Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

IT - ITeS SSC
NASSCOM

Participant Handbook

Sector

IT-ITeS

Sub - Sector

IT Services

Occupation

Test Engineer

Reference ID: **SSC/Q7001, Version 2.0**
NSQF Level 4

Test Engineer

Scan this QR Code to access eBook
https://eskillindia.org/Home/handbook/1430

Published by

IT – ITeS Sector Skill Council NASSCOM
Sector Skill Council Contact Details:
Address: Plot No. – 7, 8, 9 & 10 Sector – 126, Noida, Uttar Pradesh – 201303
New Delhi – 110049
Website: www.sscnasscom.com
Phone: 0120 4990111 – 0120 4990172

Second Edition, August, 2022

IT – ITeS Sector Skill Council NASSCOM
Sector Skill Council Contact Details:
Address: Plot No. – 7, 8, 9 & 10 Sector – 126, Noida, Uttar Pradesh – 201303
New Delhi – 110049
Website: www.sscnasscom.com
Phone: 0120 4990111 – 0120 4990172
This book is sponsored by IT – ITeS Sector Skill Council NASSCOM

Disclaimer

> Skilling is building a better India. If we have to move India towards development then Skill Development should be our mission.

Shri Narendra Modi
Prime Minister of India

Skill India
कौशल भारत-कुशल भारत

IT - ITeS SSC
NASSCOM

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

Certificate

COMPLIANCE TO
QUALIFICATION PACK- NATIONAL OCCUPATIONAL STANDARDS

is hereby issued by the

IT-ITeS Sector Skills Council NASSCOM

for

SKILLING CONTENT: PARTICIPANT HANDBOOK

Complying to National Occupational Standards of
Job Role/ Qualification Pack: **'Test Engineer'**
QP No. **'SSC/Q7001,** NSQF Level **4'**

Date of Issuance : January 27th, 2022
Valid Up to* : January 27th, 2025

*Valid up to the next review date of the Qualification Pack or the 'Valid up to' date mentioned above (whichever is earlier)

Authorised Signatory
(IT-ITeS Sector Skills Council NASSCOM)

Acknowledgments

This participant's handbook meant for Test Engineer is a sincere attempt to ensure the availability of all the relevant information to the existing and prospective job holders in this job role. We have compiled the content with inputs from the relevant Subject Matter Experts (SMEs) and industry members to ensure it is the latest and authentic. We express our sincere gratitude to all the SMEs and industry members who have made invaluable contributions to the completion of this participant's handbook. We would also like to thank all the experts and organizations who have helped us by reviewing the content and providing their feedback to improve its quality.

This handbook will help deliver skill-based training in the field of Test Engineer. We hope that it will benefit all the stakeholders, such as participants, trainers, and evaluators. We have made all efforts to ensure the publication meets the current quality standards for the successful delivery of QP/NOS-based training programs. We welcome and appreciate any suggestions for future improvements to this handbook.

About this book

This participant handbook has been designed to serve as a guide for participants who aim to obtain the required knowledge and skills to undertake various activities in the role of a Test Engineer. Its content has been aligned with the latest Qualification Pack (QP) prepared for the job role. With a qualified trainer's guidance, the participants will be equipped with the following for working efficiently in the job role:

- **Knowledge and Understanding:** The relevant operational knowledge and understanding to perform the required tasks.
- **Performance Criteria:** The essential skills through hands-on training to perform the required operations to the applicable quality standards.
- **Professional Skills:** The Ability to make appropriate operational decisions about the field of work.

The handbook details the relevant activities to be carried out by a Test Engineer. After studying this handbook, job holders will be adequately skilled in carrying out their duties according to the applicable quality standards. The handbook is aligned with the following National Occupational Standards (NOS) detailed in the latest and approved version of Test Engineer QP:

1. SSC/N1301: Design tests for software products/ applications/ modules
2. SSC/N1302: Carry out automated tests on software products/applications/modules
3. SSC/N1303: Carry out manual tests on software products/ applications/modules
4. SSC/N9001: Manage your work to meet requirements
5. SSC/N9002: Work effectively with colleagues
6. SSC/N9003: Maintain a healthy, safe and secure working environment
7. SSC/N9004: Provide data/information in standard formats
8. SSC/N9014: Implement & Improve the Gender Sensitivity, PWD (Person/People with Disability) Sensitivity and Greening

The handbook has been divided into an appropriate number of units and sub-units based on the content of the relevant QP. We hope it will facilitate easy and structured learning for the participants, allowing them to obtain enhanced knowledge and skills.

Symbols Used


Key Learning Outcomes


Exercise


Notes


Unit Objectives


Activitiy

Table of Contents

Table of Contents

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

IT - ITeS SSC
NASSCOM

1. IT-ITeS/IT Support Services Industry – An Introduction

Unit 1.1 - Overview of the Indian It-IteS Industry

Unit 1.2- Career Opportunities for Test Engineers and Job Responsibilities

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Explain various delivery models used in the IT/ software Products development industry.

UNIT 1.1: Overview of the Indian IT-ITeS/IT Industry

Unit Objectives

By the end of this unit, participants will be able to:

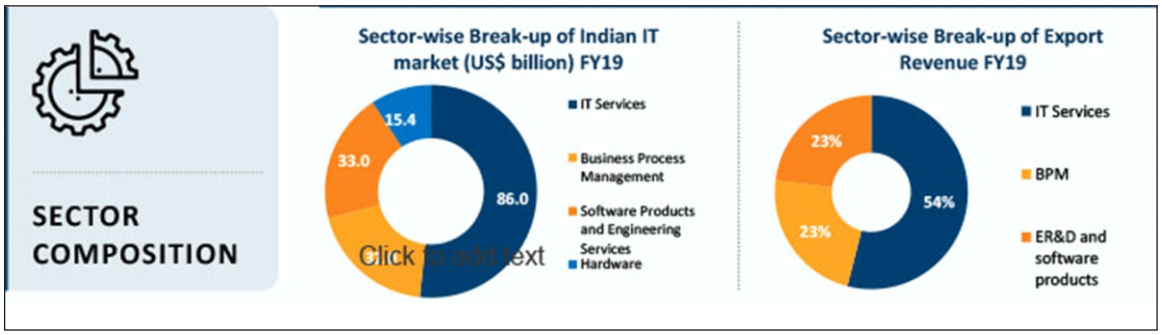
1. Explain the relevance of the IT-ITeS sector.
2. Outline the future of the IT-ITeS industry.
3. Conduct an Internet search to collect data, evidence, and articles pertaining to IT-ITeS/support services.

1.1.1 India's IT-ITeS/Test Engineers Industry

- Information Technology (IT), Information Technology Enabled Services (ITes)/ Business Process Management (BPM) are vital to the Indian economy.
- The IT and BPM market accounts for 9.3% of India's GDP and 56% of the global outsourcing market.
- India's IT and business services market is projected to reach US$ 19.93 billion by 2025.
- According to an estimate, IT spending in India is forecasted to increase to US$ 101.8 billion in 2022 from US$ 81.89 billion in 2021.
- India's IT & BPM industry is well-diversified across verticals, such as Banking, Financial Services, and Insurance (BFSI) sector, telecom and retail.
- In FY21, India ranked third worldwide with 608,000 cloud experts across all verticals, including technology.
- The computer software and hardware sector in India attracted cumulative foreign direct investment (FDI) inflows worth US$ 81.31 billion between April 2000 and December 2021.
- IT companies are one of the top employers in the country's organized sector.

Source: www.ibef.org/industry/information-technology-india

Sector Composition



Source: www.ibef.org/industry/information-technology-india/infographi

Fig. 1.1.1 Sector Composition of the Indian IT Market

It has been noticed that the IT Services and ITeS-BPO industries have impacted the Indian economy's growth. The Indian IT/ITeS industry has become one of the country's greatest success stories, putting it on the worldwide map as a leader in Information Technology (IT) and Business Process Outsourcing (BPO). In every way, the Indian information technology (IT) and information technology-enabled services (ITeS) industries are intertwined. The industry has not only improved India's global image. However, it has also fueled economic progress and contributed significantly to social transformation. With its low cost, large resource pool, and competence, India has the opportunity to tap into a booming market.

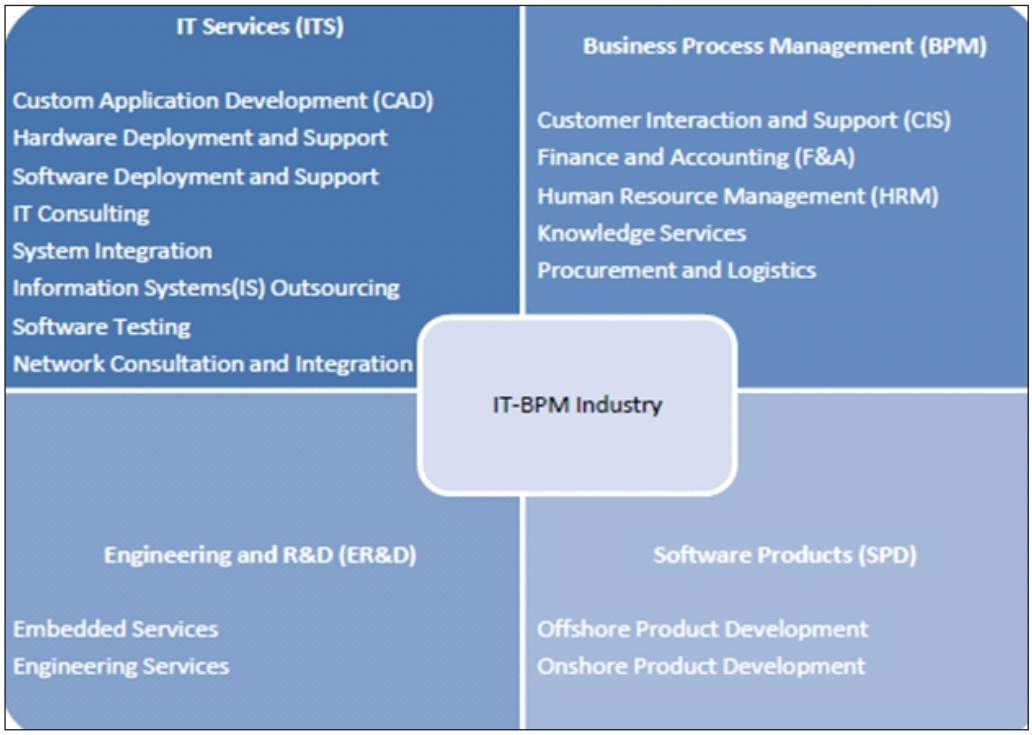


Fig. 1.1.2 Structure of the IT-BPM Industry

1.1.2 IT-ITeS Industry

IT-ITeS or Information Technology - Information Technology Enabled Services mainly comprises different operations and usage of information technology to improve an organization's efficiency. Two components stand as strong pillars to the IT industry. They are:

- IT Services
- BPO (Business Process Outsourcing)

Important Information related to the Indian IT Sector

- As per the total foreign direct investment share of India, the rank of the Indian IT sector is #3 on a global level. *(Source: https://telanganatoday.com/it-sector-ranks-3rd-indias-fdi).*
- The contribution from Indian venture investments and private equity is nearly 37% in the IT sector.

- The last recorded contribution of this sector to the Indian GDP was 7.7 percent (2017-2018).
- The aggregated revenues from the Indian IT sector in 2017 were estimated to be around US$160 billion. *(Source: NASSCOM).*
- If compared with the US, the Indian IT and ITeS sector is nearly 5 to 6 times cheaper and offers a low-cost advantage to companies.
- The delivery centres of Indian IT firms are well-diversified and spread globally. The verticals include Retail, Telecom, and BFSI (Banking, Financial Services, and Insurance).

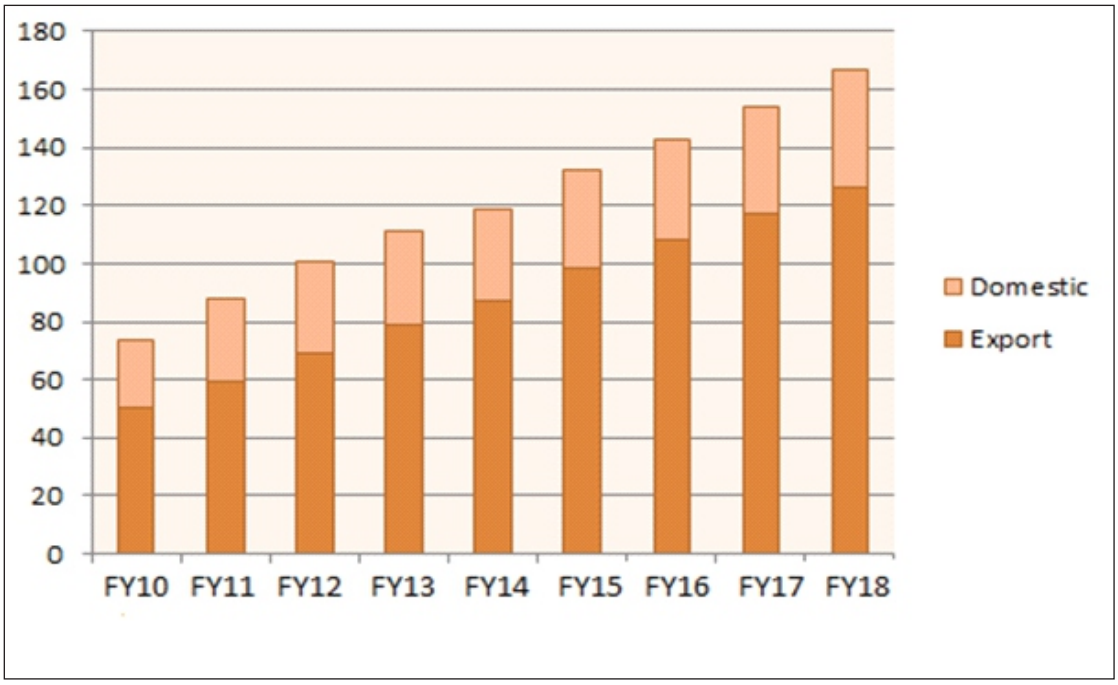


Fig.1.1.3 Market Size of the Indian IT-ITeS industry in India

1.1.3 Outlook and Beyond - Future of IT-ITeS Industry

This section highlights the predictions of prosperity that IT industry sees itself in the coming years.

- The IT industry sees the past 3 – 5 years as its "Golden Period."
- It is because of the growth in the IT-BPM sector, the CAGR or compound annual growth rate in FY17 (the financial year 2017) grew to 11.14%.
- The denotation of this percentage growth seemed to jump from the US $ 74 billion to the US $ 154 billion in only 7 financial years (FY10-17). This is basically 3 – 4 times higher as compared to other times.
- As per the constant and consistent growth in the IT sector, the growth of the industry, by 2025, is anticipated to be the US $ 350 million.
- The employment growth in the IT industry is forecasted to increase by 7- 8%.

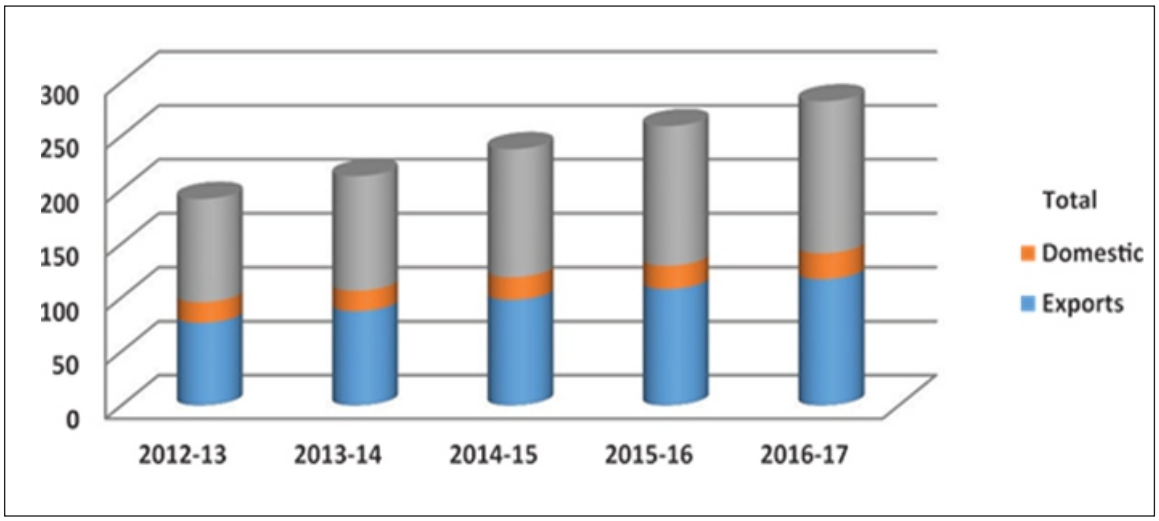


Fig.1.1.4 Revenue Growth of Indian ITeS sector over the last 5 years
(Source: NASSCOM) (All amounts in USD billions)

1.1.4 Search on the Internet About IT-ITeS/Test Engineers Industry

1. Android/Tablet

- On the Android phone or tablet, open the Chrome app Chrome.
- In the address bar, type IT-ITeS/Test Engineers industry and search.
- Tap the result, Go, or Continue Continue.

