated in the SRS where as some are implied.
- general/specific: some requirements are genearl to the type or nature of the software whereas some are specific given by the customer.
- present/future: present requirements are used when the software is used in the present or current circumstances and the some are future requrements which are needed in the future when we may need after some time span.
- must and must not requirements or primary: Must requirements are the primary requirements for which the customer is pay for while acquiring the product. These are essential requiremets and add value to the product. Denoted by P1
- shoud be and should not be requirements or secondary: these are the requirements which may be appreciated by the customer if they are present/absent and may add some value to product. customer may pay little bit extra for these requirements. Denoted by P2
- could be and could not be or tertiary: these are the requirements which may add a competitive advantage to the product but may not add
  much valuein terms of price paid by a customer. These requirements give product identiy in the market. If two products are same the these requirements are useful for the customer to decide which one to buy.
  denoted by P3

Software Development Process

- Waterfall model

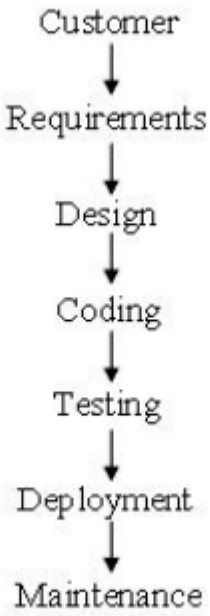


Fig. Waterfall model

- simplest software development model
- also called as classifcal view of software development
- in this model developers get the requirements from the customers in one go
- requirements are converted into low level and high level design
- waterfall model are used for fixed price projects and the price is calculated from the customer requirements and the price si revised if the requirement changes.
- disadvantages:
  - No feedback loop

- Iterative Developement Model

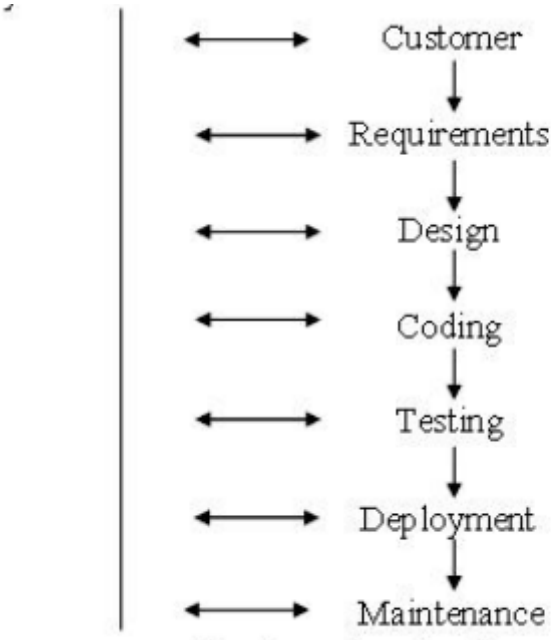


Fig. Iterative development model

  - more practical then waterfall model

    - this model does not assumes that customer gives all requirements in one go
    - consists of many cycles of waterfall model
- Incremental Model

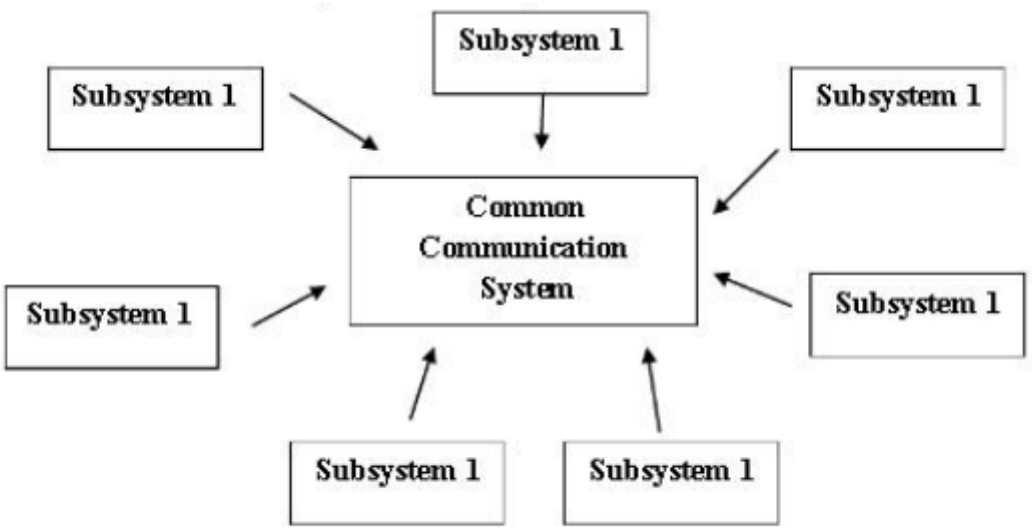


Fig. Incremental Model

    - Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle.
    - used in developing huge systems
- Spiral Development Model
    - In spiral development model there are multiple iterations of requiremnts and development
    - Initiallly customer give some basic requiremetns and the software is developed and then acoridng to the customer feedback new requiretns are given and the process goes on.
    - used for big sofwares
- Prototype development model
    - represents top to bottom reverse integration approach
    - In prototyping initially prototype of the system is created.
    - It helps the customer to understand what they can expect from the given set of requirements.
- Rapid Development Model
- Agile Model

Quality Management System Structure

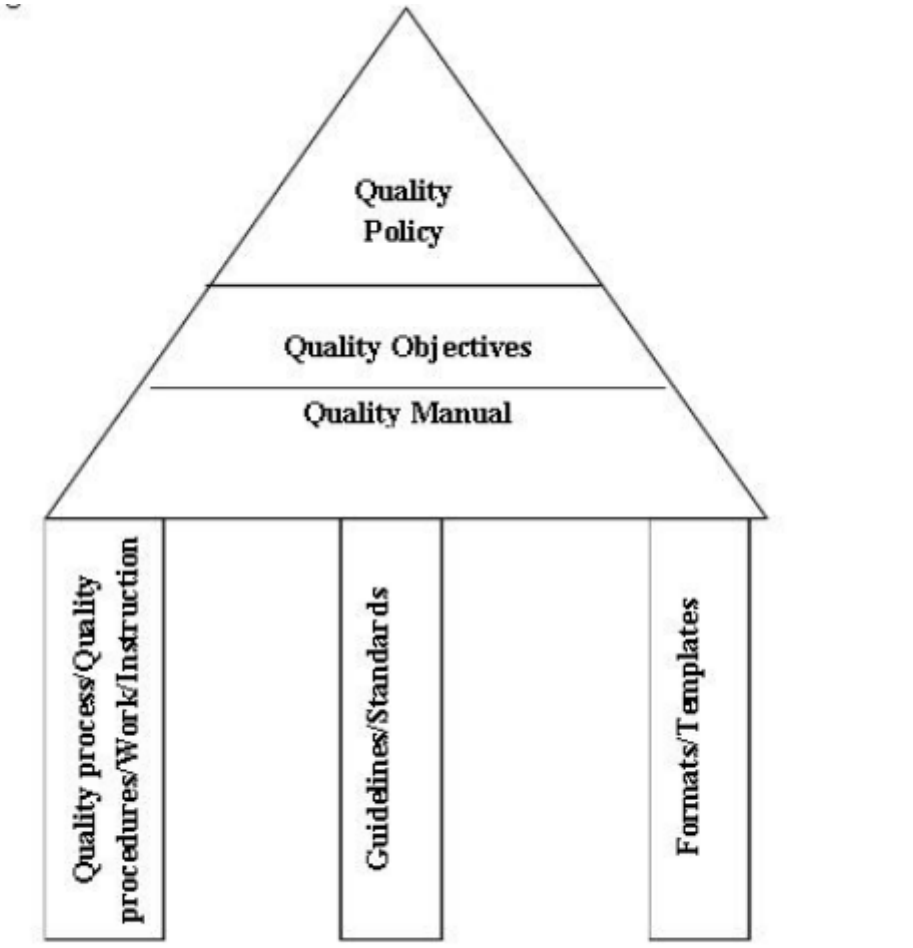


Fig. Quality Management System of a typical organization

Quality Policy: Quality policy sets the intent & direction by the management about how activities will be conducted by the organization.
Quality Objectives: Quality obectives are the measurements established by the management to define progress & achievements in a numerical way.
Quality Manual: also termed as policy manual. It sets a framework for other process definitions and is a foundation of quality planning.

