

* Software Quality

- 1) Software quality assurance is simply a way to assure quality in the software.
- 2) Software quality assurance is a process which works parallel to development of software.
- 3) SQA is a set of activities for ensuring quality in software engineering process that ultimately results in the quality of software products.
- 4) The SQA is a process in which conformed to the requirement of the product is met.
- 5) Process that ensures that development software meets & complies with defined or standardized quality specifications.

* Quality assurance and Quality control

Quality Assurance:

- 1) It checks if the implemented techniques & approaches are good to assure quality in software.
- 2) It also finds the approaches, techniques to assure the quality in software.
- 3) It deals with the process used to develop the software and try to assure the quality by selecting the quality process, approach & technique.
- 4) Its all about the process & techniques to assure the quality.

Quality Control:

- 1) Quality Control is the process in which we verify if we achieved the required quality by applying all the process or not.

- » It is the process after the QA process mostly.
- » It also checks if all the processes required to achieve and assure the quality are implemented correctly.
- » It checks the product we get by following the approaches and try to fix the bugs if defects as well.

Quality assurance

defn QA is a set of activities for ensuring quality in the process by which products are developed

Quality control

QC is a set of activities for ensuring quality in products. The activities focus on identifying defects in the actual product produced

focus QA aims to prevent defect QC aims to identify defect with a focus on the process used to make the product. It is proactive quality process. QC aims to identify defects in the finished product. Quality control, therefore, is a reactive process.

Goal) The goal of QA is to improve development & test processes so that defects do not arise when the product is being developed

The goal of QC is to identify defects after a product is developed & before it is released.

How) Establish a good management system & the assessment of its adequacy. Periodic conformance audits of the operations of the system

Finding & eliminating sources of quality problems through tools & equipments so that customers requirements are continually met.

QA

What Prevention of quality problems through planned and systematic activities including documentation

The activities or techniques used to achieve & maintain the product quality, process for

Response Everyone on the team involved in the development of the product is responsible for quality assurance

Quality control is usually the responsibility for a specific team that tests the product for defects.

Example Verification is an eg. of QA

Validation / software testing is an example of QC

Statistical tools & techniques can be applied to both QA & QC when they are applied to process they are called statistical process control (SPC) & become a part of QA

When statistical tools & techniques are applied to finished product, they are called statistical quality control (SQC) & becomes under QC

As a tool QA is a managerial tool QC is a corrective tool

* Software Quality metrics:

They are classified into following categories

1) Product metrics - Describes the characteristics of the product such as size, complexity, design features & quality level.

2) Process metrics : These characteristics can be used to improve the development & maintenance activities of the software.

3) Project metrics: These metrics describe the project characteristics of execution

↳ Software quality metrics focus on the quality aspects of the product, process, & project.

↳ These are more closely associated with process & product metrics than the project metrics.

↳ Software quality metrics can be further divided into three categories

↳ Product quality metrics

↳ In-process " "

↳ Maintenance " "

⇒

Product Quality metrics include the following

↳ Mean time to failure

↳ Defect density

↳ Customer problems

↳ Customer satisfaction

* Software Process standards

↳ Process improvement model

↳ Process is a set of activities that an organization does. Holding organizations people & technology.

↳ CMM is applied to the areas of development, services & acquisitions

↳ Development: companies which manufacture products

↳ Services: Service sector

↳ Acquisition: companies which are in outsourcing.

CMM: Capability Maturity Model

level 1: Processes are unpredictable, poorly controlled, reactive.

level 2: Processes are planned, documented, performed, monitored and controlled at the project level, often reactive. Every project follows some standards.

level 3: Processes are well characterized & understood, Processes, standards, procedure tools etc. are defined at organizational level.

level 4: Processes are controlled using statistical & other quantitative techniques.

level 5: Processes performance continually improved through incremental & innovative technological improvements.

PCMM: People Capability Maturity Model

- ▷ It is a maturity framework that focuses on continuously improving the management & development of the human assets of an organization.
- ▷ It describes an evolutionary improvement path from of the workforce that enhance strategic business performance.
- ▷ The people CMM document describes the practices that constitute each of its maturity levels & information on how to apply it in guiding organizational improvements.
- ▷ It describes an organization's capability for developing its workforce at each maturity level.
- ▷ It also describes how the people CMM can be applied as a standard for assessing workforce practices or as a guide in planning &

Implementing improvement activities

Difference b/w ISO & CMM

ISO	CMM
1) It is applied to any industry	2) Specially designed for software industry
3) ISO 9000 addresses corporate business process	3) Focuses on software engineering activities.
3) ISO 9000 specifies minimum requirements	3) Gets into technical aspect of SE
4) ISO 9000 provides pass or fail criteria	4) Provides grades.
5) ISO 9000 has no levels	5) CMM has 5 levels

* Software Testing.