Tip: As one types, one may get suggestions based on the web and app activity. Users can delete individual suggestions from the search history or hide sections of suggestions based on the activity when they appear.

2. Computer

- On the computer, open Chrome application.
- In the address bar, enter **IT-ITeS/BPM** industry search.
- Select a result or press Enter.

Tip: As one types, one may get suggestions based on the web and app activity. Users can delete individual suggestions from the search history or hide sections of suggestions based on the activity when they appear.

Activitiy

Con-Vid Session

- The trainer will play a video during this session.
- The video will be about an Overview of the IT Industry.
- The YouTube link for the video is www.youtube.com/watch?v=Gk9dcfR7Oec&t=54s (Video Courtesy: Scrooch).
- The trainees will observe the video with pin-drop silence.
- They can note down pointers from the video that they may find relevant.
- Trainees will maintain decorum in the class and not talk, whisper or discuss in the class.
- In case of any queries or confusion, trainees will write those down in their notebooks.

Discussion Session

- The trainer will inquire whether the trainees comprehended the chapter.In this activity, the trainer will ask the trainee some questions related to the topic.
- The trainees will answer the questions according to their understanding of the topic.
- If the students have any questions or concerns, they can present them to the instructor.
- The trainer will ensure that he or she can give appropriate answers to the queries placed by the trainees.

Notes

UNIT 1.2: Career Opportunities for Test Engineers and Job Responsibilities

Unit Objectives

By the end of this unit, participants will be able to:

1. Discuss the role and responsibilities of a Test Engineer.
2. Describe the qualities a Test Engineer should possess.
3. Determine the career path for a Test Engineer.
4. Categorize key applications for Testing and Quality Assurance services.

1.2.1 Introduction

Any software product's or application's quality has a significant impact on its success. Today, testing is thought to be the best way to guarantee the quality of any product. Quality testing can significantly lessen the cascading effects of project rework, which have the potential to increase budgets and push back the schedule. Businesses are under pressure to create complex applications in less time, which increases the need for testing. Testing is a type of investigation used to judge how well a software service or product works. Additionally, it is the process of determining whether a product is correct and how well it functions.

A method of comparison is used in the testing process to identify product flaws. The behavior and condition of a given product are compared against a set of standards, which may include specifications, contracts, and previous iterations of the product. Software testing is a gradual and iterative process to find inconsistencies, flaws, or errors. According to Myers, "Testing is the process of running a program with the intention of finding errors."

Software testing, according to IEEE 83a, is the process of manually or automatically exercising or evaluating a system or system component to ensure that it complies with predetermined requirements.

Why Software Testing

Humans make mistakes all the time!!

"Software testing is really necessary to highlight the flaws and mistakes that were made during the development phases." We humans are unable to recognize our errors in our own work. We should ask someone else to review our work because they might catch the errors we made. Similar to this, software developers may not notice inconsistencies in a program or application they have created that a department known as software test engineers can spot.

It saves time and money to use high-quality software. Software will be less likely to have bugs, which will free up time during the testing and maintenance phases. Lower maintenance costs and greater customer satisfaction are two benefits of greater reliability. Because a significant portion of software costs are allocated to maintenance, the overall cost of the project will likely be less than comparable initiatives. Testing is essential because software bugs can be expensive or even dangerous. History is replete with instances where software bugs have potentially led to loss of money and lives.

Here are two examples that show how important software quality is:

- Due to software issues with the airbag sensory detectors, Nissan recalled over 1 million vehicles off the road. As a result of this software malfunction, two accidents have been reported.
- On April 26, 1994, a China Airlines Airbus A300 crashed as a result of a software error, killing 264 unintentional victims.

1.2.2 Roles and Responsibilities of a Test Engineer

- **Requirement Analysis**. The first thing a tester should do is analyze the requirements the customer has provided. The tester's duties include comprehending the requirements and pertinent areas of responsibility, creating a query list, and sharing it with the test lead.
- **Test Effort Estimation.** The tester is expected to comprehend the specifics of the tasks to be carried out during the test planning meeting. Additionally, he or she should estimate how long it will take to complete the task effectively.
- **Test cases documentation.** The tester is expected to record test cases for the task that has been defined and to which modules have been assigned. Test cases or test scenarios are created in a specific format based on organization and development methodology.
- **Reporting and tracking Defect.** It's critical to report any bugs found while carrying out testing tasks. Again, each organization employs a unique set of templates and tools for defect reporting. The tester must be aware of how a particular tool functions and must provide a thorough defect report. The tester must keep track of any reported flaws and ensure that they are fixed, depending on how important they are.
- **Listing improvement areas.** The expectation of a tester is to make suggestions to enhance the product's usability because they are a new set of eyes on the product.
- **Reporting to Test Lead / QA Lead.** The tester is required to send the test lead a daily status report outlining the testing activities carried out and any pertinent developments. A daily status report is a tool that junior testers can use to inform the test lead about the work they have completed.
- **Reports to :** Most frequently, Test Lead (depending on the organizational structure)

1.2.3 Knowledge and Skills Required as a Test Engineer

The technical and non-technical abilities and competencies needed to become a software tester are listed below.

- **Non-Technical Skills:** To be a successful software quality tester, you must have the following abilities. Check your skill set against the checklist below to see if software testing is a possibility for you:
- **Analytical skills:** Sharp analytical skills are necessary for a good software tester. To better understand and develop test cases for a complex software system, analytical abilities will be necessary. If you're unsure whether you have strong analytical abilities, click on this link. If you can solve even just ONE problem, you have excellent analytical abilities.
- **Communication skill:** A good software tester should be capable of communicating both verbally and in writing. Software testers should create test artifacts like test cases/plans, test strategies, bug reports, etc. that are easy to read and comprehend. Dealing with developers will require a certain amount of tact and discretion (in the case of bugs or any other issue).
- **Time Management & Organization Skills:** Testing can occasionally be a challenging task, especially when new code is being released. A software tester needs to manage their workload effectively, be highly productive, and have excellent time management and organization skills.
- **Great Attitude:** You need a geat attitude in order to test software effectively. an eagerness to learn and suggest ways to improve processes, as well as a "test to break" mentality. Because technologies advance so quickly in the software industry, a good software tester should keep up with changing technologies by updating their technical software testing skills. Your actions must show a certain level of independence, showing that you take ownership of the task at hand and complete it without much direct supervision.
- **Passion:** One needs a significant amount of passion for their profession or job in order to excel at it. A software tester needs to be passionate about what they do. But if you've never tested software before, how can you tell if you're passionate about it? Simple if software testing does not capture one's interest, one should move on to something else.

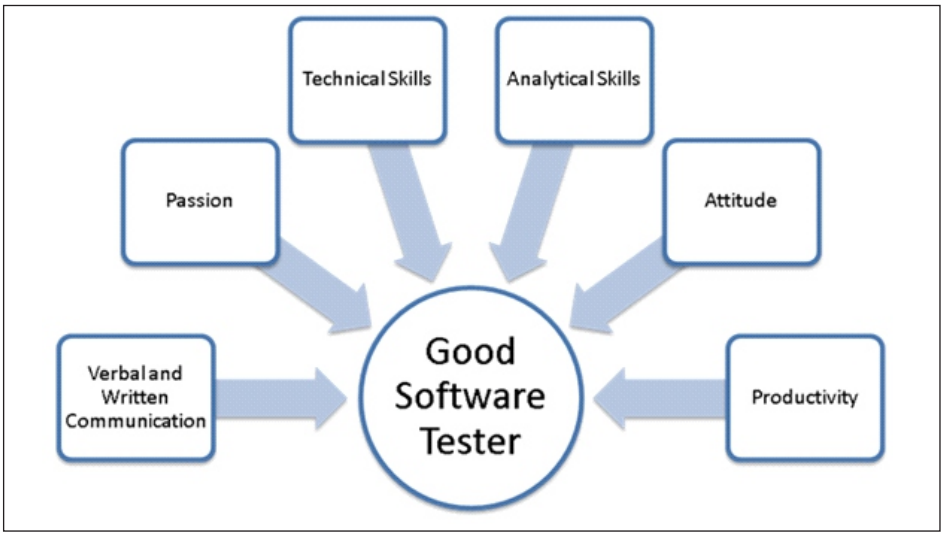


Fig. 1.2.1 Non-Technical skills of a Software Tester

Technical Skills

Technical tools have a lot of potential because non-functional testing requires that applications be tested for performance.

- **Basic knowledge of Database/ SQL:** In the background, software systems have a lot of data. On the backend, this information is kept in various databases, including Oracle, My SQL, etc. Therefore, there will be instances in which this data needs to be verified. If this is the case, simple or complex SQL queries can be utilised to determine whether the correct data is stored in the backend databases.
- **Basic knowledge of Linux commands:** Most software programs, including Web services, databases, and application servers, are installed on Linux-based computers. So it is essential that testers are familiar with Linux commands.
- **Knowledge and hands-on experience of a Test Management Tool:** An essential component of software testing is test management. The software testing process will not succeed without appropriate test management techniques. Managing your testing-related artifacts is the essence of test management.

For Example: Keeping track of all the test cases your team has developed can be done with the help of a tool like **Testlink.**

Additional tools are available that can be used for test management. As a result, it is crucial to have knowledge of and experience using these tools because they are utilized by the majority of businesses.

- **Defect Tracking tool knowledge and practical experience -**The defect life cycle and defect tracking are two essential elements of software testing. The proper management of defects and their systematic tracking are absolutely essential. The need for defect tracking arises from the fact that managers, developers, and testers should all be aware of any defects. Defects are logged using a variety of tools, including QC, Bugzilla, Jira, etc.
- **Knowledge and hands-on experience of Automation tool:** After a few years of manual testing, if you decide you want to become an "Automation tester," you need to become an expert in one tool and gain extensive, practical knowledge of automation tools.
- If you are searching for a job in automation, having **knowledge of any scripting language, such as V B Script, JavaScript, or C#, is always beneficial.** It's always advantageous for testers to be familiar with scripting languages like V B Script, JavaScript, or C# when looking for work in automation. Regardless of the fact that only a limited number of companies employ Shell/Perl scripting, there is a considerable need for testers with this skill. Once more, it will depend on the business and the tools that are employed by that business.

1.2.4 Career Map for a Test Engineer

One can advance from being a test engineer to a senior test engineer, test lead, and test manager; alternatively, one can become a QA lead, or QA Manager. There are many options available for testing tools. Anyone interested in a career in software testing should think about the following positions:

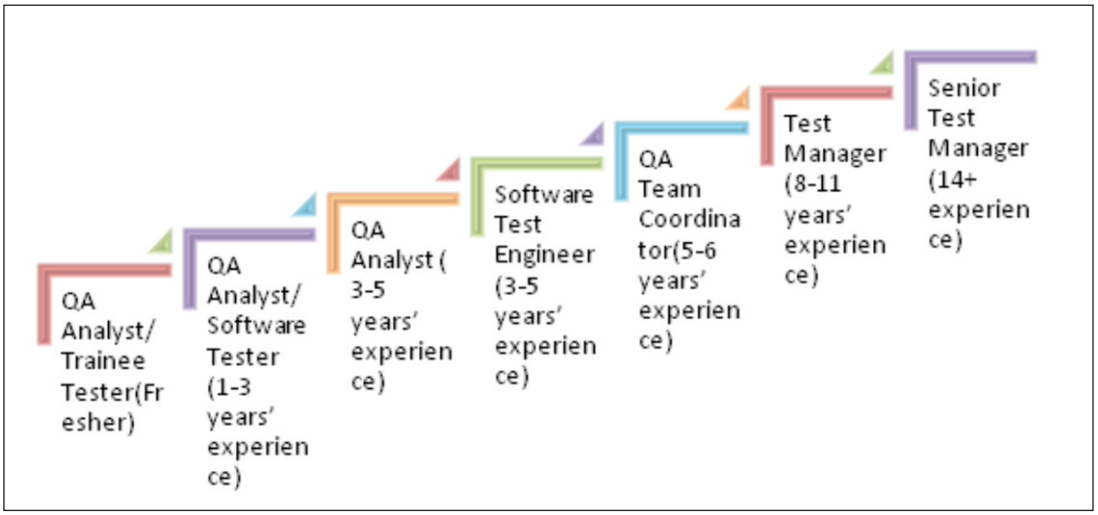


Fig. 1.2.2 Career Map for a Test Engineer

Alternate Career Tracks as a Test Engineer

Once you've had enough practice, you can pursue the following specializations.

- As an automation test engineer, it is your responsibility to automate the manual test case execution procedure because doing so would otherwise be time-consuming. Tools included IBM Rational Robot, Silk Performer, and QTP.
- Performance Testing: Among other things, your job as a performance test engineer will entail figuring out how quickly and how much load an application can support.
- One of the main advantages testers have over developers is their thorough understanding of business, according to a business analyst. Business analysis is a career that testers should consider pursuing. You, a business analyst, will analyze and assess the business model and workflows of your company. As a BA, you will integrate technology into these workflows and models.

As was already mentioned, one can advance from being a test engineer to a senior test engineer, test lead, and test manager; alternatively, one can become a QA lead, QA Manager. The profile of a software tester has drastically changed over the past ten years. For many application/product implementations, careers in software testing have become a deal-breaker, and business has realized how crucial it is to structure testing of applications prior to release. The testing community now has a variety of career options to choose from instead of just being a dull executor of test cases. The testing tool side has a vast array of options. In addition to test management tools like Quality Center from HP, CQTM from IBM, etc., there are numerous functional, performance, and security testing tools available.

There is a growing need for specialized knowledge such as that of security and SOA testers. There are a lack of scripting skills in test automation fields, including VB, Java, and other tool languages as well as scripting languages like Perl, Shell, Python, etc.

There is a demand for technical resources with the ability to assess automation tools, develop automation frameworks, and produce reusable components. Good performance testers who can analyze test results, locate bottlenecks, and recommend tuning methods are always in demand. The value of testing professionals is increased by having a strong understanding of the application's domain. Domains like BFSI, Telecom, Healthcare, Manufacturing, and embedded are continuously active.

1.2.5 Applications for Testing and Quality Assurance Services

Three generally accepted testing techniques are as follows:

1. **Black Box:** Regardless of how the inputs are converted into outputs, black-box testing involves validating the AUT against its specifications while taking into account the expected inputs and outputs. The internal organization or code that implements the application's business logic is least important to testers.

   The four main methods for creating test cases for black box testing are as follows:

   - B V A (Boundary Value Analysis)
   - EP (Equivalence Partitioning)
   - Decision Tables
   - State Transition Tables (and diagrams)

   Black box testing is commonly employed for functional, non-functional and regression testing

2. **White Box:** This methodology's main objective is to confirm how the application's business logic is implemented in the code or program.

3. **Grey Box:** In actuality, this is a hybrid of the black box and the white box. In this methodology, the tester primarily uses the Black-box approach to test the application. Testing is carried out using a white box, though, for some business-critical or vulnerable modules of an application.

SQA Techniques & Applications

Quality assurance (QA) testing is the procedure you use to make sure your product is the best it can be for your customers. Simply put, quality assurance (QA) refers to the methods used to avoid problems with your software product or service and to guarantee a positive user experience for your clients.