Pillars of Quality Management System:

- Quality Process/Quality Procedures
- Guidelines and Standards
- Formats and templates

Unit 2

Testing process involves following activities:

- Plannign and control: involves determinign objecttives, goals of testing as well as risks, scope, etc
- analysis and design
- implmentation and execution
- evaluating exit criteria adn reporting

- test closure
- **SDLC Phases**

The entire SDLC process divided into the following stages:

Fig - Testing during development life cycle

- SRS (Software Requirement Specification) is created in feasibility study

Requirement traceability Matrix (RTM)

- RTM captures all requiremnts by the client and their traceability in a single document delivered at the conclusion of the life cycle.
- It is a document that maps and traces user requiremnts with test cases.
- main purpose of this matrix is to ensure that all the test cases are coverd so that no functionality should miss while doing software testing.
- RTM is a worksheet that contains the requirements with its all possible test cases and their current state i.e. passed or failed.
- Exhaustive testing is impossible

7 Software testing principles

- Testing shows presence of defects
- Exhaustive testing is not possible
- Early testing
- Defect clustering
    - Defect Clustering which states that a small number of modules contain most of the defects detected.
    - Pareto Principle to software testing: approximately 80% of the problems are found in 20% of the modules.
- Pesticide paradox
    - running the same set of tests over and over again, chances are no more new defects will discovered by those test cases.
- Testing is context dependent
    - testing techniques, methodologies are dependtdent on the type and nature of the application.
    - for ex medical software requires more testing then game
- Absence of errors fallacy

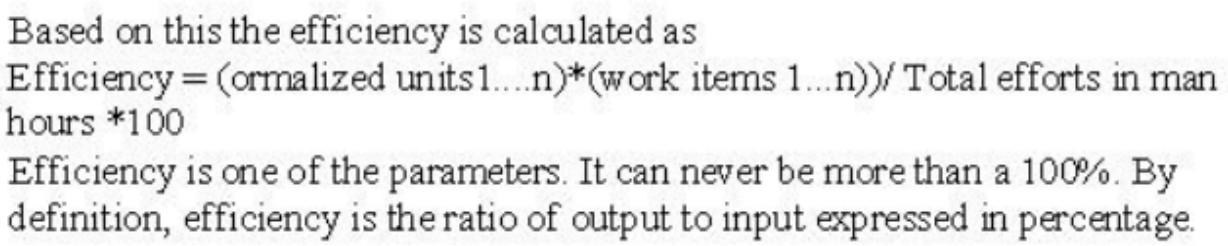
- Just because testing didn't find any defects in the software, it doesn't mean that the software is ready to be shipped.
- It is possible that software which is 99% bug-free is still unusable. This can be the case if the system is tested thoroughly for the wrong requirement.

salient and policy of software testing

- whereever organizations have ISTQB certified test managers the management of sofwtare testing can be done either at programmer level or at individual level.
- For every such level of test management there is an associated document. When structured in a top down fashion it is called the test management documentation hierarchy.

Test Plan

- A Test Plan is a detailed document that describes the test strategy, objectives, scope, schedule, estimation, deliverables, and resources required to perform testing for a software product.
- There are three types of the test plan
  - Master Test Plan: test plan that addresses multiple test levels
  - Phase Test Plan: addresses one test phase
  - Testing Type Specific Test Plans: performance and security test plan
- A test suite is a collection of test cases that are necessary to validate the system being built, againt original reuqirements.



Mutation testing

- It is a type of testing where we mutate or chnage some statements in the source code check whether the test cases can find the defect or not.
- It is a type of white box testing
- mainly used for unit testing
- the goal of mutation testing is to assess the quality of test cases.
- also called as fault based testing stretegy

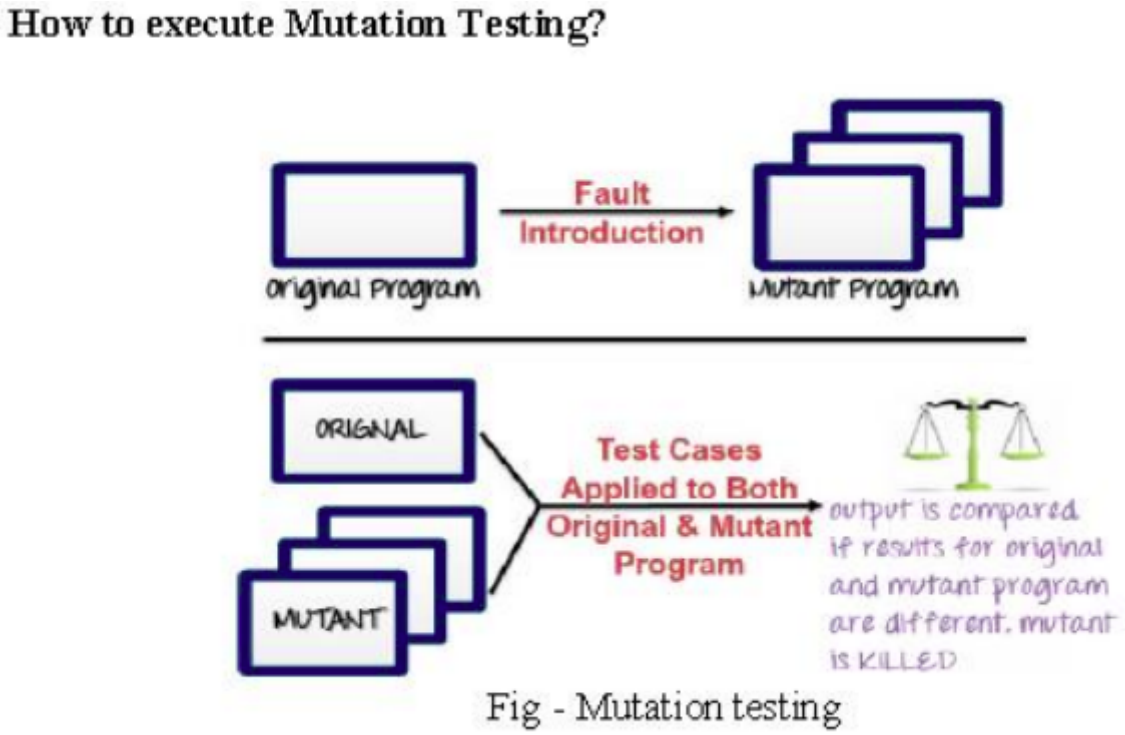


Fig - Mutation testing

-

Test Approach

- A test approach is the test stretegy implmentation
- two types of approach:
  - proactive: An approach in which the test deisgn process is initiated as early as possible in order to find and fix the defects before thr build is cretaed.
  - reactive: An approach in which the testing is not started until design and coding are completed.

Cost of Quality (COQ)

- cost of quality is a measure that quantifies the cost of conformance and cost of non-conformance. In other words it sums up the costs related to prevention and detection of defects and the cost due to occurance of defects.
- Accoridn to ISTQB cost of quality is nothing but prevention costs, appraisal costs, internal and external failure costs.
- Cost of control also known as cost of conformance

Some broad categories of documents that exist in every testing project:

- Test Policy: It explaints the goals that the organization wishes to acheive through testign activities.
- Test Stretegy: This explaints the stretegy of testing. Which activies, methods will be used.
- Master test plan: also called as project test plan. addresses project specific testing stretegy and test implemnetation.
- Level test plan: also called as phase test plan.

