Difference between black box testing and white box testing

Sr. No.	Black box testing	White box testing
1.	Black box testing is called behavioural testing.	White box testing is called glass box testing.
2.	Black box testing examines some fundamental aspect of the system with little regard for internal logical structure of the software.	In white box testing the procedural details, all the logical paths, all the internal data structures are closely examined.
3.	During black box testing the program cannot be tested 100 percent.	White box testing lead to test the program thoroughly.
4.	This type of testing is suitable for large projects.	This type of testing is suitable for small projects.

Fig. 3.5.1 Testing strategy

1. **Unit testing** - In this type of testing techniques are applied to detect the errors from each software component individually.
2. **Integration testing** - It focuses on issues associated with verification and program construction as components begin interacting with one another.
3. **Validation testing** - It provides assurance that the software validation criteria (established during requirements analysis) meets all functional, behavioural and performance requirements.
4. **System testing** - In system testing all system elements forming the system is tested as a whole.

3.5.3.1 Unit Testing

- In unit testing the individual components are tested independently to ensure their quality.
- The focus is to uncover the errors in design and implementation.
- The various tests that are conducted during the unit test are described as below.

1. Module interfaces are tested for proper information flow in and out of the program.
2. Local data are examined to ensure that integrity is maintained.
3. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing.
4. All the basis (independent) paths are tested for ensuring that all statements in the module have been executed only once.
5. All error handling paths should be tested.
6. Drivers and stub software need to be developed to test incomplete software. The "driver" is a program that accepts the test data and prints the relevant results. And the "stub" is a subprogram that uses the module interfaces and performs the

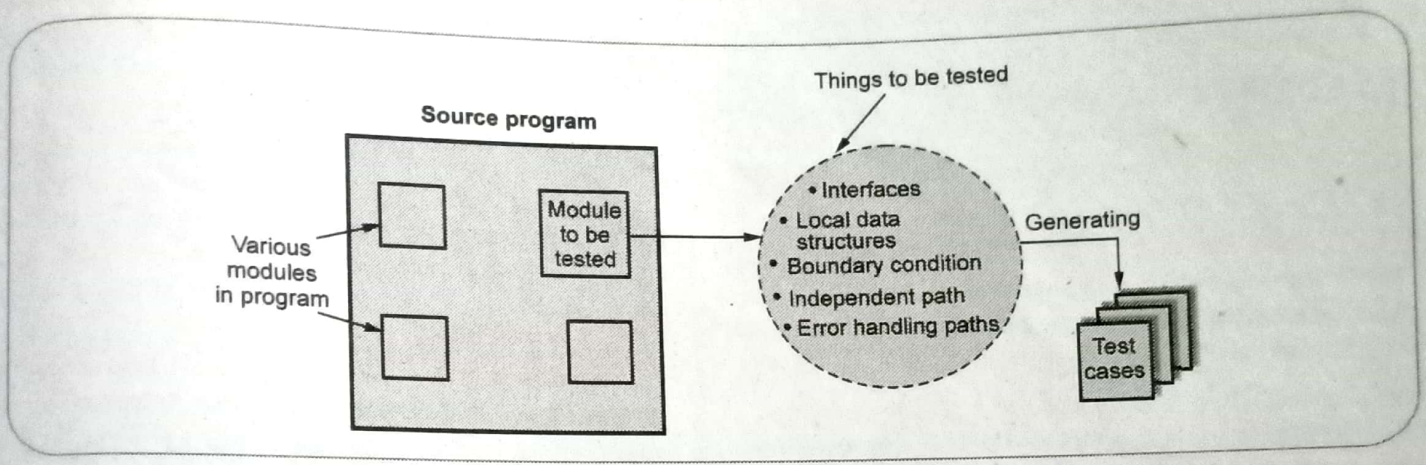


Fig. 3.5.2 Unit testing

minimal data manipulation if required. This is illustrated by following Fig. 3.5.3.