There are several techniques for SQA.

- The main method that is frequently used is auditing. There are, however, a few additional methods as well.

Various S Q A Techniques include:

- **Auditing:** Performing an audit entails looking over the work products and the information that goes with them to see if the established standard operating procedures were adhered to.
- **Reviewing:** A gathering where internal and external stakeholders review the software product and provide feedback and approval
- **Code Inspection:** In order to find bugs and stop defect growth in its later stages, the most formal type of review performs static testing. It is carried out by a trained mediator or peer and is based on guidelines, a check list, and entry and exit criteria. The author of the code shouldn't be the reviewer.
- **Design inspection is carried out using a checklist to look at the areas of software design listed below:**

- Interface and functional specifications
- Fundamental specifications and design
- Traceability of requirements
- Interfaces and structures
- Conventions
- Logic
- Performance
- Error handling and recovery
- Testability, extensibility
- Coupling and cohesion

- **Simulation**:A simulation is a tool that simulates a real-world scenario so that the behavior of the system under study can be virtually examined.
- **Functional Testing:** It is a quality assurance technique that checks what the system accomplishes without focusing on how it does it. The primary objective of this kind of black box testing is to evaluate the features or specifications of the system.
- **Standardization:** Standardization is essential to quality control. It ensures quality by reducing ambiguity and speculation.
- **Static Analysis:** This type of software analysis is carried out by an automated tool without the need to run the program. Software quality assurance for the nuclear, medical, and aviation industries frequently employs this method. Static analysis methods include, for example, software metrics and reverse engineering.
- **Walkthroughs:** A software walkthrough, also known as a code walkthrough, is a type of peer review in which the developer leads the team members as they examine the product and ask questions, offer solutions, and make comments about any potential flaws, standard violations, or other problems.
- **Path Testing:** With this white box testing method, every independent path is run at least once in order to guarantee complete branch coverage.
- **Stress Testing:** This kind of testing is done to determine a system's robustness by putting it through heavy load, or conditions that are outside of its typical range.
- **Six Sigma:** Producing nearly perfect goods and services is the aim of the Six Sigma quality assurance methodology. It is utilised extensively in many industries, including software development. Six sigma's primary purpose is process improvement in order to produce software that is 99.76% free of defects.

Some Essential QA Tools

- **Project Strategy** - Microsoft Word
- **Project/Sprint Planning** –Jira

- **Communication** - Slack, Microsoft Teams, Yammer
- **Document Repository –** SharePoint
- **Test Repository & Defect Management –** JIRA
- **Automation Tools** - Jenkins, Selenium, Cucumber, Java

1.2.6 Distinctions Between Software Quality Assurance and Software Testing

Following table explains on differences between Software Quality Assurance (SQA) and Software Testing:

Software Quality Assurance (SQA)	Software Testing
The engineering process that ensures quality is known as software quality assurance.	A product is tested for issues using software before it is released for use.
Involves actions related to the application of processes,	Involves actions related to product verification, such as Review Testing.
Process focused	Product focused
Preventive technique	Corrective technique
Proactive measure	Reactive measure
All future products produced by the organization were included in the scope of SQA.	The product that is being tested is covered by the scope of software testing.

Summary

- The Indian market for IT and business services is expected to reach $19.93 billion by 2025.
- As per the total foreign direct investment share of India, the rank of the Indian IT sector is #3 on a global level.
- As per the constant and consistent growth in the IT sector, the growth of the industry, by 2025, is anticipated to be the US $ 350 million.
- Testing involves running a program with the goal of identifying errors.
- A test engineer must run both automated and manual tests to make sure the software developed by developers is appropriate for its intended use and that any bugs or problems are fixed before it is released to general users.

- A Test Engineer may advance into different job roles as they gain knowledge and experience.
- Majorly of hard skills and soft skills are required to succeed in this Job Role.
- Three testing methodologies that are generally accepted include Black Box, which is frequently used for functional, non-functional, and regression testing. Grey Box, which is a combination of the black box and white box, is primarily concerned with validating how the business logic of the application is implemented by the code or program.
- Applications on which Test Engineer need to work are like Jenkins, Selenium, Cucumber, Java, Microsoft Word, Slack, Microsoft Teams, Yammer, Share Point, J I R A.

Activitiy

Con-Vid Session

- In th ndustry.
- The video will give a glimpse of the Job Role of a Test Engineer.
- The link for the You Tube video is: https://www.youtube.com/watch?v=3eOd9NTRgJo
- The trainees will observe the video with pin drop silence.
- They can note down pointers from the video that they may find relevant.
- Trainees will maintain decorum in the class and not talk, whisper or discuss in the class.

iIn case of any queries or confusions, trainees will write those down in their notebooks.

Discussion Session

- The trainer will inquire whether the trainees comprehended the chapter.
- In this activity, the trainer will ask the trainee some questions related to the topic.
- The trainees will answer the questions according to their understanding of the topic.
- In case, the trainees have queries and confusions in their minds; they can put forth those in front of the trainer.
- The trainer will ensure that he or she can give appropriate answers to the queries placed by the trainees.

Exercise

A. Fill in the Blanks

1. The IT industry sees the past 3 – 5 years as its ".

   a. Platinum Era b. Golden Period c. Diamond Bonanza

2. The aggregated revenues from the IT sector in 2017 were estimated to be around

   a. US$ 160 billion b. Rs. 5 Crores c. US$ 1 billion

3. Any unsolved problem needs to be escalated properly to the appropriate

   a. Software developer b. Internal teams c. Calculus

4. The employment growth in the IT industry is forecasted to increase by:

   a. 15-50% b. 7-8% c. 55-80%

5. Software testing is a process to _____ and _____ a discrepancy, a flaw, or an error.

6. Indicate whether each of the following claims is true or false:

   a. Software testing reveals the weak points in a program or product. ☐

   b. Myers made the observation that testing is the process of running a program with the aim of identifying errors. ☐

7. The primary objective of testing is to confirm the dependability of software systems by utilising them repeatedly in tightly regulated conditions. Justify.

_____
_____
_____
_____
_____

Notes

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

IT - ITeS SSC
NASSCOM

2. Concept and Principle of Quality Testing

Unit 2.1 - Concept and Principle of Quality Testing

SSC/N1301

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Discuss the objectives and scope of the quality assurance work being undertaken.
2. Discuss the key differences of quality testing vs quality control based on types of data.

UNIT 2.1: Concept and Principle of Quality Testing

Unit Objectives

By the end of this unit, participants will be able to:

1. Discuss various aspects of quality assurance pertaining to controls, job management, adequate process, performance, and integrity criteria.
2. Discuss the key requirements for quality testing, including policies, standards, processes, procedures, and version control.
3. Discuss the principles of subjective and objective data gathered from various sources.
4. Demonstrate how to categorize subjective and objective data based on rejection principle.
5. Perform quality control through objective and subjective data analysis.
6. Analyse the impact of soft elements, such as personnel integrity, confidence, organizational culture, motivation, team spirit and quality relationships on quality process.

2.1.1 What is QA Testing?

Quality assurance (QA) testing is the procedure used to make sure that your product is of the highest quality for your customers. Simply put, quality assurance (QA) refers to the methods used to avoid problems with your software product or service and to guarantee a positive user experience for your clients.

History of Quality Assurance

The concept of quality control is not new. A clear example of this can be found in the Middle Ages, when guilds established requirements for membership in order to uphold the standard of the goods and services that their members offered.

Royalty was also given the responsibility of quality control, such as maintaining their warships. They also contributed to the standardization of navies' training and equipment to ensure the safety of lengthy sea voyages.

Quality control also underwent a revolution during the Industrial Revolution. Workers were no longer solely accountable for the caliber of their own work. Foremen and management were given the responsibility of supervising this work, and systems and groups with expertise in quality assurance were formed.

The idea of software quality was introduced to ensure that the released software is secure and performs as intended. The phrase "degree of conformance to explicit or implicit requirements and expectations" is frequently used to describe it. These so-called explicit and implicit expectations correspond to the two basic levels of software quality:

- **Functional –** the item's adherence to design guidelines and functional (explicit) requirements. This aspect of software focuses on how it is actually used by the user, including its performance, usability, and lack of bugs.
- **Non-Functional –** the item's adherence to design guidelines and functional (explicit) requirements. This area of software focuses on the user's actual interaction with it, including its functionality, usability, and bug-free status.

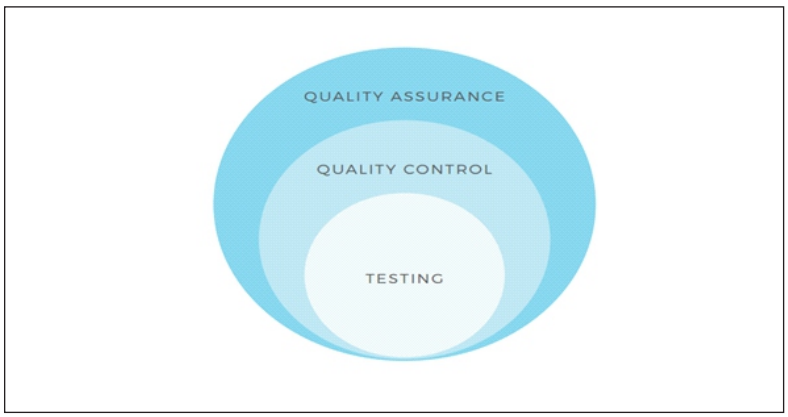


Fig. 2.1.1 QA, QC and Tsting in Software development process

Source:https://content.altexsoft.com/media/2016/10/qaqc-and-testing.png)

2.1.2 What is Version Control? How It is Important in Quality Assurance?

A version control system offers features for storing a series of iterations of software and related information files that are being developed. In terms of version control tools, a system release is a grouping of the related files. These files might include documentation, source code, compiled code, a version of the tool used to create the software, as well as information about the environment. Version control's main goal is to maintain a well-defined state for a software system at all times by ensuring a structured and traceable software development process in which all changes are carefully managed. The repository is a main location that houses the master copy of all the files for the majority of version control tools.

Any business may encounter issues with data backup, employee collaboration, and storing multiple iterations of files. For a company to succeed, it is imperative that all of these issues be solved. A version control system is then necessary in this situation.

These are a few of the major difficulties these companies face:

- **Collaboration**. Because there are so many people spread out in different locations, there might be a need to communicate for a specific reason, or a group of people from different regions may be working on the same project.
- **Storing Versions.** The project has been completed in multiple versions, making it difficult to keep all of these commits in one location.

- **Restoring Previous Versions.** Finding the source of a bug occasionally requires going back to an earlier version.
- **Figure Out What Happened.** Knowing what changes were made to earlier versions of the source code or exactly where changes were made in a file are crucial.
- **Backup.** All of the work is useless if the user's disk or system malfunctions without a backup.

Solution. Version Control to the rescue!

Multinational corporations can collaborate among their employees and help with remote repository backups thanks to the version control system. The system will enable developers to revert to earlier commits and the most recent version of the source code when necessary.

A single server repository may be used by a sizable number of workstations. Every workstation will have a copy of the current working code, and each of these workstations will save its source code to a specific server repository.

Any developer can now easily access the work being done with the repository thanks to this. The work will continue because there will be a copy of the source code in the central repository, even if ondeveloper's system fails.e particular

2.1.3 Benefits of Version Control

- **Managing and Protecting the Source Code.** The Version Control System assists the software team in managing the source code by keeping track of all code modifications. Additionally, it guards against unintended human error and its effects on the source code.
- **Keeping Track of All Code Modifications.** The team responsible for the project consistently creates new source codes and makes changes to the current code. These modifications are documented for future use and can be consulted if necessary to identify the underlying cause of a specific issue.
- **Comparing Earlier Versions of the Code**. Developers can go back at any time and compare earlier versions of the code to help fix the error while minimizing disruption for all team members because all the versions of the code are saved.
- **Supports the developers' work flow** rather than employing any rigid working methods. Any reliable version control system won't impose a particular way of doing things.. When using version control systems, which are renowned to deliver a seamless and continuous flow of code changes, it is common for developers to feel upset with this difficult mechanism.

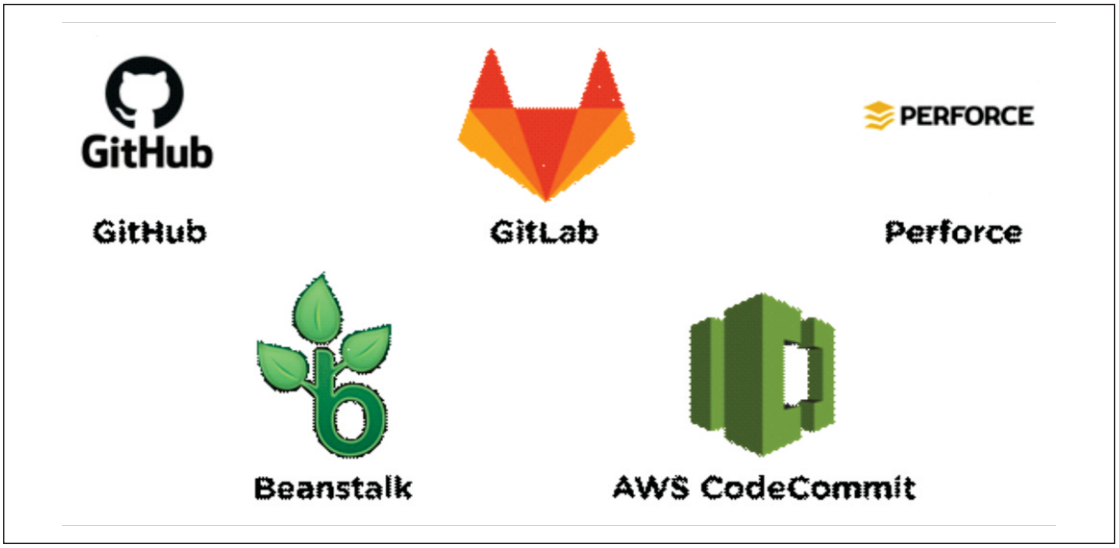


Fig. 2.1.2 Examples of Some of the Version Control Systems

2.1.4 Test Entry & Exit Criteria

Software testing is an essential phase of the software development life cycle since it demands a significant amount of time and effort from testers to evaluate the effectiveness and quality of software products. Although extremely helpful, this process frequently grows tiresome due to the numerous occasions it must be carried out across various platforms. Additionally, there are a variety of requirements that must be taken into account and tested, which can occasionally cause confusion for testers, particularly when deciding where to start and stop testing. To prevent this misunderstanding, the QA team establishes specific conditions and requirements prior to the start of testing that assist testers throughout the testing life cycle. These requirements are known as entry and exit criteria, and they are essential to the life cycle of software testing.

Entry Criteria

The accuracy of the testing process is guaranteed by entry criteria, which are finalized and selected after a thorough analysis of the software and business requirements. Ignoring them can reduce the process's quality.

The following are a few of the entry requirements, which are typically used to signal the start of the testing:

- There is available fully or partially testable code.
- There are defined and approved requirements.
- availability of the desired and necessary test data.
- Test cases are prepared and developed.
- The test environment has been set up, and all other resources, including equipment and tools, are available.

Exit Criteria

Exit criteria is a crucial document created by the QA team to follow the established deadlines and budget. This document outlines the conditions and demands that must be met prior to the conclusion of the software testing process. The teams of testers are able to wrap up the testing without sacrificing the quality and efficacy of the software with the help of exit criteria.

Exit criteria can be defined for each test level, from test planning and specification through execution, and heavily depends on the by-product of the software testing phase, such as the test plan, test strategy, test cases, test logs, etc.

The following are the commonly accepted exit criteria for stopping or wrapping up the testing process:

- Budget depletion or deadline adherence.
- execution and updating of every test case.
- coverage of the test's requirements and functionalities that is both desired and adequate.
- All found flaws are fixed and closed.
- No critical, high priority, or severe bug has been overlooked.
- Maintaining defects with their current status.

The following are a few of the leading best practices for entry and exit criteria considerations in the market:

- It is important to clearly define the entry and exit criteria for each test type before starting any process.
- Measures or conditions that are represented quantitatively rather than qualitatively
- If the entry or exit criteria are not met, corrective action must be allocated, or the entire process must be restarted with the necessary changes.
- During the creation and review of the process, the moderator must maintain constant watch and provide follow-up.