Software testing methods

- Black box testing:

    - In this type of testing the sructure/implementation/design of the software is not known to the tester.
    - These tests can be functional or non-functinal but mostly it is functional
    - test design techinuqes are equivalence partitioning, boundary value analysis, cause effect graphing.
- White box testing:
    - In this type of testing the sructure/implementation/design of the software is known to the tester.
    - techniues are control flo, data flow, branch, and path testing
- Gray box testing:
    - combination of black and white box testing
- Agile testing:
    - a mthod of testing that follows the principles of agile software development.
- Ad Hoc testing: A method of testing without any planning and documentation.

Structured approach:

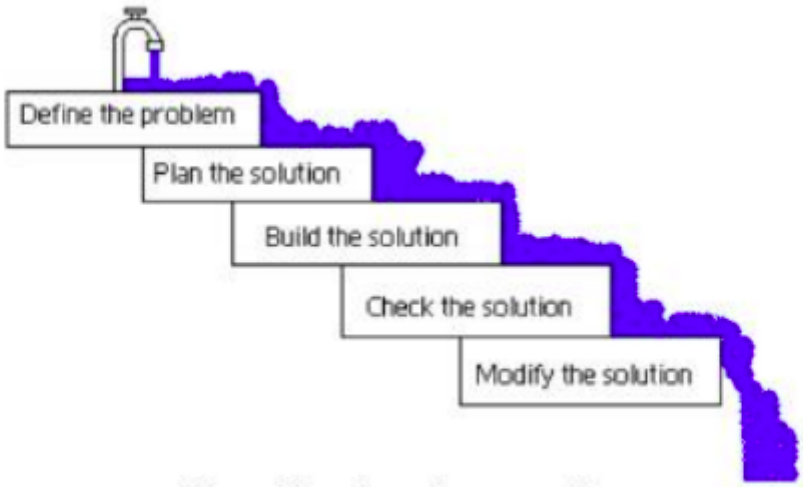
- In structured approach the big project is divided into smaller modules for development.
- process oriented approach
- 

Fig - Structured approach
- Here first stage is define the problem where basically the customer requirements are gatherd and analyszed.
- In the next phase the charts and design is planned. and the problem is decomposed into multiple sub modules.
- In the next phase which is build the solution, here the individual modules are coded and tested and then integrated together. this process of integration is called as software integration.
- Then check the solution involves acceptance testing, i.e. check wherther software is developed according to the client requirements or not.
- Modify the solution

Categories of defect

Severity basis: Here first we calculate the degree of impace of defect on the software and then we caegories into four categories based on the value of severity.

- Critical: This type of defect need urgent attention and treatment. The critical defect affect the main software functionality such as failure of a feature, bring down the whole system, etc.
- Major: these defect affect the major functionalities of the app but not such major which makes whole system down.
- Minor: defect which are occur on the client side however it it does not stops uses to execute it task.
- Trivial: small defects such as spelling and grammatical mistakes.

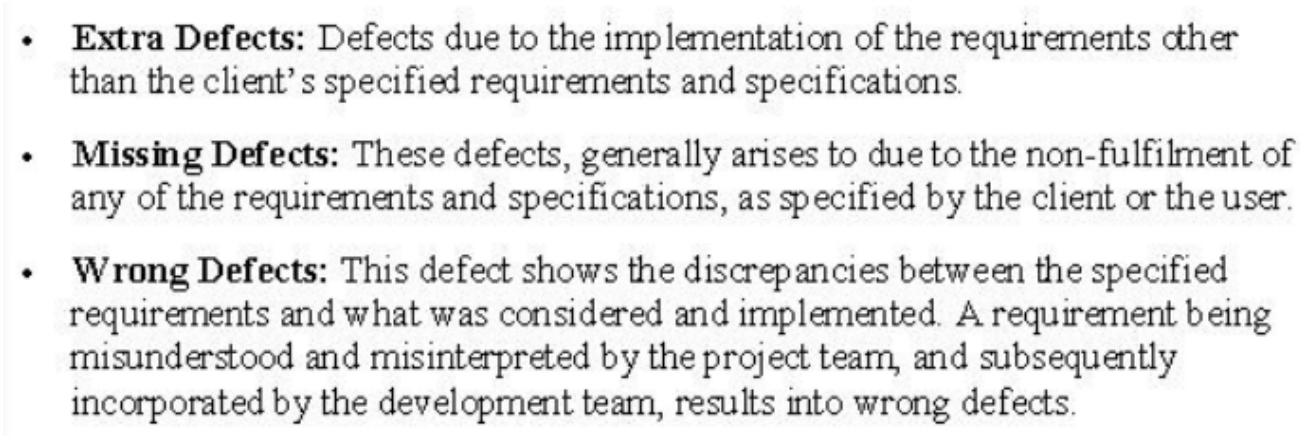
Probability Basis: Here we calculate the probability of defect occurance on the client side.

- High: high probability of getting traced out by almost all the users.
- Medium: half of the users
- Low: not detected by any user or detected by only few users

Priority Basis: Here defects can be categoriezed as per the buisness perspective. Like which defect needs to be corrected first and which is at later stage.

- High: correct a defect as soon as possible.
- Medium: may be addresssed in next version of product
- Low: It may or may not be corrected or considered.

Other defects:



Test Strategy Document Components

- Scope and Objective
- Business Issues
- Testing approach
- Test deliverables
- Defect tracking approach
- Training

Test plan Components

- Test plan ID: unique id of test plan. It can be number or name or mix of both.
- Test environment: testing envrionment such as device testing, virtual setup, etc,
- Features to be tested / Not tested
- Entry / Exit criteria: when to start or stop the testing
- Status: test case is passed or failed
- Types of testing: regression, functional, non-functional, stress, etc
- 100% requirements coverage: all business and techincal requirements have to be covered by testing.
- Minimum % pass rate: 90% of all test cases to be passed i sbest practice.

Unit 3

Test Automation:

- Software test automation makes use of specialized tool to contrl the execution of test cases adn compares the actual results with exepcted results.
- Useful for repetative regression tests.
- Used for both functional and non-functional testing

Approaches to test automation:

- Graphical user interface testing: GUI testing with mouse clicks and keystorkes.
- API driven testing: testing without UI. tests libraries, prgramming models, classes with differnr inputs parametrs.
- One way to generate test cases automtically is to use model based testing appproach. In some cases model based approach enables non-technical users to created aumated business test cases in plain english.

Terms used in automation

- Black box testing
- Acceptance testing: this is the final level of testing conducted by users with the purpose to accept or reject the system before release.
- Actual result: Result after testing
- Alpha testing: Alpha testing is performed by the potential users, test teams at vendor site. This group does not involve the people who actually developed the system. Alpha testing is sometimes used as acceptance testign by the vendor.
- Big-Bang integration: All the modules of the software are assembled and then tested together.
- Bottom up integration: an integration testing stretegy in which we start integrating from the lowest level of the system architecture.
- Anomaly: Any condition that deviates from the expectations. A good way to find annomalies is by testing the software.