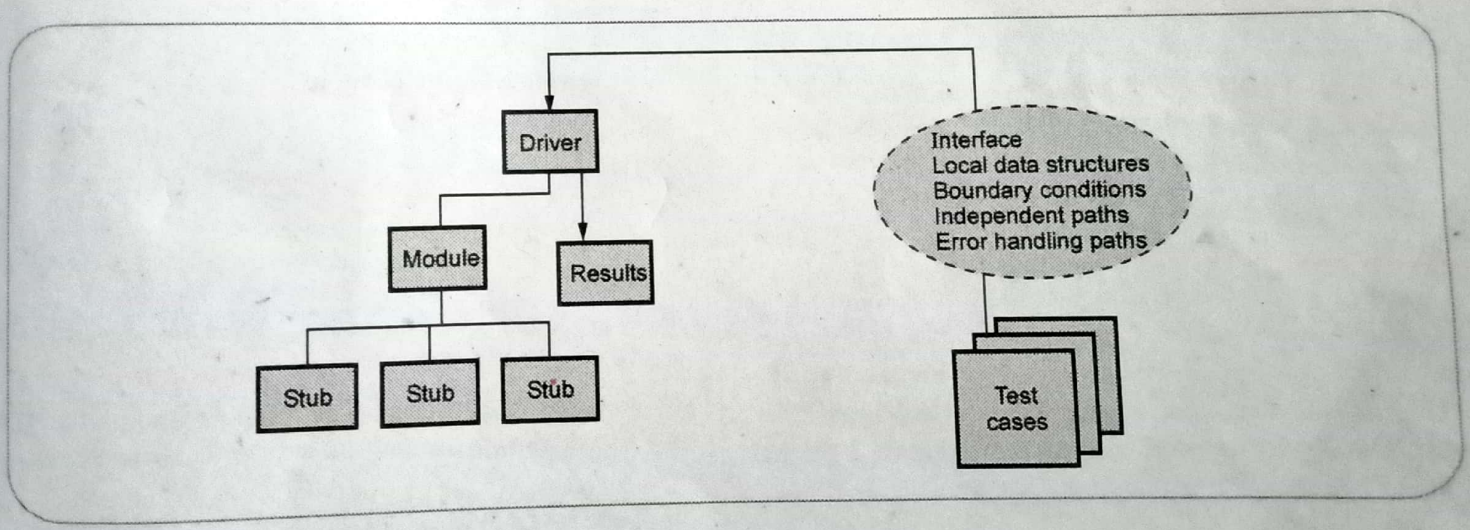


Fig. 3.5.3 Unit testing environment

7. The unit testing is simplified when a component with high cohesion (with one function) is designed. In such a design the number of test cases are less and one can easily predict or uncover errors.

4.1 Management Spectrum

MSBTE : Winter-16, Summer-18

Effective software project management focuses four P's i.e. **people, product, process and project.** The successful project management is done with the help of these four factors where the order of these elements is not arbitrary. Project manager has to motivate the **communication between stakeholders.**

(1) Product objectives an established.

(2) Alternative solutions shou

(3) Technical and managem be identified.