2.1.5 Data Collection Methods

Getting information is important in a time when "knowledge is power." But the question is, which data collection technique is best for your specific requirements? As such, raw data need not always be particularly beneficial. It's just a collection of unrelated facts and figures without a proper framework and context. However, if you organize, categorize, and analyze that data, you'll have a strong "fuel" for decision-making at your disposal.

Although the term "data collection" may sound very high-tech and digital, this doesn't mean that big data, the internet, and computers are always involved. Data gathering could take the form of a phone survey, a mail-in comment card, or even a man with a clipboard asking random bystanders some questions. But let's try to group the various data collection techniques into something resembling coherent categories.

There are two ways of gathering data. Aside: Many terms, including techniques, methods, and types, can mean different things depending on who uses them. For example, one source might refer to data collection methods as "methods." Regardless of the labels we choose, the fundamental ideas and breakdowns remain the same whether we're discussing a scientific research project or marketing analysis.

The two methods are:

1. **Primary**

   This is authentic, first-hand data that the data researchers have collected, as the name suggests. Prior to conducting any additional or connected research, this procedure is the first step in gathering information. Results from primary data are very accurate as long as the researcher gathers the data. However, there is a drawback because in-person research may be costly and time-consuming.

2. **Secondary**

   Secondary data is information that has already undergone statistical analysis and has been obtained from other sources. This information is either information the researcher has looked up or information the researcher has asked others to gather. It's second-hand information, to put it simply. Although secondary information is more accessible and less expensive than primary information, its authenticity and accuracy are questioned. The majority of secondary data is composed of numerical information.

   A taxonomy of the data collection methods is shown in Table 2.1. Each technique is categorized based on how much contact with people it necessitates. Direct contact with an audience is necessary for first degree contact. Second-degree contact necessitates access to participants' working environments, but it does not call for either direct access to participants or interaction between participants and researchers. Third-degree contact, on the other hand, only needs access to work artifacts, like source code or documentation.

Category	Technique
First Degree (direct involvement of software engineers)	**Inquisitive techniques** • Brainstorming and Focus Groups • Interviews & Questionnaires • Conceptual Modeling
	Observational techniques • Work Diaries • Think-aloud Protocols • Shadowing and Observation Synchronized Shadowing • Participant Observation (Joining the Team)
Second Degree (indirect involvement of software engineers)	**Instrumenting Systems** • Fly on the Wall (Participants Taping Their Work) • Analysis of Electronic Databases of Work Performed • Analysis of Tool Use Logs • Documentation Analysis • Static and Dynamic Analysis of a System

Table 2.1.1 Data collection techniques

2.1.6 Understanding Objective and Subjective Analysis

Knowing the definitions will help you distinguish between objective and subjective terms:

- Objective (adj) - not influenced by personal feelings, tastes, or opinions
- Subjective (adj) - influenced by subjective thoughts, feelings, or opinions Knowing the definitions will make it easier to distinguish between objective and subjective concepts.

As a result of these definitions, we can say that design is primarily an objective process. Early on in a project, it is advised that we follow an objective process rather than letting our feelings, "taste," theories, and unfounded assumptions influence us.

The goal of user research is to comprehend user needs, behaviors, and motivations through task analysis, observational methods, and other methods of feedback. According to Mike Kuniaysky, it is "the process of understanding the impact of design on an audience." User research is meant to assist us in designing unbiasedly with the end user in mind. Research keeps us from designing for ourselves and incorporating our personal preferences into the design. Instead, user research identifies the person, the environment in which they will use the product or service, and the assistance they require from us

2.1.7 Stress Testing

Stress: Stress testing is a procedure where software is examined to determine its ability to maintain consistent or acceptable performance under challenging circumstances. These might include high network traffic, a high process load, under- or overclocking of the underlying hardware, and operating under strict requirements for peripheral or system resource utilization. Stress testing aids in determining the degree of robustness and dependability, even when the system's normal operating limits are exceeded. Software that functions in high-stakes or real-time situations is thought to benefit greatly from stress testing.

Take a look at a browser window. To switch between multiple pages at once, users can open multiple browser windows. However, because of their interdependence, if one of these windows crashes, they all crash. In this scenario, the browser is put through stress testing.

How are Stress Tests Performed?

There are five main steps to the stress testing process:

- **The Stress Test preparation:** Here, you gather system data, analyze the system, and specify the objectives of the stress test.
- **Automated Stress Testing Script Creation:** During this stage, you produce test data for the stress scenarios and create the automation scripts for the stress tests.
- **Script Execution:** In this phase, you run the automation scripts for stress testing and store the results.
- **Results Analysis:** At this point, you examine the Stress Test results to find any bottlenecks.
- **Tweaking and Optimization:** In this phase, you adjust the system's settings and improve the code in an effort to achieve the desired benchmark.

To check that the adjustments have led to the desired results, you run the entire cycle once more. To achieve the performance goals, the Stress Testing process, for instance, typically requires three to four cycles.

Tools recommended for Stress Testing:

- **Load Runner :**A popular load testing tool is Load Runner from HP. Bench marking is done using load test results created by load runner.
- **J meter:** An open source testing tool is J meter. It is a stress and performance testing Java application only. J meter is designed to handle various test types, including load, functional, stress, etc. J D K 5 or later is required for it to work.
- **Stress Tester:** This tool provides an in-depth investigation of the performance of the web application and graphically displays the results. It is also quite user-friendly. It provides an excellent return on investment and requires no sophisticated coding.
- **Neo load:** This well-liked tool is readily available on the market to test mobile and web applications. To assess the performance of an application under load and to examine response times, this tool can simulate thousands of users. Additionally, it supports performance, load, and stress testing that is cloud-integrated. It is simple to use, economical, and offers good scalability.

- **Stress Testing Metrics:** Metrics are usually examined at the conclusion of a Stress Test and aid in assessing a system's performance. Metrics that are frequently used are-

Scalability and performance measurements

- The number of pages requested each second is measured by Pages per Second.
- Basic Metric for Throughput: Response Data Size/Second
- Rounds: The anticipated number of test scenarios versus the actual number of client executions pages per second measures the number of pages requested every second.

Application Response

- Hit Time An image or page usually takes a certain amount of time to load.
- **First byte time:** The amount of time required to return the first byte of data or information
- **Page Time:** The time required to retrieve every piece of information on a page.

Failures

- **Failed Connections:** The amount of unsuccessful connections that the client has rejected (Weak Signal)
- **Rounds Failed:** The total number of rounds it fails
- **Failed Hits:** The total number of system failures (e.g., broken links or hidden images).

The purpose of stress testing is to examine the system under difficult circumstances. It keeps an eye on system resources like memory, processor, network, etc. and evaluates the system's capacity to return to normal operation. It examines whether the system issues suitable error messages when under pressure.

2.1.8 Software Testing Principles

Software testing is a process that involves putting software or an application to use in order to find bugs or flaws. Following certain guidelines will help us test software or applications without creating any defects, and it will also save the test engineers' time and effort as they put their time and effort into doing so. We will learn about the seven fundamental tenets of software testing in this section.

Below are the seven different testing principles:

Fig.2.1.3 Software Testing Principles

1. **Testing shows the presence of defects**

   The application will be put through testing by the test engineer to ensure that there are no bugs or flaws. Only the presence of errors in the application or software can be determined during testing. The main goal of testing is to find any flaws that might prevent the product from fulfilling the client's needs by using a variety of methods and testing techniques. Since the entire test should be able to be traced back to the customer requirement.

   Testing reduces the number of bugs in any application, but this does not imply that the application is defect-free because occasionally software appears to be bug-free despite extensive testing. However, if an end user runs into bugs that weren't discovered during testing, it's when the server is deployed into production.

2. **Exhaustive Testing is not possible**

   In the course of the actual testing process, it can appear to be very difficult to test all the modules and their features using effective and ineffective combinations of the input data.

   Therefore, since it requires endless determinations and the majority of the hard work is unsuccessful, extensive testing is preferred instead. As a result, we can complete this type of variations in accordance with the significance of the modules since performing such testing scenarios would violate the product timelines.

3. Early Testing

Here, early testing refers to the idea that all testing activities should begin in the early stages of the requirement analysis stage of the software development life cycle in order to identify the defects. Early bug detection allows us to address them, which may end up costing us significantly less than if they are found later in the testing process.

In order to conduct testing, we will need the requirement specification documents; as a result, if the requirements are incorrectly defined, it can be corrected now rather than later, possibly during the development phase.

4. Defect clustering

The defect clustering specified that we can identify the quantities of bugs that are correlated to a limited number of modules throughout the testing process. We have a number of explanations for this, including the possibility of complicated modules, complicated coding, and more.

These kinds of software or applications will adhere to the Pareto Principle, which states that roughly 80% of complexity is present in 20% of the modules. This allows us to locate the ambiguous modules, but it has limitations if the same tests are run frequently because they will not be able to spot any newly introduced flaws.

5. Pesticide paradox

This rule stated that if the same set of test cases were run repeatedly over a set period of time, the tests would not be able to discover any new bugs in the software or application. Reviewing all the test cases frequently is crucial to overcoming these pesticide paradoxes. Additionally, in order to implement multiple components of the application or software, new and different tests must be written, which aids in the discovery of additional bugs.

6. Testing is context-dependent

The testing is a context-dependent principle that states that there are many markets available, including e-commerce websites, business websites, and others. Every application has its own requirements, features, and functionality, so there is a clear way to test both commercial and e-commerce websites. We will use a variety of testing methods, as well as various approaches, techniques, and methods, to examine this kind of application. As a result, testing is dependent on the application context.

7. Absence of errors fallacy

Once the application has undergone thorough testing and no bugs have been found prior to release, we can say that it is 99.9% bug-free. However, there's a chance that testing the application with incorrect requirements, finding flaws, and fixing them within a predetermined timeframe wouldn't be successful because testing is done using the incorrect specification, which doesn't apply to the client's requirements. Finding and fixing bugs won't help if the application is ineffective and unable to meet the needs and requirements of the client, claims the absence of error fallacy.

2.1.9 Various Aspects of Quality Assurance

A product or service's quality can be described as how well it performs in comparison to expectations. When output doesn't meet these predetermined standards, corrective action is taken as part of the quality control process, which compares output to a standard. Therefore, quality control in terms of customers would be the ongoing process of making sure that products are created to satisfy and even exceed customer needs. Quality control concerns are among the main causes of not renewing a contract for contract work, especially for work given by government organizations.

With this strategy, three elements are highlighted:

- Controls, job management, clearly defined processes, performance and integrity standards, and record identification are some of the components.
- Knowledge, skills, experience, and qualifications that demonstrate competence
- Soft factors include teamwork, moral character, organizational culture, motivation, and good relationships.

2.1.10 Principles of Quality Assurance

There are two principles to quality assurance. To ensOne is "fit for purpose," which means the good or service serves the purpose for which it was created. The alternative is "right first time," in which errors are corrected right away. The alternative is "right first time," in which errors are corrected right away. The objective is to make the product or service operate correctly all the time by managing the project's variables while keeping the two aforementioned principles in mind.Thus, managing the quality of raw materials, assemblies, products, and components is a part of quality assurance, as are services related to production, management, production, and inspection processes.

2.1.11 Quality Assurance Approaches

- **Failure Testing:** Failure testing, which is also known as stress testing, is a technique for pushing a product to its absolute limits by raising vibration, temperature, humidity, etc. to reveal inherent weaknesses. The results are then used to improve the product in order to maintain a higher standard.
- **Statistical Control**: This kind of quality assurance is based on analyses of objective and subjective data, and it tracks quality data and charts it against a common cause variance.
- **Total Quality Management:** The quality of the final product depends on the constituents involved, some of which are sustainable and controllable and others which are not. The quality is not guaranteed if the specification does not correspond to the true quality requirements.
- **Models and Standards**: These basic requirements for proficiency are laid out in this international standard. In an accredited laboratory, there are exams to pass along with 15 management requirements and 10 technical requirements.
- **Company Quality:** This idea first surfaced in the 1980s and is centered on how all departments should approach quality under the direction of management to create a process for quality improvement. Infrastructure as well as controls, job management, process, performance, knowledge, skills, and experience are used to achieve this.

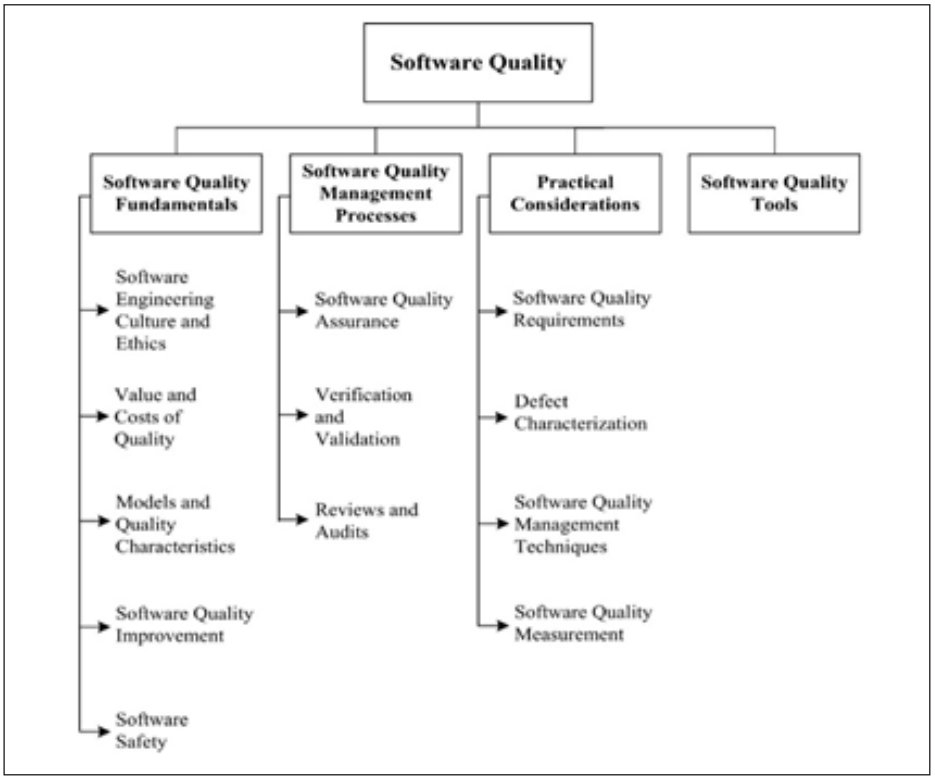


Fig.2.1.4 Software Quality

2.1.12 Quality Assurance in Practice Across Industries

Almost all industries use quality assurance, and using contractors or consultants to handle quality assurance is not uncommon.

- **The Medical Industry**

  Quality control is crucial in the medical sector because errors could result in the loss of human life. Nearly everywhere, from hospitals to laboratories, uses agencies to ensure that their standards are in line with the sector.

- **The Aerospace Industry**

  Quality assurance, also known as product assurance in the aerospace sector, collaborates with project management and engineering to prevent catastrophic failures with a human cost. The department responsible for product assurance reports to the highest levels of management and is funded independently.

- **The Software Industry**

  Quality assurance is a way to keep an eye on the software engineering processes during software development. This can be done in a various ways, including using ISO 9000 or a model like the Capability Maturity Model Integration (CMMI). Software is occasionally used to fix problems.

2.1.13 Software Tester Skill Set

An individual who tests software is often known as a software test engineer. He should be capable of creating test plans and understanding usability issues. This level of software tester is expected to be proficient in test execution and design approaches. A software tester must have excellent communication skills in order to work effectively with the development team. The following are the roles and duties of a usability software tester:

- Designing testing scenarios for usability testing is the responsibility of a software tester.
- He is in charge of running the tests, examining the results, and then reporting his findings to the development team.
- He might need to speak with the customers to better comprehend their needs for the product or if any changes to the design are necessary.
- Software testers frequently produce test-product documentation and take part in walkthroughs that are related to testing.