Generation of Automation

- Capture/Playback and Test Harness tools:
  - Here a tester performes the test in a capture mode and then after capturing we can run these recorded test scripts.
  - A capture and playback tool can be intrusive or non-intrusive
  - intrusive capture/playbac tools are called as native tools as they along with software under test (SUT).
- Data-driven tools
  - In this technique test script are generated from the input and output list.
- Action-driven tools
  - fully automated testing
  - no exepcted input and output conditions required

Automation Architecture

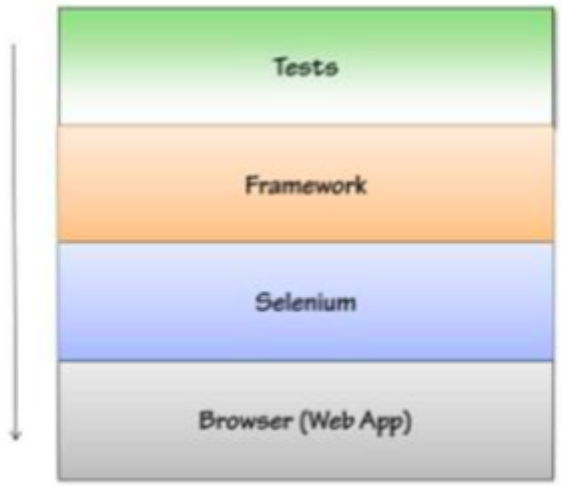


Fig - Architecture of automation

- Browser running our actual application which we want to test
- Selenium is the driver which is used for browser automation. It is just a API to interact with the browser autmatically.
- Framework is the actual code which will read our tests and perform the equivalent actions on the browser throgh selenium. Framework knows our UI of the applicaiton.
- Tests are the actual test scripts.

Types of tools

Types of Tools:

Sr.No.	Tool Type	Used for	Used by
1.	Test Management Tool	Test Managing, scheduling, defect logging, tracking and analysis	testers
2.	Configuration management tool	For Implementation, execution, tracking changes	All Team members
3.	Static Analysis Tools	Static Testing	Developers
4.	Test data Preparation Tools	Analysis and Design, Test data generation	Testers
5.	Test Execution Tools	Implementation, Execution	Testers
6.	Test Comparators	Comparing expected and actual results	All Team members
7.	Coverage measurement tools	Provides structural coverage	Developers
8.	Performance Testing tools	Monitoring the performance, response time	Testers
9.	Project planning and Tracking Tools	For Planning	Project Managers

Process model for automation

Fig - Process Model for Automation

- Which tool to select for the automatio is depend on the type of application.
- For ex QTP does not support informatics applications.

Data driven testing

Data-driven testing is creation of test scripts where test data and/or output values are read from data files instead of using the same hard-coded values each time the test runs. This way, testers can test how the application handles various inputs effectively. It can be any of the below data files.

- Data pools
- Excel files
- ADO objects
- CSV files
- ODBC sources

Flow Diagram:

Data Driven Testing can be best understood by the following diagram:

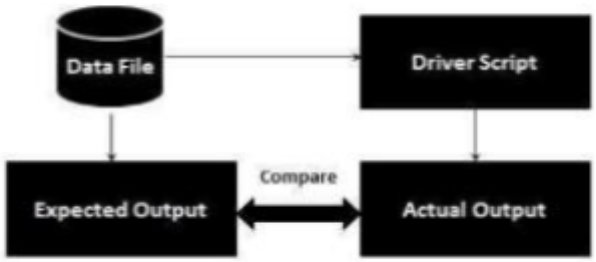


Fig – Data Driven Testing

JUnit and Jmeter

JUnit

- junit is the most popular unit testing framework for the java language.
- junit5 is th enext generation of Junit

JMeter

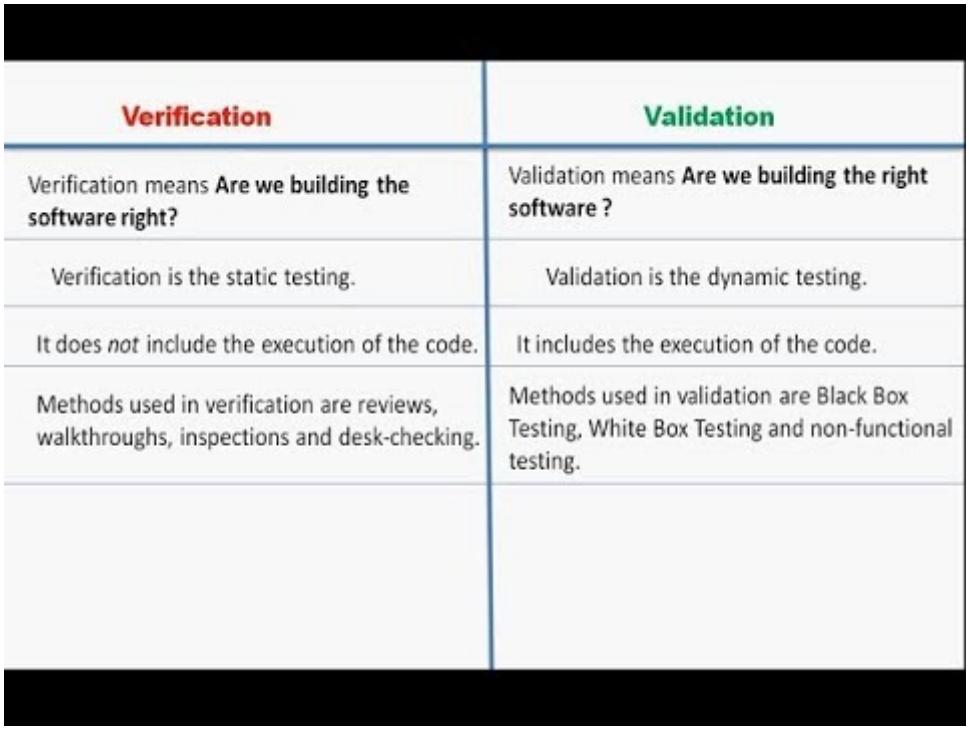
- jmeter is apache pure java open source software
- initally developed by stefano Mazzoochi
- Jmeter designed to load test funcional behaviour and measure performance.
- used for performance testing of web applications.
- nowadays it is used for a functional test, database server test, etc

Extra Concepts

Verification and Validation

- verification is whether we are developing the right product according to the customer requirements are not. Are we building the software right?
- validation is whether we have developed the product according to the customer requirements or not. Are we built the right product.

- validation is static whereas validation is dynamic



Verification	Validation
Verification means **Are we building the software right?**	Validation means **Are we building the right software ?**
Verification is the static testing.	Validation is the dynamic testing.
It does *not* include the execution of the code.	It includes the execution of the code.
Methods used in verification are reviews, walkthroughs, inspections and desk-checking.	Methods used in validation are Black Box Testing, White Box Testing and non-functional testing.

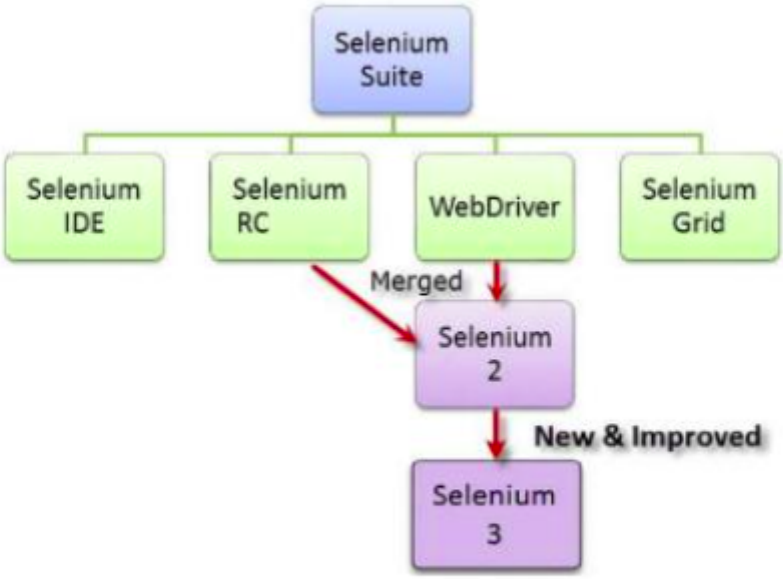
Quality control and Quality Assurance

- white box testing is also known as structural testing, glass box, transparent, clear box
- beta testing is also known as field testing

Unit 4

Selenium

- Selenium is a free open source testing suite for testing web application.
- It is same like HP QTP (Quick Test Pro) now named as UFT
- selenium is not just a single tool instead it is suite of software each one is used for diff testing needs of an organization.
- Four Components
    - Selenium IDE
    - Selenium Remote Control (RC
    - Web Driver
    - Selenium Grid



- Fig – Architecture of Selenium
- selenium supports to run one tests on diff browser platforms
- selenium first introduces in 2004 when Jason Huggins was tetsing an internal application at Thought Works.
- Jason huggins developed a javascript library which serve as a selenium core and by using that anyone can interact with the browser from the language of their choice.
- Because of the javascript libarry automation engine the browser implies various security and other restrictions.
- In 2006, engineer at google named Simon Stewart started work on a project called web driver.