Introduction

- 1) It is a process used to identify the correctness, completeness and the quality of developed software.
- 2) It also helps to identify errors, gaps or missing requirements.
- 3) It can be done either manually or by using some software tools.

• General characteristics of strategic testing:

- 1) To perform effective testing, the Software team should conduct technical reviews.
- 2) Testing begins at component level & work

towards the integration of entire computer based system.

- 3) Different testing techniques are appropriate at different points in time.
- 4) Testing is done by developer of software and for large projects by an independent test groups.
- 5) Testing and debugging are different activities, but debugging must accommodate in any test strategy.
- 6) In other words, software testing is verification + validation process.

* Principles of software Testing :

There are seven principles of software testing.

- 1) Testing shows presence of defects
 - 2) Exhaustive testing is impossible
 - 3) Early testing
 - 4) Defect clustering
 - 5) Pasteille Paradox
 - 6) Testing is content dependent
 - 7) Absence of error-fallacy.
-
- a) Testing shows presence of defects:
 - i) The goal of testing is to make software fail. Sufficient testing reduces the presence of defects
 - ii) The effective testing can show the defects that are present in software, but it cannot prove that the software is defect free.

b) Exhaustive testing is impossible.

- 1) Testing all functionalities using all valid and invalid inputs and preconditions is known as exhaustive testing.
- 2) For eg:- In an application, one screen there are 15 fields having 5 possible values, then too many valid combinations are present.
- 3) Accessing and managing the risk areas be most important for testing any project.

c) Early testing :-

- 1) Defects detected in the early phases of SDLC are less expensive to fix
- 2) It is cheaper to change incorrect requirement compared to fixing bug in fully developed functionality.

d) Defect clustering :-

- 1) Small module of functionalities contains most of the bugs or it has the most operational failures.
- 2) As per the 80-20 rule, 80% of issues comes from 20% of modules.

e) Pesticide paradox :-

- 1) It is the process of repeating the same test cases again and again.
- 2) It will not find the new bugs.
- 3) So, to overcome this pesticide paradox, it is necessary to review the test

case regularly and add or update them to find more defects

6) Testing is context dependent :-

- 1) Testing approach depends on the context we develop
- 2) We test the software differently in different contexts.
- 3) eg.: online banking application requires a different approach of testing compared to e-commerce site.

7) Absence of Error-fallacy

- 1) 99% of bug free software may still be unusable, if we incorporated wrong requirements into the software and the software is not addressing the business needs.
- 2) The software which we built should not only be bug free but it should also be addressing the business needs.

* Types of testing

- 1) Unit Testing :- It focuses on testing each module component independently based on implementation.
- 2) It compares the actual results with the results defined in the specification & design of module.
- 3) Integration testing :- components are integrated together step by step to form a system.

complete software.

- 3) validation testing :- Requirements established are validated against the development software
- 4) System testing :- The software is tested with other system elements as a whole.
- 5) Regression testing :- In addition of a new module as a part of integration testing, may cause problems with the function that previously worked flawlessly.
 - ↳ System testing ensures that all system elements have integrated well with the software and the entire system functions properly.
- 6) Acceptance testing :- Conducted when software is developed for a specific customer & not for large public / audience.
 - ↳ Allows customer to validate all requirements.
- 7) Alpha & Beta Testing :- Conducted when the product is developed for anonymous customer.
 - ↳ For large public
 - ↳ carried by the customer.

Beta testing

Alpha testing

- ▷ Done by the customer by the developer's site
- ▷ Conducted in controlled environment
- 3) Developer present? : Yes
- 4) carried out before the release of the product to the customer
- 5) Errors / failures are recorded
- 6) white & black box testing

- ▷ Done by the customer at the user/customer site
- ▷ conducted in real time environment of the user. Not under the developer's control.
- 3) Developer present?
- 4) carried out after the release of product to customer.
- 5) Failures are reported
- 6) Black box testing

* verification & validation

Verification

- ▷ Are we building the system right?
- 2) It is a process of defining a system or component to determine if the products of a development phase satisfy the conditions imposed at the start of that phase

Validation

- ▷ Are we building the right system / software?
- 2) It is the process of evaluating a system / component during or at the end of development process to determine whether it satisfied the specified requirement