A software tester is responsible for a variety of tasks. His understanding of software testing should be extensive. He should be well-versed in the technical (GUI or non-GUI human interactions) as well as practical aspects of the system. The software tester must be knowledgeable about various testing methodologies and which strategy works best for a given system in order to develop test cases. He should be aware of the different software testing phases and how testing should be done during each one.

The responsibilities of the software tester include:

- Creation of test designs, test processes, test cases and test data.development of test plans, test procedures, test cases, and test information.
- Preparation of test cases, test information, test plans, and test procedures.
- Adhere to the guidelines as you carry out the testing process.
- Participate in walkthroughs of testing procedures.
- Make all necessary reports for the carried out software testing.
- Verify that all testing-related tasks are carried out in accordance with the accepted standards and guidelines.
- He must be skilled in coding, software development, and implementing test programs.
- Creation of source code
- Management of source code

2.1.14 Coding Skill for a Test Engineer

Even testers must have a strong understanding of coding. The following factors influence the response to this query:

- Black box testing doesn't require a tester to be knowledgeable in coding. Testers simply input data to the product, check the results, and repeat as necessary.
- Testers must have coding skills for white box testing. Because of this, this testing is also known as "code-driven testing." It may be useful to have knowledge of various concepts, such as C, C#, C++, DBMS, and RDBMS.
- Because they involve statement coverage, code coverage, cyclomatic complexity, etc., automation testing and white box testing require testers to have a solid understanding of coding. These ideas call for appropriate database and programming knowledge and abilities.
- Database verification may be necessary for testing. As a result, testers need to be familiar with basic SQL commands like select, create, update, etc.
- SQL injection is a method for breaking into databases by inserting unauthorized commands. One needs a solid understanding of SQL and JavaScript commands in order to defend the product against such security threats.lysts to determine whether any bugs can be identified earlier, before the testing phase begins. It hekps testing team actively participates in code review.
- The tester is in charge to ensure that the software is of high quality in agile testing. The tester collaborates on the coding throughout the process with various programmers. Thus, in order to create an automated test, testers must be familiar with coding.
- Reviewing Code. Code review is regarded as a component of static testing, an activity that is primarily carried out by quality analysts to determine whether any bugs can be identified earlier, before the testing phase begins. It hekps testing team actively participates in code review.

2.1.15 Quality Control

The objective of quality control is to ensure that particular procedures adhere to a company's established criteria.

This strategy lays an emphasis on three factors:

- **Elements** including controls, task management, well-defined and managed procedures, performance and integrity requirements, and record identification.
- **Competence, i**ncluding knowledge, abilities, experience, and credentials
- **Soft elements,** including people integrity, self-assurance, company culture, motivation, team spirit, and quality connections

Before a product is offered on the external market, it is inspected visually, frequently with the aid of a stereomicroscope to study the smallest of details. Inspectors will be presented with lists and explanations of prohibited product flaws, such as cracks and surface imperfections.

During World War II, there was a greater emphasis on quality control. At that time, quality control evolved into quality assurance, which is today referred to as a Strategic Approach, a method for enhancing not just goods but also processes and services. The quality of products and services should be evaluated based on a distinct set of dimensions. The responsibility for overall quality rests with the executive team. Management must set strategies, implement quality programmes, and motivate managers and employees. Typically, managers attempt to improve or maintain the quality of the entire organisation; this is known as Total Quality Management (TQM). TQM entails an organization-wide commitment to quality improvement on an ongoing basis. A company must involve the entire supply chain in order to reach and exceed its quality control goals.

Summary

- The process of keep a track that your product is of the highest possible quality for your customers is known as quality assurance (QA) testing.
- Functional and Non-Functional software quality are the two fundamental tiers.
- A set of conditions or goals that must be met in order to create an appropriate and favourable testing environment are known as entry criteria.
- The QA team created an important document called exit criteria to ensure adherence to the set deadlines and budget.
- User research is meant to assist us in designing unbiasedly with the end user in mind. Research keeps us from designing for ourselves and incorporating our personal preferences into the design.
- Stress testing involves evaluating the software's ability to maintain consistent or acceptable performance under challenging circumstances.
- Software testing is the process of putting software or an application to use in order to find errors or bugs.
- A software tester should be able to design test suites and comprehend usability problems.
- Database verification may be necessary for testing. Therefore, testers need to have a basic understanding of SQL commands like select, create, update, etc.
- Testers must have coding skills for white box testing. That is why this testing is also referred to as code-driven testing. It may be useful to have knowledge of various concepts, such as C, C#, C++, DBMS, and RDBMS.

Activitiy

Con-Vid Session https://www.youtube.com/watch?v=veG_WE2n-Lk

- In this session, the trainer will play a video.
- The video will give a glimpse of the Fundamentals of the Testing.
- The YouTube link for the video is: https://www.youtube.com/watch?v=veG_WE2n-Lk
- The trainees will observe the video with pin drop silence.
- They can note down pointers from the video that they may find relevant.
- Trainees will maintain decorum in the class and not talk, whisper or discuss in the class.
- In case of any queries or confusions, trainees will write those down in their notebooks.

Discussion Session

- The trainer will inquire whether the trainees comprehended the chapter.
- In this activity, the trainer will ask the trainee some questions related to the topic.
- The trainees will answer the questions according to their understanding of the topic.
- In case, the trainees have queries and confusions in their minds; they can put forth those in frontof the trainer.
- The trainer will ensure that he or she can give appropriate answers to the queries placed by thetrainees.

Exercise

1. Multiple Choice Questions:

I. What is pesticide paradox?

a) Testing can show that defects are present , but can not prove that there are no defects

b) If the same tests are repeated over and over again , eventually the same set of test cases will no longer find any new bugs

c) Finding and fixing defects does not help if the system built is unusable and does not fulfill the users need and expectations.

d) A small number of modules contain most of the defects

II. Which of the following testing is also known called by white-box testing?

a) Error guessing technique

b) Design based testing

c) Structural testing

d) None of the above

III. In Which ot the following Test Document is used to define the Exit Criteria of Testing?

a) Test Summary Report

b) Test Case

c) Test Plan

d) Defect Report

IV. ---------- are those software errors that occurred during the coding phase?

a) Failures

b) Errors

c) Defects

d) Bugs

2. List out theseven essential principles of software testing.

Notes

Scan the QR Code to watch the related videos

https://www.youtube.com/watch?v=LKUtzEHYK2I

QA Testing and Version Control

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

IT - ITeS SSC
NASSCOM

3. Design Tests for Software Products/ Applications/Modules

Unit 3.1 - Design Tests for Software Products/Applications/ Modules

SSC/N1301

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Demonstrate how to identify issues with the software requirements for testing using codes.
2. Examine the process to modify test cases relevant to the requirements.

UNIT 3.1: Design Tests for Software Products/Applications /Modules

Unit Objectives

By the end of this unit, participants will be able to:

1. Discuss the concept of source code and its use in application development.
2. Discuss the process of modification of the test plan, test cases and/or automated scripts.
3. Discuss the hierarchy of escalation in designing team for technical issues.
4. Discuss the process of Identifying issues with the requirements for testing in consultation with relevant stakeholders.
5. Demonstrate writing source code, reviewing code, etc. for software designing.
6. Modify test cases relevant to the requirements for automation.
7. Examine the steps to develop a test methodology to cover all the requirements.
8. Apply the design process for automated scripts relevant to the requirements.
9. Demonstrate how to develop source codes for various applications.

3.1.1 Source Code

The source code is a collection of instructions written by a programmer in computer programming languages. Once a programmer has written a line or set of source code, they can implement it in a website, application, or other sort of computer programme to instruct it how to operate. Someone might, for instance, write a line of source code that instructs a website to respond in a particular manner when a user hits a specific element, such as a button or link, on a web page.

Typically, programmers compose source code in a text-based application, such as a word processor, and then use a compiler to convert it into a format that computer programmes can understand. When source code is translated in this manner, it becomes object code. However, prior to translation, source code is typically comprehensible, as it explains the function a programmer wants to add to a website or programme. A programmer can store source code on a computer for future reference, either in a database or on its hard disc, following translation. Additionally, programmers can print physical copies of their source code.

3.1.2 Users of Source Code

Typically, the role that develops and uses source code the most is computer programmer, a technology expert that writes code for new websites and applications. Depending on the organisation they work for and the tasks they undertake, computer programmers are able to generate and implement source code in nearly any form of application or programme. They can also utilise a variety of tools, such as a visual programming tool, a text editor, or an integrated development environment, to write new source code (IDE).

Typically, computer programmers employ source code by writing a single piece of code for simple programmes, but they may write numerous sets of source code for a single project if the application is extensive. In such situations, it may be required to employ distinct programming languages that facilitate complex commands. The following are some of the most common computer programming languages used by professionals to write source code:

- C
- C++
- Java
- JavaScript
- Pascal
- Python
- Basic
- PHP

3.1.3 Common Source Code Applications

Here are some frequent applications of source code:

1. **Construct the basis of a website**

   One of the most common uses of source code is to establish a framework upon which web developers and web designers may build new websites. Source code is sometimes referred to as the foundation of a website because it contains the instructions that allow a website to run and respond to user interactions.

   The programming language used by programmers is a distinguishable part of constructing a website's foundation with source code. This is because HTML, a markup language, is often used to create a website's foundation, whereas most programmers use programming languages to write source code. HTML allows coders to define and modify features of a website, such as the primary content paragraphs, the headings, and any instances of highlighting or emphasis, such as bold or italic text, making it a highly useful tool for developing website foundations.

2. **Add a specific function into a Programme**

   Adding a specific function to a computer programme, application, or website is yet another typical use of source code. This is considered by many technology professionals to be the major purpose for source code, as it can significantly affect the functioning and presentation of programmes under development. A programmer working on a video game, for instance, can add an instruction to the source code that instructs the software to display a particular screen when the player's avatar reaches a specific point in the game.

3. **Communicate Algorithms**

   Additionally, programmers can share algorithms via source code. This is possible because the majority of programmers create source code in word processors or other text-based software, which allows programmers to print out hard copies of their code or store it as a document file. Then, they can share these files with other programmers who may be interested in their work or who wish to include a specific source code into one of their own projects. Some programmers utilise printed source code in items such as instruction manuals and reference books to teach others how to write code.

3.1.4 Compiler & Interpreter

In order for the computer to further process the programmer's source code, there must be a translation between the two, which takes the shape of an additional programme. This appeal for assistance may take the shape of a compiler or an interpreter.

Compiler: This type of programme converts (compiles) the source code into executable code that the process understands. This machine code is saved in an executable file format.

An interpreter translates the source code line by line and directly executes it. The translation process is significantly faster than that of a compiler, but execution is slower and memory requirements are high.

3.1.5 Structure of a Source Text

When writing a programme, you are restricted to particular standards depending on the programming language. However, very few programming languages arise from nothing, and the vast majority are built upon one another. Consequently, the following elements are utilised in several computer codes:

- **Commands:** Instructions are typically the foundation of all apps. Here, the programmer clarifies for the future programme what must be accomplished. Such commands can, for instance, initiate certain calculations or show text.
- **Variables:** Variables are empty slots that can be filled with information. These are repeatedly referred to with an assigned name within the source code.

- **Comparisons:** The majority of programme structures are based on questions that employ the if-then scheme, or propositional logic. If a certain truth value is entered, one solution is activated rather than another.
- **Loop:** Queries can also serve as the basis for loops in source code. A command is repeated until a specified threshold is reached. When the programme completes the loop and executes the remaining code.
- **Comments:** In all common programming languages, lines of code can be annotated with comments. This allows for the inclusion of text in the source code that is ignored by the application. For example, comments are added to the source code so that you or another developer can continue to comprehend portions of the code in the future.

Every time a source code is created, a problem must be resolved. Developers create software to deliver solutions. However, the method for doing so is not set in stone. Even though they are using the same programming language, it is possible for two programmers working on the same problem to produce source codes that are significantly different from one another.

Even while there isn't always a single correct solution, all programming tasks have one thing in common: a decent source code avoids needless text. The added text makes the problem more perplexing, slower, and error-prone. Since the structure of spaghetti code can be as twisted as a plate of dangling noodles, it is used to refer to particularly cryptic source code that even experts cannot decipher.

3.1.6 Create Code Source

To write source code, you need only a simple text editor, such as Notepad on Windows or TextEdit on the Mac. Thus, source code can be saved as plain text (e.g., in ASCII or UTF-8) with the appropriate file name extension for the programming language. Therefore, if you discover a file with the extension ".cpp" on your hard drive, it is likely a text file containing C++ programming code.

3.1.7 Software Testing

The process of developing software includes testing as a crucial step. It involves thoroughly evaluating software to make sure it satisfies the needs and objectives of your client. Prior to the implementation phase, testing's main objective is to find all flaws and errors in the software. Software flaws can harm the client's business if they are not fixed before deployment. Resolving those issues would involve high costs. Testing, on the other hand, enables one to preserve software quality and win your clients' confidence. The finished product will also perform precisely, consistently, and dependably, which will result in lower maintenance costs. Having said that, there are various approaches to software testing. The method that executes the testing process quickly and adheres to the Agile tenets is the best one. We will examine various software testing terminologies and test design in this unit.

3.1.8 Test Scenario

Software development is incomplete without software testing. The development of test cases is greatly aided by test scenarios. In-depth descriptions or records of how a user will interact with an application during software testing are called test scenarios. It is also known by the names Test Condition or Test Possibility. In order to show what the tester has done; test scenarios are used. Test cases are improved with the aid of test scenarios, which also help testers identify all potential outcomes—both expected and unexpected—so that bugs can be reported as soon as possible. We evaluate the performance of the applications from the viewpoint of the end user by using test scenarios. When building test scenarios, testers must put themselves in the perspective of the end user in order to comprehend the real-world scenarios the software will need to handle after release.

3.1.9 Designing Methods to Test High Level Scenarios

Instead of using test cases to test the software application, scenario testing is a method used in software testing.

Testing that uses scenarios created from use cases is known as scenario testing. Furthermore, by using scenario testing, difficult-to-evaluate test scenarios can be used to test complex application logic.

Some characteristics of scenario testing are:

- **Coherent:** The test scenarios ought to be based on a logical account of how the software is employed.
- They should be Credible and concentrate on a scenario that could occur in reality.
- **Motivating:** In the event of a failed test scenario, they should encourage the stakeholders to fix the problems.
- **Complexity:** Complex program or application flows are typically a part of the test scenarios.
- **Easy to evaluate:** Since the test scenario involves complex logic, the test result should be simple to assess.

3.1.10 Test Scenario Template

A Test Scenario document can have the below fields:

- **Module:** The application's module or component.
- **Requirement:** Id is an optional field that can be connected to the SRS.
- **Test ScenarioId:** The test scenarios are identified by this field.
- **Description:** The test scenario's purpose is described in the description field.

Template for a Test Scenario			
Module	**RequirementId**	**TestScenarioId**	**Test Scenario Description**
Login	US0001	TS_01	Check to see if the user can log in using valid information..
	US0002	TS_02	Make sure the user cannot log in using faulty information.
	US0003	TS_03	Verify the login page's mandatory field check validation.

Table 3.1.1. Test Scenario Template

Test Case for Gmail – Inbox Functionality

1. Check to see if a recently received email is highlighted and shown in the Inbox section.
2. Check to see if a recently received email has the sender's email address or name, the message's subject, and the message body (trimmed to one line) displayed correctly.
3. Make sure the user is directed to the email content when they click the newly received email.
4. Make sure the email's contents are displayed correctly with the desired source formatting.
5. Check that any email attachments are both attached and downloadable.
6. Before downloading, make sure the attachments have been virus-scanned.
7. Check to make sure none of the emails marked as read have been highlighted.
8. Check that the email list displayed in the inbox section includes a mail read time at the end of every email, both read and unread.
9. Check to see if the number of unread emails is shown next to the text "Inbox" in the left sidebar of Gmail.
10. Check that each time you receive a new email, the number of unread emails rises by one.
11. Check to see that when an email is read (marked as read), the number of unread emails drops by one.
12. Make sure all users can see the email addresses in the cc field.
13. Check to make sure that email recipients in the bcc are hidden from the user.
14. Check to see that all incoming emails are piled up in the "Inbox" section and are periodically deleted depending on the available space.
15. Check to see if email can be received from email addresses other than Gmail, such as Yahoo, Hotmail, etc.