- Simon wants the testing tool that spoke directly to the browser using the native method for the browser and OS thus avoiding the need of javascript environment.
- The web driver project began with the aim to solve the selenium pain points.
- In 2008 selenium core and selenium web driver merged together.

Selenium 1

- Selenium RC was the main selenium project for a long time.

Selenium 2

- Selenium RC + Web Driver = Selenium 2
- selenium 2 still runs selenium1 RC interface for backwards compatability.

Selenium IDE

- Selenium IDE is a prototyping tool for building test scripts.
- It is easily available as a plugin for chrome and firefox browsers.
- Also it has a recording feature by using that a tester can record the tests and then export them as a reusable script in one of many programmign languages that can be later executed.
- It is not designed to run all test passes also not designed to build all automated tests you will need.

Selenium RC

- Unfortunately, testers using Selenium Core had to install the whole application under test and the web server on their own local computers because of the restrictions imposed by the same origin policy.
- So another Thought Work's engineer, Paul Hamm ant, decided to create a server that will act as an HTTP proxy to "trick" the browser into believing that Selenium Core and the web application being tested come from the same domain. This system became known as the Selenium Remote Control or Selenium 1.

Selenium Grid

- selenium grid allows the selenium RC solution to scale for larget test suits and for test suits that must be run in mutliple environments.
- Selenium grid allows you to run your test in parallel that is differnet tests can be run at the same time on differetn remote machines.
- Selenium grid divides the large test suite and run the tests of test suite parallely on remote machines.
- Also we can run tests parallley by considering diff enevironmetn on the remote machines.

Unit 5

Software Quality

- In software engineering software quality refers to two related but distinct notations:
  - Software functional quality:
    - It is nothing but how well the software compiles with based on functional requirements or specifications.
    - It is the degree to which the correct software was produced.
  - Sofware structural quality
    - structural quality refer to how it meets non-functional requirements such as robustness and maintainability.
    - It is the degree to whcih the software works as needed.

Software Quality Dilemma

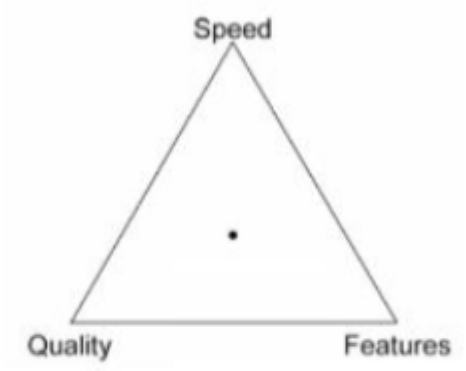
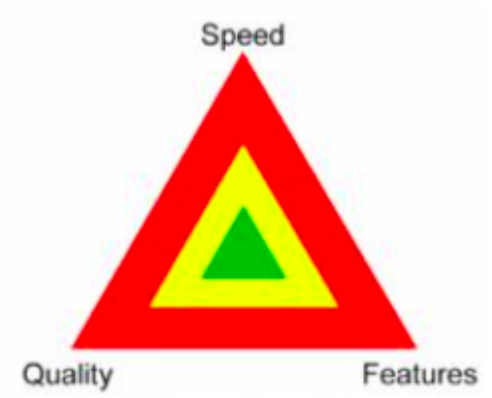


Fig1 - Software Quality Dilemma



- However it is recommended to stay at the center of the traingle.
- also when we move towards acheving the one target we are going away from the other targets. Like if we go towards speed then we are going far from the quality and features.

Achieving Software Quality

- Coverage testing helps the tester create a thorough set of tests and gives a measure of test completeness as well as also reverls the important aspects of software structures.
- ATAC (Automatic Tests Analysis for C language) is a dataflow coverage-testing tool.

Software Quality Assurance (SQA)

- SQA is a process which assures that all software engineering processes, methods, activities and work items are monitored and comply against the defined standards.

SQA Activities

- Creating an SQA management plan
- setting the checkpoints
- Apply software engineering techniques

SQA Elements

10 elements are there in th SQA

- risk management
- security management
- safety
- change management
- vendor management
- educational programs
- sofwtare engineering standards
- technical reviews and audits
- software testing for quality control
- error collection and analysis\

SQA Techniques

- There are several techniques for SQA, auditing is the most popular SQA technique.
- Auditing: involves inspection of the work products and its realted information to determin if the set of standard processes were followed or not.
- Reviewing: here the software product is examined by both the internal and external stackholders.
- Code Inspection: formal kind of review to find the bugs or defects in the product.
- Design Inspection

SQA Tasks

SQA Task 1:

The Engine Software Engineer will check with the Requirements Specification on a weekly basis to make sure that what he is coding conforms to the original design. This process will ensure that the product meets the client’s expectations and standards and that the engine, up to its current point, is working properly.

SQA Task 2:

The User-Interface Software Engineer will check with the Requirements Specification on a weekly basis to make sure that what he is coding conforms to the original design. This process will ensure that the

product meets the client’s expectations and standards and that the user-interface, up to its current point, is working properly.

SQA Task 3:
Each member of the group will routinely perform a hands-on evaluation of the user-interface. Noted evaluation criteria will be: ease of use, principle of least astonishment, unobtrusiveness, and overall attractiveness.

SQA Task 4:
Each member of the group will routinely perform a hands-on evaluation of the DirectX engine.

SQA Task 5:
An SQA leader will be appointed to (1) control the frequent SQA reviews; (2) keep track of all SQA meetings; and (3) manage the flow of information to the correct software engineer.

Statistical Software Qualitity Assurance

- SQA is used to identify the potential variations in the manufacturing process and predict potential defects on a parts-per-million (PPM) basis.
- It provides a statistical description of the final product and addresses quality and safety issues that arise during manufacturing.

Statistical quality assurance methodologies:

- Force diagram
- Test to failure (TTF): TTF tells manufacturers on how many defects they are likely to find in every million units of output.
- Intervention: Here products are separetd into groups based on the production quantity or production lines. then z value is calculated and based on that value manufactureres can make the decisions about improving the particular production line.

Six sigma for software engineering

- Six sigma is nothing but the set of techniques and tools for process improvement.
- introduced yb Bill Smith in 1980 while working at Motorola.
- Six Sigma is the process of producing high and improved quality output.
- This can be done in two phases – identification and elimination. The cause of defects is identified and appropriate elimination is done which reduces variation in whole processes.
- A six sigma method is one in which 99.99966% of all the products to be produced have the same features and are of free from defects.
- Denoted by greek later - standard deviation

Six sigma methodologies

- DMAIC
  DMAIC is used to enhance an existing business process. The DMAIC project methodology has five phases:

1. Define
2. Measure
3. Analyze
4. Improve
5. Control

- DMADV
  DMADV is used to create new product designs or process designs. The DMADV project methodology also has five phases:

1. Define
2. Measure
3. Analyze
4. Design
5. Verify

ISO 9000 Quality Standards

- The ISO 9000 series was created by International Organization of Stadardization (ISO) as requirements and guidelines for quality management systems.
- Originally introduced in 1987
- ISO 9000 series standards are process standards not product standards.
- ISO 9000: Fundamentals and vocabulary
- ISO 9001: requirements that organization must comply to become ISO 9001 certified
- ISO 9002: Guidlines for the application of ISO 9001:2015
- ISO 9004: Guidelines to acheive sustained success (Continuous Improvement)
- ISO 9001 lists requirements while all other standards in the 900- family provide guidelines and information.
- People often say ISO 9000 certified but what they mean is they have met the requirements of the ISO 9001 standard.
- The ISO 9000 series is not industry specific and is applicable to any manufacturing distrbution and service organization.
- Managed by Technical Committee (TC) 176

SQA Plan

- The SQA plan is a document that specifies the process to be followed in each step of the software development and the procedures to be followed in each activity of such a process.
- The SQA plan is governed by several quality standadards and policies such as ISO 9000, SEI CMM, Baldrige.