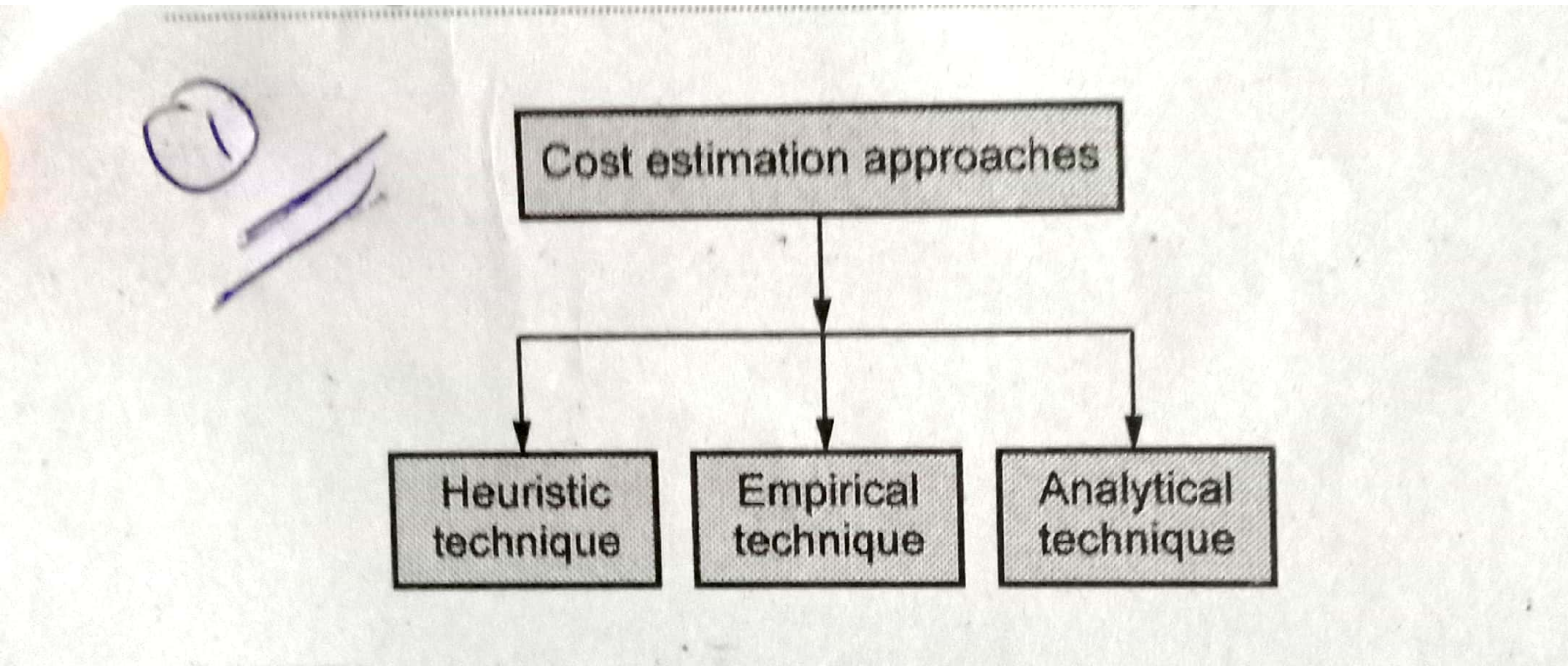
- The software developer communicate with each other

is well designed but shorter program may get suffered. Programming language.

3. It does not accommodate non procedural languages.
4. In early stage of development it is difficult to estimate LOC.

4.2.2 Function Points

- The oriented model is based on functionality of the delivered application.
- These are generally independent of the programming language used.
- This method is developed by Albrecht in 1979 for IBM. It uses function points.
- Function points are derived using :
  1. Countable measures of the software requirements domain
  2. Assessments of the software complexity.



4.6 Risk Management

MSBTE : MSBTE : Summer-15, 16, 17, 18, Winter-15, 16

Definition of risk : The risk denotes the uncertainty that may occur in the choices due to past actions and risk is something which causes heavy losses.

- Another approach of risk analysis is making analysis of various things like -
  - Outcome of various decisions(Decision analysis)
  - Risks on various factors such as reliability and usability(quality factor analysis)
  - Performance constraints(performance analysis).

Risk Prioritization

- After risk analysis the impact of each risk on the project can be analysed. Based on this impact risks can be prioritised.
- **Risk exposure** computing is done for prioritising the risks. Risk exposure is also called as **risk impact.**
- The risk exposure can be calculated by following formula,

Risk Exposure = Probability of occurrence of risk *Loss due to unsatisfactory outcome

- Thus risk exposure for each risk from risk table is calculated. The total risk exposure of all risks helps in determining the final cost of the project.

4.6.5 Risk Containment

- Risk containment means reduction of risk.
- The project manager and team will be able to identify strategies to minimize or eliminate the risk factors.
- For example - If a project is facing high risk due to lack of experience in development platform, then the recruiter or hiring expert contractor can control this risk by hiring the skilled and experienced employee for the desired project.
- Lot of high risk factors can be eliminated or reduced during the risk assessment.

4.6.6 RMMM Strategy

RMMM stands for **risk mitigation, monitoring** and **management**. There are three issues in strategy for handling the risk is

1. Risk avoidance
2. Risk monitoring
3. Risk management.

Risk mitigation

Risk mitigation means preventing the risks to occur(risk avoidance). Following are the steps to be taken for mitigating the risks.