3.1.11 What is Test Case?

A test case is described as a set of circumstances under which a tester determines whether or not a software application is operating in accordance with the needs of the customer. Preconditions, case name, input requirements, and anticipated outcome are all included in test case design. An action at the first level, test cases are derived from test scenarios.

A test case is described as a set of circumstances under which a tester determines whether or not a software application is operating in accordance with the needs of the customer. The design of a test case includes preconditions, the case name, input conditions, and the anticipated outcome. A test case is an action at the first level and is derived from test scenarios.

- The test case provides comprehensive information on the testing approach, testing procedure, preconditions, and anticipated results. These are performed during the testing phase to determine whether the software application is capable of executing the function for which it was created.
- By associating the defect with the test case ID, the test case aids the tester in defect reporting. If a developer missed something, it can be discovered during the execution of these full-proof test cases, so detailed test case documentation acts as a full proof guard for the testing team.
- The test cases must be written so that we don't miss any features for testing, which requires that we have the requirements from which to derive the inputs. Then, in order to maintain uniformity, we should have a template for test cases that all test engineers can use to create test documents.

3.1.12 Why We Write the Test Cases?

We write the test case for the following reasons:

- We will view the test case and start testing the application in order to demand consistency in the test case execution.
- To ensure better test coverage, we should cover all potential scenarios and document them so that we don't have to keep track of them all.
- Rather than a specific person, it depends on the process:An application was tested by a test engineer during the first, second, and third releases before he left the company. The test engineer thoroughly tested the application by deriving many values as he comprehended a module. It becomes challenging for the new person if the person misses the third release. As a result, all derived values are documented for future use.
- To prevent having to train each new test engineer on the product:The test engineer departs with a wealth of information and scenarios. These scenarios ought to be recorded so that the new test engineer can test using the examples provided and also create new scenarios.

The efficiency of the application is the main goal of writing a test case.

3.1.13 Test Case Template

An organized set of test cases for various test scenarios are contained in a document known as a test case template. These test cases are used to determine whether or not the program has the expected functionality. The test case template below can be used to check whether an application changes a user's username correctly.

Test Case 1 : Changing username

Description:

- User selects valid values when customizing game.

Pre-conditions for this test case

- User is logged in to application game
- User has active internet connection

User confirms new username

Test Case						
Steps	**Step Description**	**Data/Value**	**Expected Result**	**Actual Result** (if different from expected)	**Successful/ Failed**	**Log Number (if failed)**
1	User types in new username in text box	Bob	The characters the user is typing should appear in the text box		Successful	
2	User presses 'Confirm' button		A prompt informs the user their username has been successfully updated		Successful	
			Test Case Status	*Successful*		

User cancels changing username

Test Case						
Steps	**Step Description**	**Data/Value**	**Expected Result**	**Actual Result** (if different from expected)	**Successful/ Failed**	**Log Number (if failed)**
1	User presses 'Cancel' button		The application returns to Home screen with no change made to the username	The application does not respond	Failed	1
			Test Case Status	*Failed*		

Table. 3.1.2. Test Case Template

3.1.14 Test Case Design Techniques

Software testing techniques make it simpler to create better test cases. Due to the impossibility of exhaustive testing, manual testing techniques help reduce the number of test cases to run while increasing test coverage. They make it simpler to spot testing conditions that would otherwise be difficult to spot.

1. **Bsoundary Value Analysi**

   BVA is a technique that involves determining the bounds of the test data. Values such as the maximum, minimum, inside or outside of borders, and errors are all included. This strategy is predicated on the notion that developers are likely to make errors when working within predetermined constraints. In essence, rather than in the center of the input domain, many errors happen on the edges. According to its basic tenet, if a system performs well for these particular boundary values, it will also perform well for any value in between them.

In essence, rather than in the center of the input domain, many errors happen on the edges. According to its basic tenet, if a system performs well for these particular boundary values, it will also perform well for any value in between them.

For example: A text field allows input from 1 to 100 as a test case.

Boundary values analysis test cases - 0,1,2 and 99,100,101

2. Equivalence Class Partitioning

Equivalence class partitioning is yet another black-box testing strategy. This method divides the input domain into a number of classes or divisions. It is assumed that software will respond consistently to test data from the same classes in this scenario. In order to ensure that all other potential inputs in a class are covered, only one input per class should be tested. The creation of test cases for each class is the next step.

Test Scenerio	Description	Outcome
1	If the user enters characters between 0 and 6	They system will not accept it.
2	If the user enters characters between 7 and 12	The system will accept it.
3	If the user enters characters more than 12	The system will not accept it.

Table. 3.1.3. Equivalence Class Partitioning

For example: Consider a password field in a software application that accepts a minimum of 7 characters and a maximum of 12 characters.

After analyzing the above situation we can make 3 partitions: 0-6, 7-12, 13-17

1. Decision-Based Table Testing

An instrument for developing test cases is a decision table, also known as a cause-and-effect table. The system offered a tabular representation of the inputs and anticipated results so that users could see how the system reacts to various input combinations. When input is used to determine the software's conclusion, it provides a methodical way to create test conditions and enhances test coverage.

For example: The development of software that guarantees that only genuine persons get the covid-19 vaccination.

Rule - 'Only anyone over the age of 60, or anybody over 45 with a history of diabetes or hypertension, should be permitted to get vaccinated.'

S.No.	Age>=60	Age>=45	Diabeties	Hypertension	Expected Outcome
1	Y	-	-	-	Allowed
2	N	Y	Y	Y	Allowed
3	N	Y	Y	N	Allowed
4	N	Y	N	Y	Allowed
5	N	Y	N	N	Not Allowed
6	N	N	-	-	Not Allowed

Table. 3.1.4. Test Case Template

2. State Transition Testing

In this method, a finite number of states are presumptively present in the program. The Application Under Test (AUT) is in one state or another depending on how it responds to user actions. The tester can test the behavior of the AUT using this method, which enables them to enter action-based input conditions. The system behavior can be assessed using both positive and negative input values.

For example: Consider an ATM system function in which the account is locked if the user enters an incorrect password three times.

In this system, the user will be successfully signed in if the password is entered correctly in any of the first three attempts. If the password is entered incorrectly for the first or second time, the user will be required to input it again. The user's account is deleted if they enter an incorrect password three times.

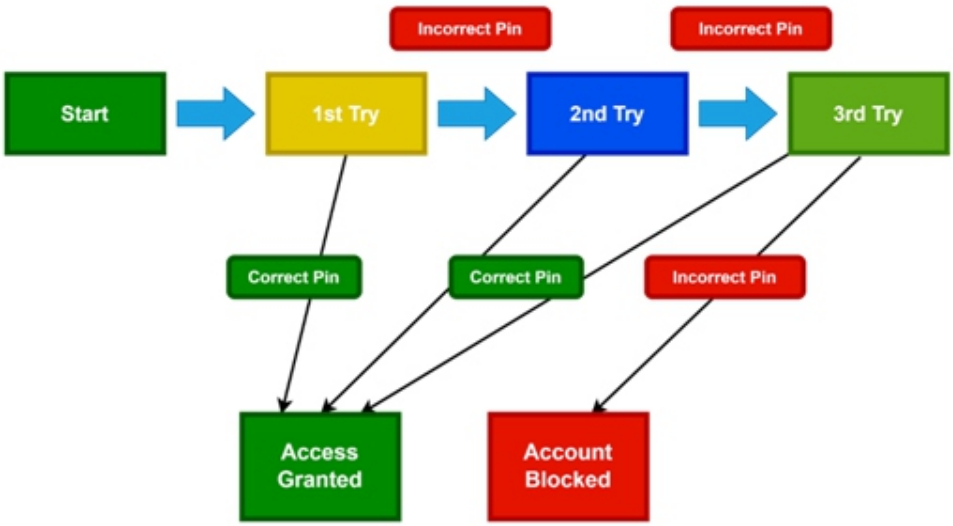


Fig. 3.1.1 State Transition Testing

In this diagram, whenever the user enters the correct PIN, he is moved to the Access granted state. He moves on to the next attempt if he enters the incorrect password.

In this diagram, whenever the user enters the correct PIN, he is moved to the Access granted state. He moves on to the next attempt if he enters the incorrect password.

Error Guessing Testing

This strategy is based on personal experience. Its success is primarily based on the test analyst's level of proficiency and comprehension of software testing, as well as its functionality and behavior. The test analyst's level of proficiency and comprehension of software testing, as well as its functionality and behavior, heavily influences the project's success. The tester foresees potential issues by drawing on their experience, understanding of edge cases, and knowledge of exception scenarios in the program flow. The method includes a list of potential errors or program sections that are prone to mistakes. The tester develops test cases to account for these situations. As a result, the tester may utilize this method to develop test cases based on previous experience.

For example: Imagine there is a software program and that program has a required field where the user must enter their mobile number and that field has some limitations.Constraints are that

1. The mobile should be numerical
2. It should be of 10 digits

Now, here comes the use of the error guessing technique

In this technique, the tester will analyze the corner cases such as:

1. What would be the result if the field is left blank?
2. What would be the result if less than 10 digits are entered?
3. What would be the result if any non-numeric character is entered?

3.1.15 A Test Script

Performing a set of instructions on a system under test to make sure it performs as expected is known as running a test script. For manual testing, they can be written in a human language, while automated testing requires them to be written in a scripting or programming language. A test case can have one test script or multiple test scripts, and a test script is a component of a test case. There are numerous test scripts attached to a single test case when:

- The scenario in the test case can be tested in various ways according to the test script. For instance, several scripts may be needed for a test case in order to test the scenario in various test environments.

There are both manual and automatic scripts to run the test case.

An illustration of a test script is provided below.

- For instance, your test script might run the following to test a website's login feature:
- Set the location where the automation tool should look for the "Username" and "Password" fields on the login screen. Let's assume that we will use their CSS element IDs.
- Click the "login" button on the website's home page. Verify that the "Username" and "Password" columns, as well as the Login screen, are all clearly visible.
- Next, enter the password "123456" and login "Charles," then look for and select the "Confirm" button.
- They must explain how a user can find the Welcome screen's title, for instance, by its CSS element ID, after logging in.
- Ensure that the title of the Welcome screen is visible.
- Read the title of the welcome screen.
- In the title text, type "Welcome Charles."
- The test was successful if the headline's wording matches what was anticipated. If not, the album fails the test.

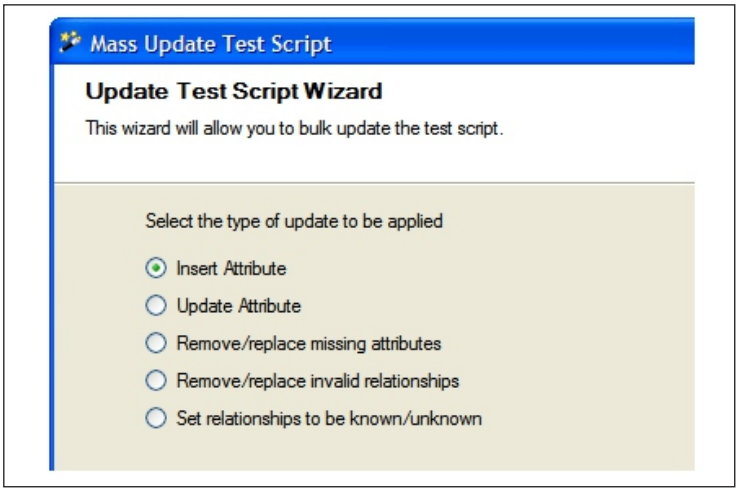
3.1.16 Modify a Test Script (Automated)

To account for changes to the rule base, test cases frequently need to be reviewed or changed. Individual test cases can be modified using the test case editor, and the Update Test Script Wizard can be used to make changes to multiple test scripts and test cases at once.

To modify several test scripts and test cases at once:

To modify several test scripts and test cases at once:

1. Right-click a test script or a folder that contains test scripts in Oracle Policy Modeling and choose Update Test Script Wizard. It displays the Mass Update Test Script dialog.



2. Select one of the four subsequent alternatives, which are detailed in greater detail below.

   - Insert Attribute
   - Update Attribute
   - Remove/replace missing attributes
   - Remove/replace invalid relationships
   - Set relationships to be known/unknown

1. Insert Attribute

This option allows you to insert a value for a test case attribute that has not yet been added. Typically, this occurs when a new attribute was introduced to the rulebase after the previous time the test cases were updated.

To insert an attribute:

- Select the **Insert Attribute** option on the first screen of the wizard and click Next.
- Select the test cases to which the attribute should be added. Use the browse button to select the attribute to be added, and enter the value which you wish to insert for the attribute, if any. Click Next.
- Review your changes on the **Summary of Changes** screen. Click Back to amend your changes if necessary, then click Next to apply the changes.
- After the wizard has applied the changes, select the Yes option to make another change, otherwise select the No option and click Finish.

2. Update Attribute

This option allows you to update the value for an attribute which already exists in your test cases.

To update the value for attribute:

- Select the Update Attribute option on the first screen of the wizard and click Next.
- Select the test cases to which the attribute should be added. Use the browse button to select the attribute to be added, and enter the new value which you wish to set for the attribute. Click Next.
- Review your changes on the Summary of Changes screen. Click Back to amend your changes if necessary, then click Next to apply the changes.
- After the wizard has applied the changes, select the Yes option to make another change, otherwise select the No option and click Finish.

3. Remove/Replace Missing Attributes

This option allows you to remove an attribute which still exists in your test cases, but has been removed from the rulebase. Alternatively, you can specify an attribute value which should replace it.

To remove or replace missing attributes:

- Select the **Remove/replace missing attributes** option on the first screen of the wizard and click Next.
- The wizard will detect whether any attributes exist in your test cases which are no longer present in the rulebase. Select the attribute you wish to change from the Attributes With Errors list. Leave the Remove Only checkbox selected if you just want to remove the attribute value from your test cases, or uncheck it and use the browse button to select an attribute to replace it with, and enter the value for the new attribute.
- Review your changes on the Summary of Changes screen. Click Back to amend your changes if necessary, then click Next to apply the changes.
- After the wizard has applied the changes, select the Yes option to make another change, otherwise select the No option and click Finish.

4. Remove/Replace Invalid Relationships

This option allows you to remove or replace any relationships in your test cases which no longer exist in the rulebase.

To remove or replace invalid relationships:

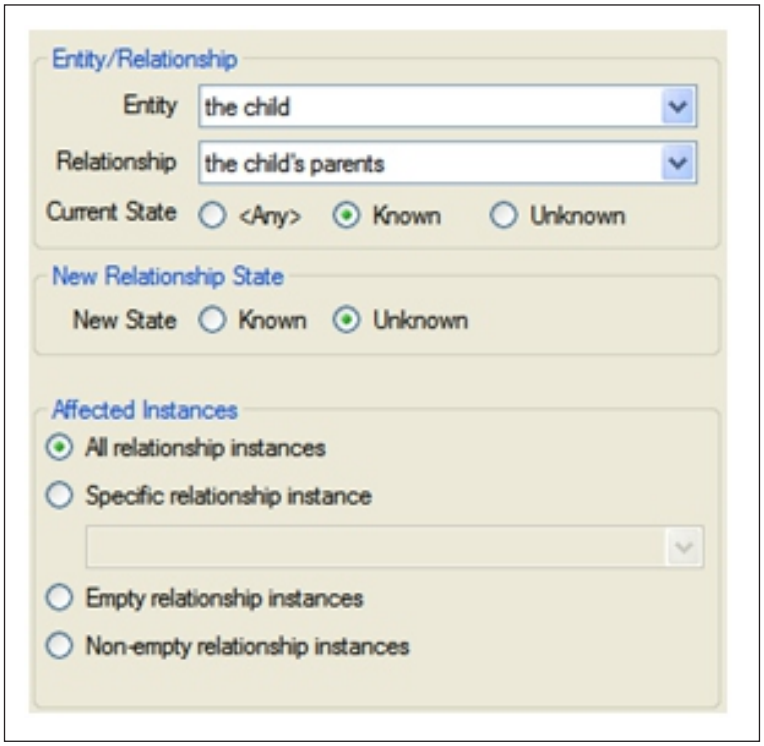
- Select the Remove/replace invalid relationships option on the first screen of the wizard and click Next.
- The wizard will detect whether any relationships exist in your test cases which are no longer present in the rulebase. For each Invalid relationship it detects, you can either remove it from the test case by selecting the Delete checkbox, or you can select a Valid relationship from the drop down list to replace it with. Once you have done this for each invalid relationship, then click Next.
- Review your changes on the Summary of Changes screen. Click Back to amend your changes if necessary, then click Next to apply the changes.
- After the wizard has applied the changes, select the Yes option to make another change, otherwise select the No option and click Finish.