- SQAP comprises of the procedures, techniques, and tools that are employed to make sure that a product or service aligns witht the requirement defined in the SRS.

SQA Plan document consists of the below sections:

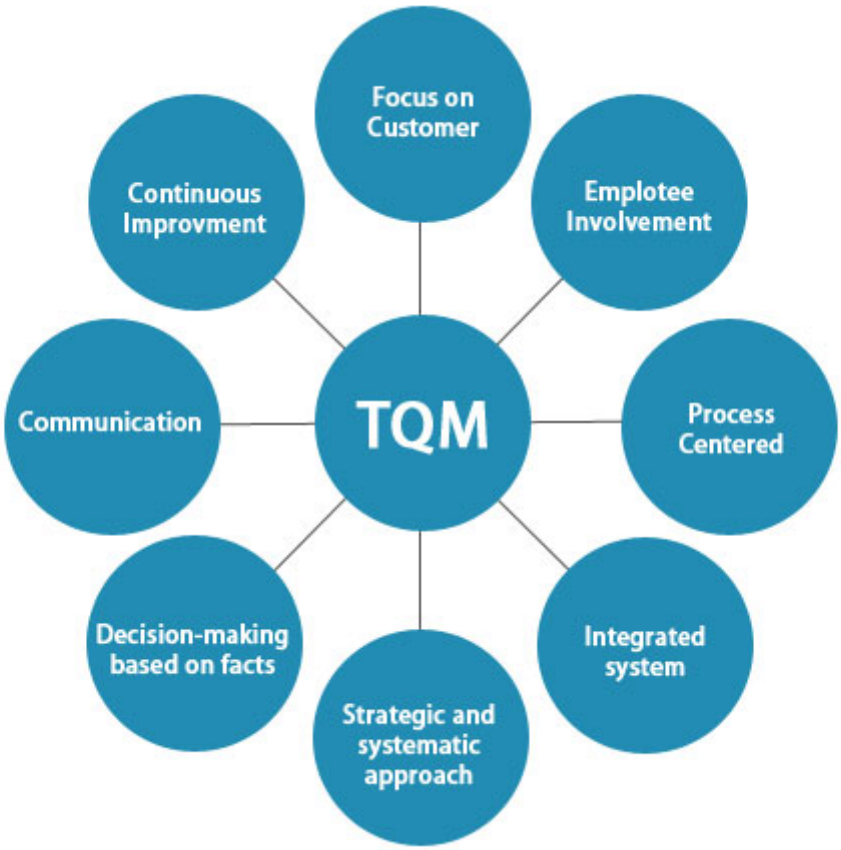
- purpose secction
- reference section
- software configuration management section
- problem reporting and correction action section
- tools, technologies section
- code control section
- Records: collection, maintanence and retention section
- Testing methodology

Unit 6

Total Quality Management (TQM)

- TQM is the structured approach that focuses on continuous quality improvement of products and services by using continuous feeback.
- Joseph Juran and William Deming are the foundars of TQM
- TQM originated in the industrial sector of japan

Basic Priciples of TQM



Focus on customer

- Focus on customer because customer is the only person determine the level of quality.

Employee Involvement:

- Employee are an organization internal customer. Employee inolvement in the products and service development largly determines the quality of these products and services.

Process Oriented:

- Process thinking and process handling are a fundamental part of TQM

Integrated System:

Strategic and Systematic approach:

- A strategic plan must embrace the integration and quality development.

Decision making based on facts:

- decision making should be based on the facts not on the opinions.

Communication:

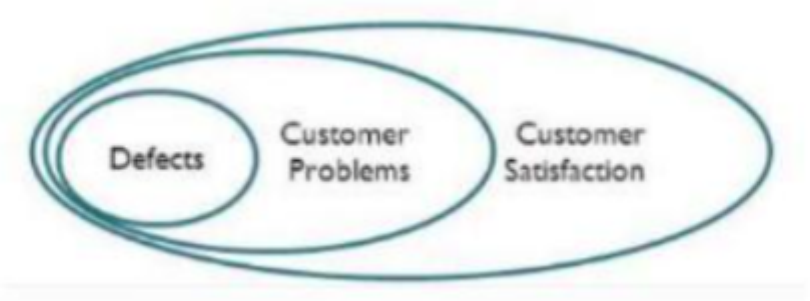
- A communication strategy must be formulated in a such a way that it is in line with mission, vision and objectives of the organization.

Continuous Improvement:

- By using the right measuring tools, innovative and creative thinking. continuous improvement proposals will be initiated and implmented so that the organization can develop into a higher level of quality.

Product Quality Metrices

- The customer problems can be regarded as an intermediate masurment between defects and customer satisfaction.



- Intrinsic product quality is usually mesured by no of bugs in the software or how long the software can run before encountering a crash.
- In operations definitions the two metrices are defect density (rate) and mean time to feailure (MTTF).
- MTTF metric is most often used with safety critical systems like missiles control, airline traffic controls, etc. Ex. US govenment mandates that airline traffic contrl system cannot be unavailable for more then theree seconds per year.

- Defect Density is the number of defects confirmed in software/module during a specific period of operation or development divided by the size of the software/module.
- Defect density is counted per thousand lines of code also known as KLOC

Difference between Error, fault, failure and defect

- An error is a human mistake that resukts in incorrect software.
- A failure occures when a function unit of a software system can no longer perform its required function or cannot perform it within specified limits.
- A defect is an anomaly in a product.
- A fault is an accidental condition that causes a unit of the system to fail to fucntion as required.
- Generallt defect and fault are same.
- MTTF (Mean Time to Failure)
- MTBF (mean Time Between Failure)
- MTTR (Mean Time to Repair)

Defect Removal Effectiveness

DRE = Defects removed during a development phase / defect latent in the product X 100%

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

Software Maintainence

- It stands for all the modification and updations done after the delivery of software product.
- There can be various resons for the modifictions:
  - Market conditions
  - Client requirements
  - Host modifications
  - Organization changes

Maintenance Activities

- Identification and Tracing
- Analysis
- Design
- Implementation
- System testing
- Acceptance testing
- delivery
- Maintenance Management