1. Communicate with the concerned staff to find of probable risk.
2. Find out and eliminate all those causes that can create risk before the project starts.
3. Develop a policy in an organization which will help to continue the project even though some staff leaves the organization.
4. Everybody in the project team should be acquainted with the current development activity.
5. Maintain the corresponding documents in timely manner. This documentation should be strictly as per the standards set by the organization.
6. Conduct timely reviews in order to speed up the work.
7. For conducting every critical activity during software development, provide the additional staff if required.

Risk monitoring

In risk monitoring process following things must be monitored by the project manager,

1. The approach or the behaviour of the team members as pressure of project varies.
2. The degree in which the team performs with the spirit of "team-work".
3. The type of co-operation among the team members.
4. The types of problems that are occurring.
5. Availability of jobs within and outside the organization.

The project manager should monitor certain mitigation steps. For example.

If the current development activity is monitored continuously then everybody in the team will get acquainted with current development activity.

The **objective** of risk monitoring is

1. To check whether the predicted risks really occur or not.
2. To ensure the steps defined to avoid the risk are applied properly or not.
3. To gather the information which can be useful for analyzing the risk.

Risk management

Project manager performs this task when risk becomes a reality. If project manager is successful in applying the project mitigation effectively then it becomes very much easy to manage the risks.

For example, consider a scenario that many people are leaving the organization then if sufficient additional staff is available, if current development activity is known to everybody in the team, if latest and systematic documentation is available then any 'new comer' can easily understand current development activity. This will ultimately help in continuing the work without any interval.

Board Questions

1. *What is risk projection ? Enlist steps of risk projection.* **MSBTE : Summer-15, Marks 4**
2. *Describe RMMM strategy in detail.* **MSBTE : Summer-15,Winter-15, Marks 4**
3. *What is software risk ? Explain types of software risks.* **MSBTE : Summer-16, Marks 4**
4. *Describe the following with respect to Risk assessment :*
   i) Risk identification *ii) Risk analysis*
   iii) Risk prioritization **MSBTE : Winter-16, Marks**
5. *Explain following terms w.r.t. risk management :*
   i) Risk identification *ii) Risk analysis*
   MSBTE : Summer-17, Marks
6. *Enlist and explain different types of Softwar Risks. (four points)*
   MSBTE : Summer-18, Mark

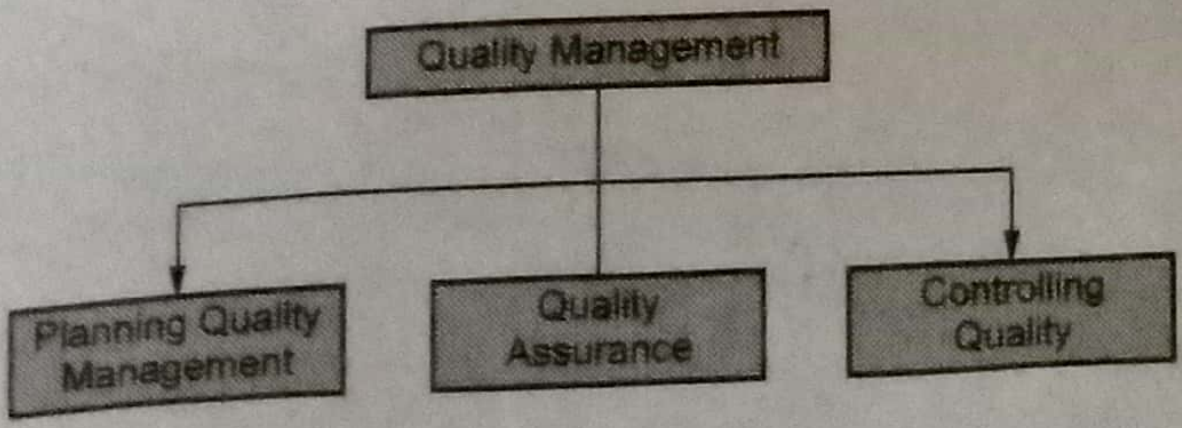
Difference between PERT and CPM

Sr. No.	PERT	CPM
1.	PERT is a project management technique used to manage uncertain activities of project.	CPM is a statistical technique of project management that manages well defined activities.
2.	This technique of planning and control of time.	A method to control cost and time.
3.	This technique is event oriented.	This technique is activity oriented.
4.	It is non repetitive in nature.	It is repetitive in nature.