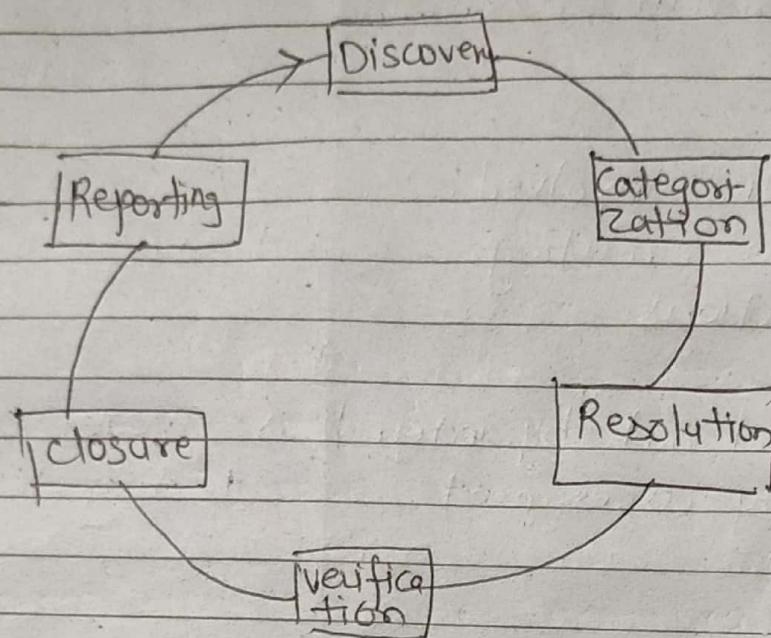
3) Applied during the development phase	3) Applied towards the end of the development phase.
4) Includes manual testing: look & review the document	4) Program execution takes place: requirements are validated.
5) Activities involved: Reviews, meetings, inspections	5) Activities involved: testing techniques
6) carried out by internal quality assessment team	6) carried out by testing team
7) Done prior to validation	7) Done after validation

* Defect Management:

- 1) A defect is a variation or derivation from the original business requirement.
- 2) A bug is the consequence / outcome of a coding fault.
- 3) These two terms have a very thin line of difference, in the industry both are faults that needs to be fixed.
- 4) When a tester executes the test case, he might come across the test result which is contradictory to expected result. This variation in the test result is referred as a software defect.
- 5) These defects or variations are referred by different names in different

organizations like issue, problem, bug or incidents.

Defect management Process



Discovery :- > In the discovery phase, the project teams have to discover as many defects as possible, before the end user can discover it

2) A defect is said to be discovered & change of status accepted when it is acknowledged & accepted by the developer

Categorization :- > Defect categorization help the software developers to prioritize their tasks

3) That means that this kind of priority helps the developers in fixing those defects first that are highly crucial

Resolution:- Once the defects are accepted and categorized, follow the following steps to fix the defect

▷ Assignment:- assigned to a developer or other technician to fix, and changed the status to responding.

2) Schedule fixing :- The developer side takes charge in this phase. They will create a schedule to fix these defects, depend on the defect priority.

3) Fix the defect :- While the development team is fixing the defects, the test manager tracks the process of fixing defect compare to the above schedule.

4) Report the resolution:- Get a report of the resolution from developers from when defects are fixed.

Verification:- > After the development team has fixed & reported the defect, the testing team verifies that the defects are actually resolved.

Closure:- > Once a defect has been resolved and verified, the defect will change the status as closed.

2) If not, you have to send a notice to the development team to check the defect again.

- Reporting :-
- 1) The management board has the right to know the defect status.
 - 2) They must understand the defect management process to support you in the project.
 - 3) Therefore, you must report them the current defect situation to get feedback from them.

* Bug Report

While reporting the bug to developer, your Bug Report should contain the following information

- 1) Defect ID :- Unique identification number for the defect.
- 2) Defect description :- Detailed description of the defect including information about the module in which defect was found.
- 3) Version - Version of the application in which defect was found.
- 4) Steps :- Detailed steps along with screenshot with which the developer can reproduce the defect.
- 5) Data raised :- Data when the defect is raised.
- 6) Reference :- Where do you provide reference to the documents like requirements, design, architecture or maybe even screenshots to the error to help understand the defect.
- 7) Detected by :- Name / ID of the tester who raised the defect.
- 8) Status - status of the defect.
- 9) Fixed by - Name / ID of the developer who fixed it.

- 103 Data closed - Data when the defect is closed
- 114 Severity - which describes the impact of the defect on the application,
- 107 Priority - which is related to defect fixing urgency.