5. Set Relationships to be Known/Unknown

This option allows you to set relationships to known or unknown.

To set the new state of a relationship:

- Select the Set relationships to be known/unknown option on the first screen of the wizard and click Next.
- In the left hand pane, select the test cases that the change is to apply to (or tick the Check all items checkbox of you want all test cases to be affected by the update).
- In the right hand pane, select the Entity, Relationship and Current State. Then select the New Relationship State and the Affected Instances.



4. Click Next.
5. Review your changes on the Summary of Changes screen. Click Back to amend your changes if necessary, then click **Next** to apply the changes.
6. After the wizard has applied the changes, select the Yes option to make another change, otherwise select the **No** option and click **Finish.**

3.1.17 Test Policy

A document referred to as "test policy" that provides organizational insight for the test activities is described at the level of the organization.

It is decided by the organization's senior management and establishes the standards that the company must follow. The test policy is a very important document that comes first in the hierarchy of test documentation.

In addition to publishing it in a separate document, some organizations prefer to publish their test policy in a sentence. They may also apply this policy to both construction and maintenance projects.

The test policy shall describe the followings:

- A specific response to the inquiry of "What does testing mean for the organization?"
- Test objectives that the organization have.
- A description of the organization's testing procedure for enhancing the caliber of developed software.
- How the organisation will assess the efficacy and efficiency of the test while meeting objectives
- The organization's plans to enhance its testing procedures.

3.1.18 Test Strategy

A test strategy document is created at the programme level and comprises general test strategy, management concepts, processes, and approaches for performing detailed tests on software. The test strategy document, which is developed at the programme level, includes the overall test strategy, management principles, methods, and methodologies for the specific tests to be performed on software.

In a top-level organisation, the test manager and project manager often write the test strategy document, which is also a high-level document. Large-scale projects typically prepare it, and it rarely needs to be updated. Test strategies and test approaches might be included in the test plan in small projects, and the test strategy document might not even be written separately.

The test strategy document's test approach and test procedures must be compliant with the organization's test policies:

- Objective / scope of testing
- In-scope / out of scope items for testing
- Test levels (Unit, System, Integration, System Integration)
- Test types (Functional / Non-Functional)
- Entry / Exit / Stop / Resumption Criteria for testing (for different levels / phases)
- Risks to be addressed
- Test environment
- Test case design methodology
- Test methodology (Top-down / bottom-up / risk based)
- Test control and reporting
- Test automation approach
- Test tools to be used
- Defect management approach
- Defect classification
- Retesting & regression approach

3.1.19 Testing Guideline

While testing the software, the following testing standards must be adhered to:

- **Software shouldn't ever be tested by the development team; instead, the testing team should be in charge of that.** Never let the software development team test it on their own. This is because, after investing many hours in creating the software, it might unintentionally grow too proprietorial, which might make it difficult to detect any flaws in the design.
- The testers ought to approach the product with a destructive mind set. Developers can perform unit and integration testing, but software testing should be handled by the testing team.**Testing will never be able to prove that a piece of software is 100% bug-free.** In other words, even after creating numerous test cases, it is impossible to demonstrate that the software is error-free.
- **Start as soon as possible: The requirement analysis process and the testing process should always begin concurrently.** In order to prevent the issue of defect migration, this is essential. The scope and test objects should be decided as soon as possible.
- **Prioritize sections: If there are any crucial sections, it is important to make sure they are tested as soon as possible and with the utmost urgency.**
- **There is a finite amount of time**: Software testing time is constrained. Before commencing the testing process, it is important to design a solid test plan and keep in mind that testing time is limited. To decide when to stop the testing process, there should be some criteria. It is necessary to choose this criterion in advance. For instance, in accordance with deadlines or financial restrictions, or when the system is left with an acceptable level of risk.
- **Testing must be done with unexpected and negative inputs:** To ensure that the system is leak-proof, testing should be conducted using both correct data and test cases and defective test cases. Evaluation must be c This demands the inclusion of test cases with precise definitions and descriptions of passed inputs and expected outputs. During testing, both accurate data and test cases and defective test cases must be utilised. Test cases must be well-documented to ensure their reusability in subsequent testing phases. This necessitates that the test cases be enumerated with definitions and descriptions of inputs passed and expected outputs. The software's functional and non-functional requirements should both be put to the test.
- **Correctly evaluating test results**: Tests and their results must be quantitatively evaluated. To ensure proper testing, the documentation should be properly cited when validating the test case results. As much as possible, testing must be supported by automated tools and methods. In addition to ensuring that the system performs its intended functions, testers must check that the system does not perform actions that are not intended.
- **Validating assumptions:** The test cases must always be adequately validated and should never be created based on assumptions. Cases should never be built on suppositions or presumptions. They must always be appropriately validated. For instance, designing test cases under the assumption that the software product is bug-free may provide very flimsy test cases.

3.1.20 Design Test Data

Test Cases use test data, which is production-like data, to evaluate software programmes. When test cases and test scripts are executed, test data is typically gathered into a document called the test data document.

Test Cases may not completely cover all possible scenarios if the test data is not planned in advance, which could ultimately compromise the software's quality.

Two categories of test data are recognised:

1. **Valid Test Data**: This refers to the positive data that the system uses to produce the anticipated outcomes. This is a synthesis of all potential inputs to ensure that the application is operating in accordance with the specifications.
2. **False Test Data:** False test data is the negative data that is utilised to test false or negative conditions and exceptions. Here are a few instances:
3. **Valid Test Data:** This refers to the positive data that the system uses to produce the anticipated outcomes. This is a synthesis of all potential inputs to ensure that the application is operating in accordance with the specifications.
4. **Unreliable Test Data:** Unexpected or unfavourable conditions and exceptions are tested using unfavourable test data. Here are a few instances:
    - Null Values – in case of mandatory fields
    - Out Of Range Values
    - Special characters that are not allowed
    - Invalid data format – for eg. mobile# with alphabet

A combination of genuine and false data that covers both positive and negative test scenarios is considered good test data.

Test Data-Wrting

Before testing starts and usually while designing test cases, sample data must be created. because configuring the test environment can involve a number of steps in many test environments while creating test data. Additionally, if test data creation is done during test execution, this could cause the test to run slowly.

The people that create the test data for software are the testers. Business Analysts may offer masked production data as test data in specific circumstances, such as with banking or medical applications, when the data is more sensitive.

Test data can be created:

1. **Testers can generate test data in one of two ways**: manually or automatically. Manually generating test data often involves creating it manually in an excel spreadsheet. To create test data, one can also use Word, Text, and XML files.
2. **Copy From Production**: When testing for Load, Stress, and Performance, a substantial amount of data is typically necessary. Utilizing Automated data production tools, a configurable utility that creates data, tables are generated (views, procedures, etc). This information is utilised during performance testing, database testing, load testing, functional testing, and usability testing.

Example:

Consider the below Test Application in order to comprehend Test Data.

Here are the requirements of User ID and Password:

User ID Requirements:

1. Minimum length is 3 characters, maximum is 15.
2. Only special character allowed is '_'
3. User ID must be Unique.

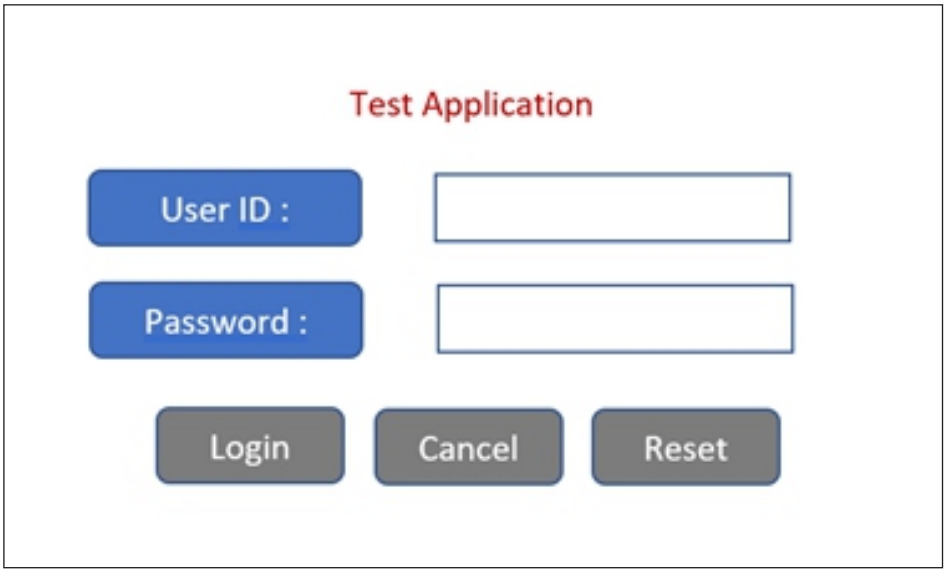


Fig. 3.1.2 Test Application

Password Requirements:

User ID	Password	Comments
John_12	Excellent@56$	Valid Input
Anna_Jose_Samul	passWORD#12345	Valid Higher Boundry Vlaues
Pet	Pst_98	Valid Lower Boundry Values
John_12	%XYZabc%	Invalid- Duplicate User ID
%^&*(uo	Password#123	Invalid- Special Characters in User ID
NULL	Pst_9890	Invalid- Blank User ID
Jippsy_45	tigress	Invalid- No Special charactors in password
Rose_merry9	68778#$	Invalid- No Alphabet

1. Minimum length is 5 characters, maximum is **15,**
2. At least **1 upper-case** Alphabet,
3. At least **1 lower-case** Alphabet,
4. Minimum **One Number,**
5. Only special characters that are allowed: **! # $ % * ? @ &**

So, below is the Test Data created for maximum test coverage of the above requirements

Test Data Template				**Test Data**		**Optional**
Module Name	Test Scenario ID	Test Case ID	Test Case Name	User ID	Password	Comments
Login Module	TS_001	TC_001	Verify User is able to login with CORRECT User Id and Password	John_12	Excellent@56$	Valid
				Anna_Jose_Samul	passWORD#12345	Higher Boundry Vlaues
				Pet	Pst_98	Lower Boundry Values
		TC_002	Verify User is NOT able to login with INCORRECT User Id and CORRECT Password	John_12	%XYZabc%	Duplicate User ID
				%^&*(uo	Password#123	Special Characters in User ID
				NULL	Pst_9890	Blank User ID
		TC_003	Verify User is NOT able to login with CORRECT User Id and INCORRECT Password	Jippsy_45	tigress	Invalid- No Special charactors in Password
				Rose_merry9	68778#$	Invalid- No Alphabet in Password

Table 3.1.5 Test Data Template

Results:

- Regular updates should be made to Test Data.
- Prior to the test case execution, when designing the test case, it should be created.
- To test negative scenarios, test data should contain invalid inputs.
- Use automation technologies to create vast amounts of test data for load or stress testing.
- Take advantage of the developer's / BA's assistance when creating test data.
- In conclusion, it is usually advisable to include test data that contains every possible combination of supported and unsupported formats. This guarantees complete test coverage.

Summary

- The primary goal of testing is to identify all the defects and errors in the software before the implementation phase.
- During software testing, test scenarios are detailed descriptions or logs of how a user will interact with an application. It is also referred to as Test Possibility and Test Condition..
- A test case is a set of situations under which a tester decides whether or not a software programme meets the customer's specifications.
- Throughout the development process, testing is performed. It is important to plan, design, and develop performance metrics to carry out testing.
- Test plans, test cases, and test reports that are well-written assist the tester in achieving the goal of accurately describing and documenting the test activity..
- The project briefing must include a comprehensive test strategy. Your testing document should be transparent, succinct, and adaptable to changes in your schedule or environment.

Activitiy

Con-Vid Session

- In this session, the trainer will play a video.
- The video will give a glimpse of the Test Plan and How to write?.
- The You Tube link for the video is: https:// www.youtube.com/ watch? v=S2_ AJP9Oeg0
- The trainees will observe the video with pin drop silence.
- They can note down pointers from the video that they may find relevant.
- Trainees will maintain decorum in the class and not talk, whisper or discuss in the class.

In case of any queries or confusion, trainees will write those down in their notebooks.

Discussion Session

- The trainer will inquire whether the trainees comprehended the chapter.
- In this activity, the trainer will ask the trainee some questions related to the topic.
- The trainees will answer the questions according to their understanding of the topic.
- In case, the trainees have queries and confusions in their minds; they can put forth those in front of the trainer.
- The trainer will ensure that he or she can give appropriate answers to the queries placed by the trainees.

Exercise

A. Multiple Choice Questions:

1. Which of the following is also known as white-box testing?

   a) Error guessing technique b) Structural testing

   c) Design based testing d) None of the above

2. What are the different levels of Testing?

   a) Integration testing b) Unit testing

   c) System testing d) All of the above

3. Which of the following is not included in the Test Plan?

   a) Incident reports b) Entry and exit criteria

   c) Schedule d) Risk

4. Which of the following Test Document is used to define the Exit Criteria of Testing?

   a) Test Summary Report b) Defect Report

   c) Test Plan d) Test Case

5. The test cases are signed off by whom?

   a) Team lead b) Developer

   c) Business Analyst d) Project manager

6. What are the inputs for Test Planning

   a) Req. Documents b) Test Strategy

   c) Both (a) and (b) d) None of the Above

7. _____ testing is used to check the code?

   a) Black box testing b) Grey box testing

   c) Red box testing d) White-box testing

B. Answer the Following Question in Brief:

a) What is the purpose of exit criteria?

b) What is test Case?

c) Define Test script

Notes

Scan the QR Code to watch the related videos

https://www.youtube.com/watch?v=RDBzZ-9XnDE

Design Tests for Software Products/Applications/ Modules

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

4. Carry Out Automated Tests on Software Products/ Applications /Modules

IT - ITeS SSC
NASSCOM

Unit 4.1 - Carry out Automated Tests on Software Products/ Applications/Modules

SSC/N1302

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Demonstrate the knowledge of identifying the nature of testing to be carried out and the test management tool to be used.
2. Examine the process of functional, usability, compatibility, performance, and regression testing on applications.

UNIT 4.1: Carry Out Automated Tests on Software Products/ Applications/Modules

Unit Objectives

By the end of this unit, participants will be able to:

1. Discuss the process for Identifying latest versions of the test cases and automated scripts.
2. Discuss how to identify the correct versions of application and data sources required for testing.
3. Describe key elements of functionality, usability, and regression method.
4. Discuss how programming languages like Java, SQL, etc. assists in development of software modules.
5. Interpret instructions to carry out automated test scripts.
6. Demonstrate the process of functional, usability, compatibility, performance, and regression testing.
7. Apply programming language, like C, C++, SQL, Java, etc. to develop software modules.

4.1.1 What are Test Management Tools and Why do You Need them?

The process of managing daily testing activities, including the development of test cases, execution of tests, and recording of test results, is known as test management.

Test case creation, test execution, test result archiving, report generation, defect tracking concurrent with test case execution, and import/export of tests from Excel to test management tools are all managed by test management tools.

4.1.2 Choosing the Right Test Management Tool

- **Pricing:** When choosing a test management tool, the license cost should be taken into consideration. A license's cost may be determined by the number of persistent, concurrent, or individual users. The costs of implementation and training should be considered in addition to the upfront license costs.
- **Integration Support:** Test Automation tools and different CI/CD tools should be well integrated with Test Management tools. The tool should have the capability to integrate with bug tracking tools like Bugzilla, Mantis, etc.
- **Adaptation to Different Development Processes:** Many teams have embraced the Agile and DevOps ways of working across the various organizations, so the tool should be able to map with either Agile, Waterfall, or a combination of both.
- **Supporting Multiple Users:** When multiple users access the tool for the management of test cases, one thing to look for that whether a particular test case is accessed by two users at the same time or not.