5.3 Software Quality Management Vs. Software Quality Assurance

- **Definition of Quality :** The International Organization for Standardization (ISO) defines quality as the totality of characteristics of an entity that bear on its ability to satisfy stated or implied needs.
- The project quality management is a process which ensures that the project will satisfy the needs for which it was undertaken.
- The main principle of project quality management is to ensure the project will meet or exceed stakeholder's needs and expectations.
- The project team must develop a good relationship with key stakeholders.
- The project quality management is performed using following three key processes.

1. Planning Quality Management
2. Performing Quality Assurance
3. Controlling Quality



MSBTE : Winter-15, 16, 17, Summer-15, 16, 17, 18, Marks 8

5.6.1 Six Sigma

Six sigma is widely used statistical software quality assurance strategy. It is a business driven approach to process improvement, reduced costs and increased profit. The word "six sigma" is derived from six standard deviations - 3.4 defects per million occurrences. Six Sigma originated at Motorola in the early 1980s.

There are three core steps in six sigma method -

Define - The customer requirements, project goals and deliverables are defined by communicating the customers.

Measure - The existing process and its output is measured in order to determine current quality performance.

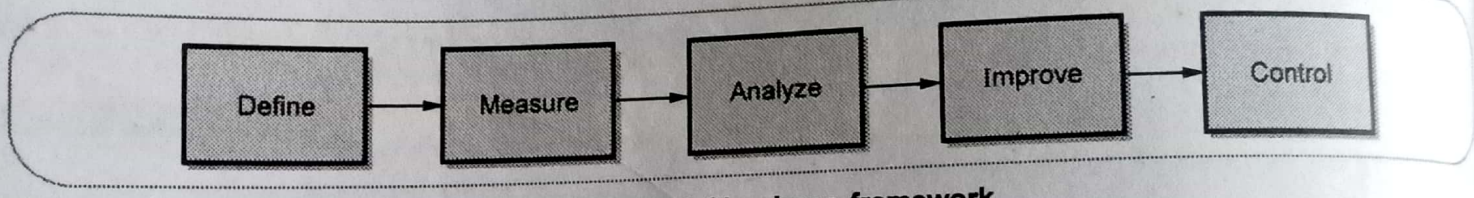


Fig. 5.6.1 Six sigma framework

Analyze - In this phase defect metrics are analyzed in order to determine the few causes.

If an improvement is needed to an existing software then there are additional two methods in six sigma -

Improve - By eliminating the root causes of defects the process can be improved.

Control - The process can be controlled in such a way that the causes of defects can not be reintroduced.

These steps can sometimes be referred as DMAIC.

For a newly developing software, some organizations are suggesting following two alternating steps -

Design - In this step avoid root causes of defects and meet the customer requirements.

Verify - To verify the process, avoid defects and meet customer requirements.

These steps can sometimes be referred as DMADV.

5.6.2 ISO for Software

operation. This process is called registration to **9000.**

- On successful registration, the company certification from accreditation bodies of ISO. company is then called **"ISO certified** compa
- **ISO 9001:2000** is a quality assurance standard is applied to software engineering systems.
- It focuses on process flows, customer sati and the continual improvement of management systems.
- ISO 9001:2000 specifies requirements for system that can be applied to any size o organization.
- The guideline steps for ISO 9001:2000 are
  - *Establish quality management system* and manage the processes in the management system.
  - *Document the quality management system*
  - *Support the quality*
  - *Satisfy the customers*

- *Make quality improvement*

- The ISO 9000 helps in creating organisational **quality manuals**. These quality manuals identify the organisational quality processes.
- Using these quality manuals, the project quality plan can be prepared for every individual project. Thus project quality management can be done.