Ishikawa's 7 basic tools

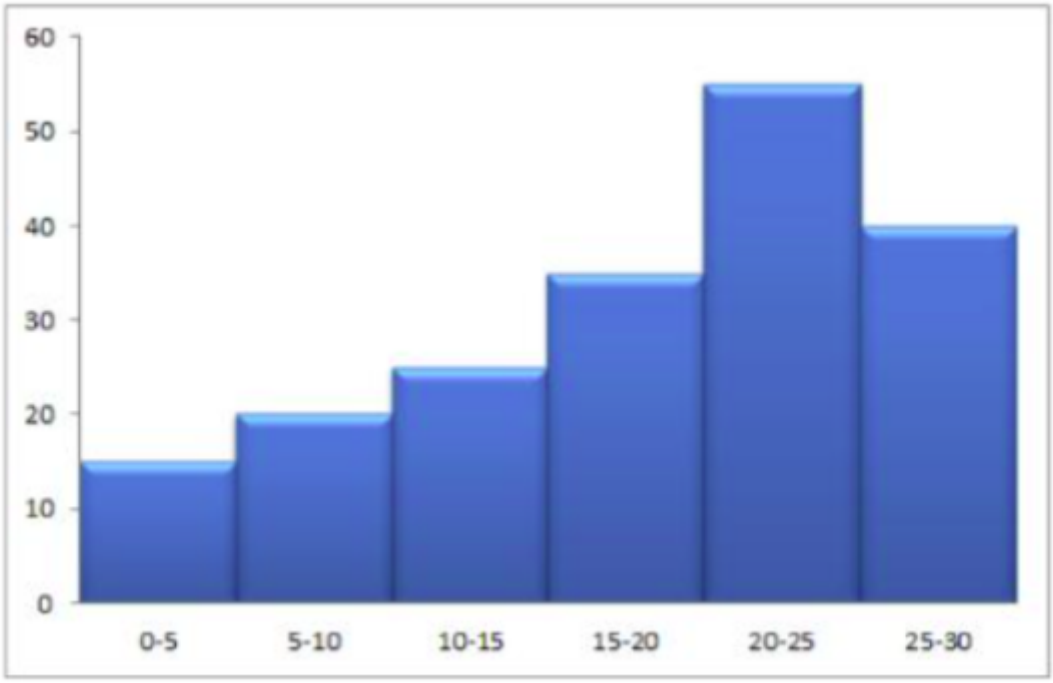
- These are the 7 basic QC tools also called as seven basic tools of quality.
- originated in Japan
- Used in various phases of six sigma (DMAIC or DMADV), in continuous improvement (PDCA) and lean management.

Tools

1. Histogram

- Histogram is the first tools introduced by Dr. Ishikawa
- It is a graph that represnts the frequenct (count) of items falling into different categories of a given population or sample.
- In this graph on Y axis we have frequency and on X axis we have individual items or intervals.
- Histogram is used to represent continuous data.
- It is same like bar chart but here there is no gap between bars to signify continuous data.
- Width of the groups = total range of population / no of categories

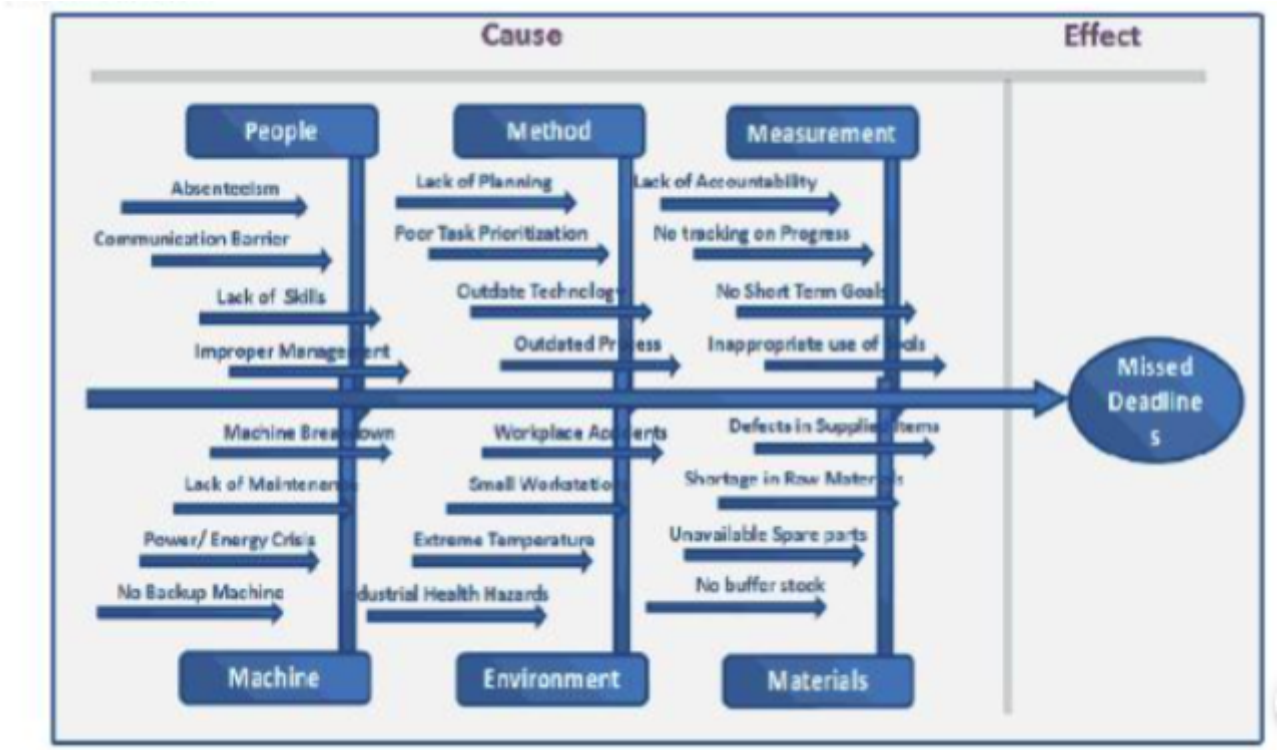
$$Width = \frac{Total\ range\ of\ population}{No.\ of\ categories}$$



2. Cause and Effect Diagram

- As histogram is used to view the type of variation but in order to find why that variation occures for that we use cause effect diagrams. Second tool created by Ishikawa's.
- Cause effect diagram helps in identifying the causes and the effect of that causes and also helps in deriving meaningful relationship between them.

- Generally in cause effect diagram we get the problem then we find its casual factors then in turn we find their sub factors until we reach to the root cause. As a result we get a diagram wiht branches and sub-branches of casual factors resembling to a fish bone diagram.
- 4M's, 1E and 1P
- People
- Methods
- Machines
- Material
- Measurements
- Environment



3. Check sheets

- If you wanted to track reasons for defective items, you developed a check sheet and tracked the reasons over time on that sheet.

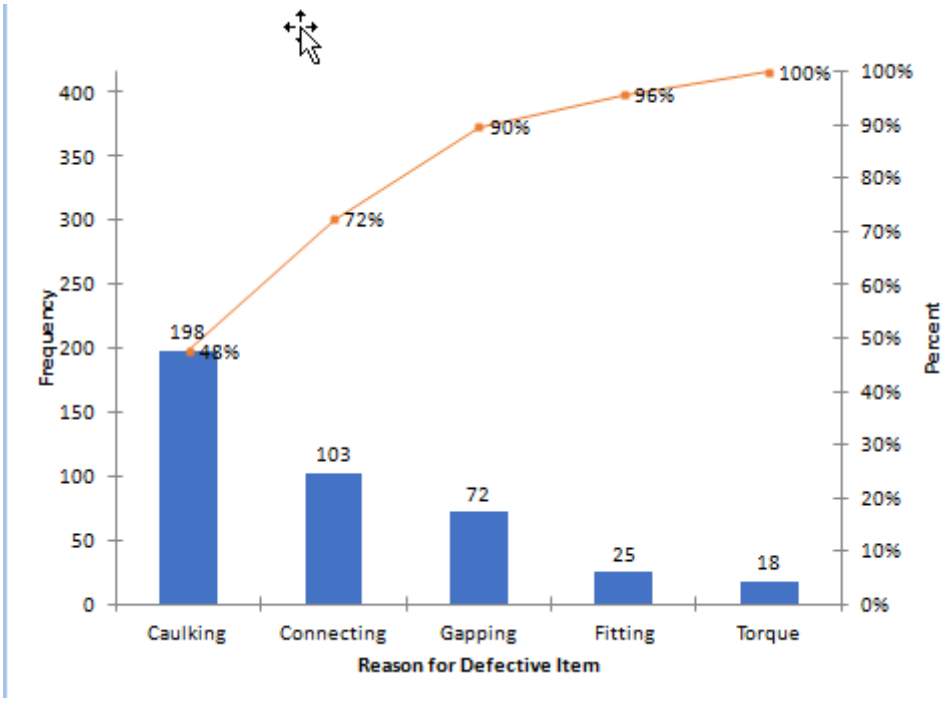
Type	Check	Subtotal
Surface scars	////////////////////////////////	32
Cracks	//////////////////////	23
Incomplete	//	48
Misshapen	////	4
Others	///////	8
	Grand Total	115
Total Rejects	// //////////////////////////////////	86

- The reasons for defects are listed on the left-hand side. Each time a defect occurs, a tick mark is placed in the column for the reason for the defect. When the product has finished being inspected, the defects are totaled and the total placed in the last column.