This is illustrated by following Fig. 5.6.2.

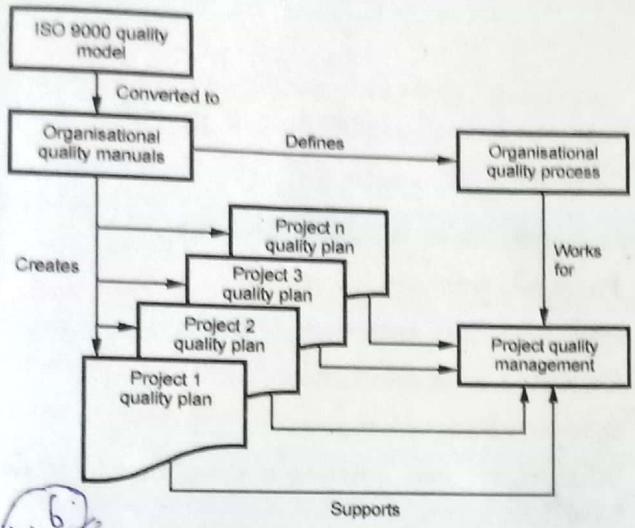


Fig. 5.6.2 ISO 9000 helps in quality management

5.6.3 CMMI

- The Software Engineering Institute (SEI) has developed a comprehensive **process meta-model** emphasizing process maturity. It is predicated on a set of system and software capabilities that should be present when organizations reach different levels of process capability and maturity.
- The Capability Maturity Model (CMM) is used in assessing how well an organization's processes allow to complete and manage new software projects.
- Various process maturity levels are

Level 1 : Initial - Few processes are defined and individual efforts are taken.

Level 2 : Repeatable - To track cost schedule and functionality basic project management processes are established. Depending on earlier successes of projects with similar applications necessary process discipline can be repeated.

Level 3 : Defined - The process is standardized, documented and followed. All the projects use documented and approved version of software process which is useful in developing and supporting software.

Level 4 : Managed - Both the software process and product are quantitatively understood and controlled using detailed measures.

Level 5 : Optimizing - Establish mechanisms to plan and implement change. Innovative ideas and technologies can be tested.

Thus CMM is used for improving the software project.

Comparison between ISO and SEI CMM Models

Sr. No.	ISO	CMM
1.	ISO 9001 addresses minimum criteria for an acceptable quality system.	In CMM, emphasis is on continuous process improvement.
2.	ISO 9001 focuses on hardware, software, processed material and services.	CMM focuses strictly on software.
3.	The basis of ISO 9001 is : "Say what you do and do what you say."	The basis of CMM is : "Say what you do and do what you say".
4.	Servicing activities are considered as separate maintenance process in ISO.	CMM does not have maintenance as a separate process.
5.	ISO 9001 maintains a quality record document which clearly shows whether or not required quality is achieved. This document also indicates whether or not existing quality system operates effectively.	The CMM emphasizes the need to record information for later use in the process and for improvement of the process.

Secure Software Engineering Life Cylce

- A Software Development Life Cycle (SDLC) is a framework that defines the process used by organizations to build an application. Over the years, multiple standard SDLC models have been proposed (Waterfall, Iterative, Agile, etc.) and used in various ways to fit individual circumstances.
- In general, SDLCs include the following phases:
  1. Planning and requirements.
  2. Architecture and design.
  3. Test planning.
  4. Coding.
  5. Testing and results.
  6. Release and maintenance.
- In the past, it was common practice to perform security-related activities only as part of testing. This technique usually resulted in a high number of issues discovered too late.
- It is better to integrate security concerning activities across the SDLC to help discover and reduce vulnerabilities early, effectively building security in.
- A Secure SDLC process ensures that security assurance activities such as penetration testing, code review, and architecture analysis are an integral part of the development effort.
- The primary advantages of secure software development Life cycle is -
  (1) More secure software as security is a continuous concern.
  (2) Awareness of security considerations by stakeholders.
  (3) Early detection of flaws in the system.
  (4) Cost reduction as a result of early detection and resolution of issues.
  (5) Overall reduction of business risks for the organization.

❑❑❑