- When the information collected is quantitative in nature the check sheet also called as tally sheet.

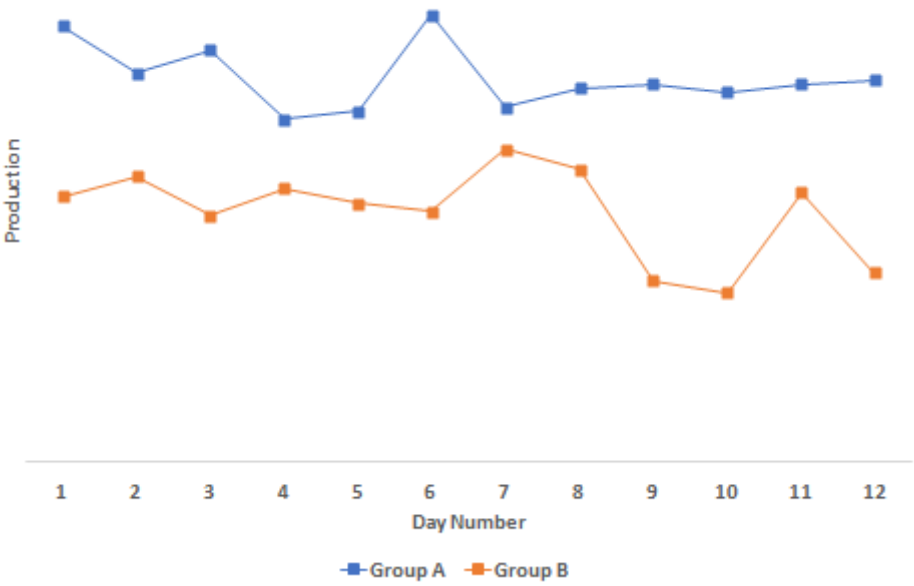
4. Pareto Diagrams

- Fourth tool
- A Pareto diagram is a bar chart that is used to help separate the “vital few” problems from the “trivial many” problems.
- It is a data-based approach to help decide what problem to work on first.
- The pareto principle also known as the 80-20 rule derived from the Italian Wilfred Pareto's.



5. Graphs

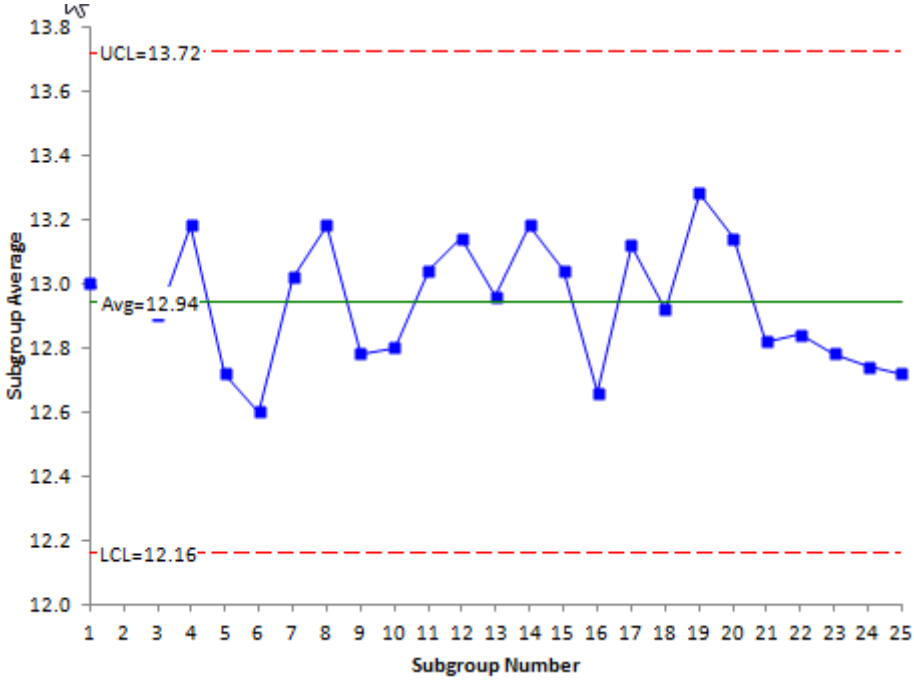
- Fifth tool
- It contains various types of graphs



- Above graph is stratification graph

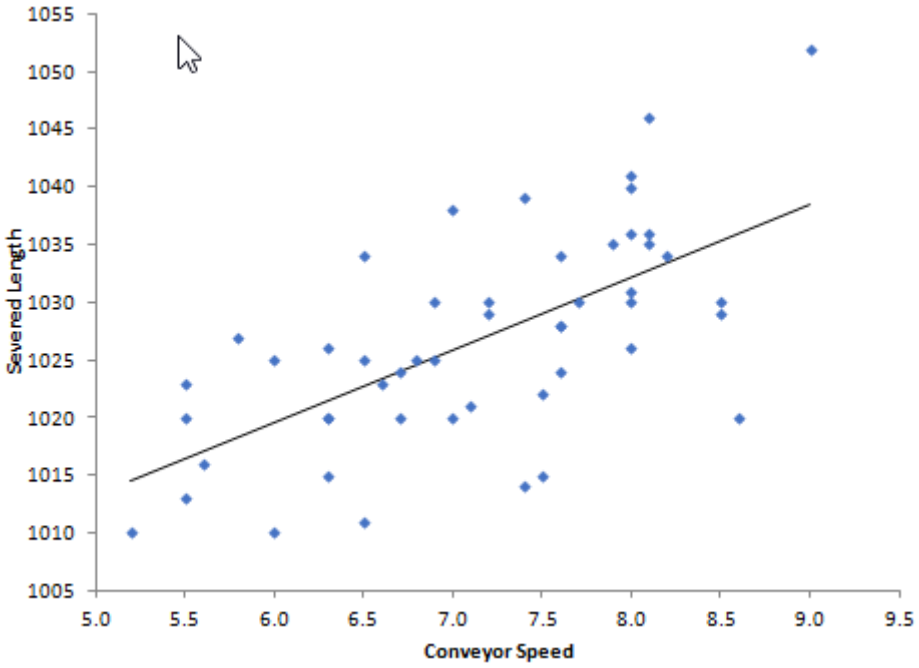
6. Control Charts

- Control chart is statistical tool used to monitor whether a proces is in control or not.
- It is a time series graph with the process mean at center and the control limits on both sides of it.
- The values lying outside the control limits show that the process is out of control.
- The variation of the process can be attributed to two causes:
    - Common cause: Common cause is the cause in which no external factors is associated. very difficult to reduce this type of variation.
    - Special Cause: variation cause by factos that are not a part of the process. These are noramally outliers and can be easily detected.



7. Scatter Diagram

- A scatter diagram shows the relationship between these two types of data.



Defect Removal Effectiveness and Process Maturity Level

The defect removal effectiveness for organizations at diff levels of the development process capability maturity model (CMM):

- Level 1: 85%
- Level 2: 89%
- Level 3: 91%
- Level 4: 93%
- Level 5: 95%