- **Easy to Use and Facilitate all Stages of STLC:** The test management tool should make all stages of the STLC easier to complete, including maintaining test cases, logging defects, connecting bugs to test case runs, test execution, and generating personalized reports.

4.1.3 Need For Test Management Tool

The following are some of the reasons why test management tools are required:

- **Test Assigning Features:** The test management tool tracks details about the tests and users can assign tasks to specific people. It also tracks details such as "who ran the tests?", "what tests were run?", "who passed the tests?" , "what requirements have been covered?"
- **Managing and Tracking Test Activities:** A test management tool is useful for managing and tracking the tests and defects found during the STLC phase.
- **Eliminate Complexity and Duplication:** To avoid the duplication of data and to manage the complexity of testing an application from an end-end-end perspective, a test management tool is a must-have for any organization.
- **Full Visibility and Traceability for Tests:** Every phase of the testing lifecycle can be traced within a test management tool.

4.1.4 Advantages of Test Management Tool

- Easy to Use and Simple UI.
- Allow access to multiple users with specific permissions to different projects.
- Full visibility and traceability
- Integration to Test Automation tool and CI/CD tools such as Jenkins, Bamboo, etc.
- Allowing the tool mapping to Agile and Waterfall methodology.
- Monitoring the project's overall progress and metrics
- One-stop shop for all phases of testing activities.

4.1.5 Management Tools that are Widely Used

1. **SpiraTest by Inflectra**

   The most effective end-to-end test management tool on the market today, SpiraTest enables users to manage test cases, requirements, and defects with graphical reporting in one location.

   Features:

   - Users can check the status of the tests and the overall project status using fully customizable reports.
   - All tests—manual and automated—can be easily created and run by users from a single location.

- Seamless reporting and tracking of defects.
- For better traceability, users can connect defects to test execution.
- It enables seamless integration with CI and CD tools, J I R A, and Azure Dev Ops tools, as well as automated testing tools like Selenium.
- Either a S aa S/ Cloud service or an on-premise solution is accessible.

2. Test Rail

Users can create test plans, test cases, and manage test execution all in one place using the web-based GUI interface called Test Rail. Control and monitor every step of the software testing process.

Features:

- Allows one to manage test suites, past test results, and track test case history.
- Record the test execution results in a web-based interface.
- Group, Sort and Filter the results effectively.
- Archiving and Auditing options for your tests.
- Tracks the tests on mobile devices, browsers, and multiple OS
- Easily Customizable reports & Metrics for test plans and test runs.
- Seamless integration with Jira Cloud, Automation testing tools such as Ranorex, BitBucket, Frogbugz, Mantis, Git Hub, and Bit Bucket.
- Teams to effectively collaborate on comments, feedback.
- Personalized to-do lists and email notifications for newly assigned tests.

3. X Ray – Test Management For J I r a

There is a Cloud App for Jira that uses Jira Issue for all testing activities that is available on the Atlassian Marketplace.

Features:

- While Installing Xray for Jira, you get to see new issues for Test Plans, Test sets, test runs, and test execution.
- Add on feature for requirement traceability which lets you track requirements, tests, defects, and executions.
- Seamless integration with Selenium, Cucumber, Junit, and Nunit.
- Integration of Rest API with CI/CD Pipelines.
- Advanced coverage for Requirements, Defects, tests, and execution.
- Customizable XRay reports tracking the test executions and Test sets with overall testing progress.
- Added feature to track the work of teams using Jira Agile boards for better visibility and transparency.

4. Zephyr Scale

Best Test management tool inside the entire testing life cycle will be managed by Jira. It is simple to use and self-explanatory.

Features:

- Allows us to configure according to the needs of a team. For example, you can add custom fields and status at the project level without the need for J ira Administrator.
- Set the permissions to allow certain users access to various functions within an application such as tests cases, test cycles, executions, and reports.
- Group your tests into subfolders and folders based on the requirements such as Automation Tests, Smoke tests, and performance tests.
- Reuse tests across different projects, modules, and test libraries to avoid performance issues.
- Managing multiple versions of test cases at the same time allows us to compare the changes between versions.
- J ira Issue view allows you to view test executions, test cases, results.
- Zephyr Scale reports can be filtered by epics, stories, and versions which allows you to view full test progress for a specific workspace or version.
- Scale your testing activities by integrating Cucumber BDD, CI/CD, and automation tools.

5. Zephyr Squad

To have a flexible and seamlessly integrated test management tool that functions as a native Jira application, Agile teams should use Zephyr Squad. One can create test cases, run tests, and view test execution reports using Zephyr Squad.

Features:

- Seamless Integration with Jira.
- Easy to use look and feel makes a better choice for Agile teams.
- View test executions and results by story view.
- Jira dashboard allows you to view test executions by test cycle and testers.
- Integrate with Automation testing tools such as Selenium and CI/CD tools such as Jenkins and Bamboo.

6. PractiTest

This tool for managing test cases from beginning to end is SaaS-based. Users can efficiently create tests, run them, keep track of bugs, and produce reports using this.

Features:

- It provides the ability to import and export issues, tests, steps, and requirements.
- Gives users the choice to perform Manual, Exploratory, and Automation testing without integrating with 3rd party tools.
- Create a custom field relevant to the project which could be used with different tests ad issues.
- It supports multiple browsers such as Chrome, Fire fox, IE, Edge, and Safari.
- Better visibility for Manual and Automation test results supported by offering multiple reporting options.
- Provides seamless integration with JIRA, Pivotal, Azure Dev Ops, Jenkins, Git Hub, Bugzilla, and Slack.
- Seamless integration with CI/CD tools such as Jenkins and Bamboo and their API gives the flexibility to add your own custom integrations as well.
- Integration with Fire Cracker tool to import XML test results to Practitest without using any API code.

7. Test Link

Users can manage test cases, test suites, test projects, and user management with the help of Test Link, a web-based open-source test management tool.

Features:

- Supports both manual and automated test execution.
- Multiple users can access the functionality of the tool with their credentials and assigned roles.
- Generation of test execution reports in various formats such as Word, Excel, and HTML formats.
- Easy Import/Export of test cases.
- Seamless integration with Bugzilla, Mantis.
- Linking of test cases with defects.
- Filter and sort test cases based on Test case ID, version.

8. Q test

A test management tool called Q Test was created by QA Symphony and is compatible with Agile development. The project requirements can be added, test cases can be made, tests can be run, and test results can be stored.

Features:

- Easy User Interface enables users to track the testing activities.
- Integrate of QTest with Bugzilla or JIRA.
- Easy Import/Export of test cases from Excel spreadsheet.
- Customizable reports to display the data useful to you using filter and sort options on date or field.
- Track changes to test cases and requirements.
- Reuse test cases and test suites across multiple releases.

9. QMetry Test Management

One of the top test management tools is QMetry. One of the top test management tools is QMetry. incorporated with Jira as well as CI/CD systems like Jenkins, bamboo, and automation frameworks. Tracking requirements, managing test cases, running tests, reporting, managing users, and managing issues are some of its key features.

Features include :

- Control of test execution and end-to-end testing.
- Integration of automation and CI/CD tools.
- Assists DevOps and Agile teams in raising product quality.

10. Kualitee

It is intended to coordinate all testing activities on a single platform. You can run test cycles, manage test case repositories, and record errors.

Features:

- Multiple people can work on the same item, and all feedback is shared in one location. Simple and easy-to-use user interface. One-stop shop for test case management and problem tracking.
- Makes it simple for users to make projects, modules, test cases, testing cycles, run test cases, record errors, and produce reports.
- Makes it simple for users to make projects, modules, test cases, testing cycles, run test cases, record errors, and produce reports.
- Simple import/export of test cases in different file types (Excel, Word, CSV).

4.1.6 Testing Life Cycle

- **Prerequisite:** Preconditions which must be attained before executing the test.
- **Planning:** Scope of the test, Environment under which test needs to be performed, Testing phase and Methodologies to be used, Manual and Automation Testing, Bugs Management, Configuration Management, liability Management, Evaluation & identification – Testing tools, test scheduling, resource sharing.
- **Designing:** Test Scenarios Identification, Test coverage and Traceability Matrix & Test scripts preparation, Test Case preparation and Test data, Test case reviews and Approval, Base lining under Configuration Management.
- **Test Environment Setup:** Network connectivity, Test Bed installation and configuration, All the tools or Software Installation and configuration, Coordination with Merchants and others.
- **Automation:** Automation Tool Identification and Evaluation, Analysing and Designing Framework and scripting, Script Assimilation, Audit and Approval, Guideline under Configuration Management.
- **Execution and Bug Tracking:** Test case execution, Test Scripts testing, Acquisition, audit and evaluate Test Results, Raise the bugs and finding for its termination.
- **Acceptance and Reports Generation**: Test Metrics, Test reports and process Enhancement made, Build release, Receiving acceptance.

4.1.7 Generic Software Testing Terms

- **Software Testing:** Software Testing is an essential activity to discover all the errors and bugs in the software before actual deployment of the product.
- **Verification:** Checking of software documents, code, design and program. It doesn't involve test execution. It use uses methods like auditing, inspections, walk-through, etc.
- **Validation:** It is the dynamic method of validating and testing the actual product. It does include executions. It uses methods like white box, black box, etc.
- **Quality Assurance (QA):** A group of activities to make sure that the maintenance or/and development process is adequate to make sure that the system meets its objectives. A standardized and planned set of activities necessary to provide adequate confidence that requirements are properly established and services or products conform to specified requirements. It doesn't involve executions.
- **Quality Control (QC):** The process by which the quality of product is correlated with applicable standards and the action is taken when non uniformity is detected. It always involves executions.

4.1.8 Types of Testing

There are three broad types of testing:

- **Functional testing:** The type of testing which checks that every function of a software application works in accordance to the requirement specification is known as functional testing. In essence, it covers user acceptance testing, unit testing, and integration testing. By providing some input (both valid and invalid) and watching the corresponding output produced, the system's functionality is tested. This kind of testing is simple to perform manually.
- **Non-functional testing:** Non-functional testing refers to the type of testing that examines the non-functional aspects of a software application, such as its scalability, usability, endurance, etc. Functional and non-functional testing are both essential. In essence, it includes usability, performance, and load testing. This type of testing is bit difficult to be carried out manually. The maintenance testing process is carried out after the software application has been deployed and after any alterations or improvements have been made. In essence, it includes both maintenance and regression testing.

4.1.9 Manual vs. Automated Testing

Testing an application manually by a human is referred to as manual testing. A quality assurance specialist (tester) verifies that an application functions correctly by running written test cases. By clicking through the various components of an application, the tester evaluates the application's performance, design, and functionality. Manual testing is useful when automation testing is not an option. Typical types of testing include exploratory testing, usability testing, ad-hoc testing, etc. Testing that is pre-scripted and executed automatically is referred to as automated testing. The tests are run to verify test results by comparing the results to what was anticipated. Automated testing is required when the same test needs to be executed several times in different contexts.

For instance: Acceptance testing, unit testing, integration testing, and system testing. Below Table presents the comparative study between automated and manual testing process.

Automated Testing	Manual Testing
Automated testing is more reliable. It performs same test N number of time, reducing human error.	Manual Testing is less reliable. Due to human errors, manual testing cannot be accurate every time.
The cost of Automated test depends on the cost of testing tools used for testing.	The amount of labour required to do manual testing determines its cost.
Due to scripts Automated testing is faster and less time consuming	Due to human interference Manual testing is slow and time consuming.
Automated testing is used when test cases needs to be executed repeatedly.	Manual testing in used when two or three test cases have to be executed
Automated testing ensures the user friendliness of the application	Manual testing doesn't ensure the user friendliness of the application.
Automated testing can be executed in parallel.	Manual testing can also be done in parallel but it hypes the testing cost
Programming knowledge is a must in Automated testing.	Programming knowledge is not necessary in Automated testing.

Table 4.1.2 Comparison between Manual and Automated Testing

4.1.10 Latest Versions of Software Testing Tools for Test Cases and Automated Scripts

Although there are many test automation tools available, not every one of them will be ideal for your project. Test engineers of various levels can use the tools on this list to test products created in various languages for various platforms.

Tool	Description
Selenium	Selenium is a testing tool, used to automate tests which are performed on web browser. It can be executed in multiple browsers. It is compatible to multiple programming languages.
Cucumber	Cucumber is an open source tool which supports Behavior Driven Development (BDD). It can be defined as testing framework, which uses plain English (Gherkin Language). It is multiple platform compatible, for example- Ruby On Rails, Selenium, PicoContainer, etc.
Ranorex	Ranorex is an all in one tool for mobile, web and desktop testing. It provides an easy click and go interface for beginners and an powerful IDE for automation experts. It is an licensed software.
Testsigma	Testsigma is one of the best automation tools available in market. It is best suited for DevOps and Agile market. It is an AI-driven tool which automates complex tests using simple English. No programming. CI/CD support.
LamdaTest	LamdaTest is one of the best cross browser test automation tool. Run Selenium automation tests on a secure, scalable and reliable cloud based Selenium Grid.

Table 4.1.1 Software Testing Tools

Software testing is a continuous process that is impacted incrementally by testing the applications, modules, subsystems, and unit processes that are at the lowest level. The testing methodology separates out software error, fault, and failure to attack the general issue of quality attainment. Error is described as discrepancy, fault as malfunction, and failure is described as incapacity. In each of these scenarios, the software must be tested; the testing must be done in a way that reveals the software's shortcomings. While automated testing tools reduced the cost and time of testing, manual testing is a labor-intensive process and therefore less cost-effective.

4.1.11 Method of Logging: Test Progress, Results, Defects & Uses of Agreed Test Management Tool

Teams creating, testing, and running software systems benefit greatly from the significant new capabilities offered by modern log aggregation & search tools. We gain deep insights into application behavior and health by treating logging as a core system component and using techniques like unique event IDs, transaction tracing, and structured log output, especially cross-component visibility.

Page by page, reviewer by reviewer, and either by the author or by a scribe, the difficulties, such as flaws, that were found during preparation are documented. For formal review types like an inspection, a separate person (a scribe) to do the logging is especially helpful.

No actual discussion is permitted while logging is taking place in order to ensure efficiency and progress. The discussion phase is where an issue is handled if it needs to be discussed. It is much more efficient to log an issue and move on to the next one rather than having a lengthy discussion about whether or not it is a defect. Additionally, a discussed and rejected defect may very well turn out to be real during rework, despite the team's opinion. Every flaw should be recorded along with its severity. The participant who finds the flaw suggests the degree of severity. Severity categories include:

- **Critical:** Defects that affect more than just the document being inspected will have downstream effects.
- Significant flaws could have an impact in the future (e.g. a fault in a design can result in an error in the implementation).
- Minor flaws are unlikely to result in further harm (e.g. non-compli ance with the standards and templates).

Spelling mistakes are not included in the defect classification in order to preserve the added value of reviews. Spelling errors are noted in the document under review by the participants and given to the author at the conclusion of the meeting, or they could be addressed in a separate proofreading exercise. The goal of the logging phase is to record as many defects as you can within a set amount of time. The moderator strives to maintain a high logging rate (number of defects reported per minute) in order to guarantee this. The logging rate must be between 1-2 defects per minute in a well-run and orderly formal review meeting.

4.1.12 Defects Management

A defect from the standpoint of software development is more than just an error message or a system crash brought on by a coding error. Any difference between the actual and anticipated outcome is a defect, such as when software performs exactly as intended despite working flawlessly.

A software requirements specification's expectations are not met when there is a defect. When functional or performance issues go undetected by pre-production tests, defects also appear in live software. Here are six illustrations:

- **Errors, omissions, or holes in the original software specifications** These flaws may appear as a result of a requirement being overlooked, poorly phrased, misunderstood by stakeholders, or misinterpreted by developers.
- **Errors in the software's architecture or design** These issues arise when software developers design ineffective software algorithms or processes, or when those algorithms or processes don't produce results with the necessary precision.
- **Coding mistakes or implementation mistakes**. Traditional errors like missing brackets and ungentle error handling are among these flaws. These defects include traditional bugs caused by everything from missing brackets to ungraceful error handling.

- **Errors in the test planning or test activities.** These defects stem from inadequately tested features and functions. These flaws result from features and functions that weren't thoroughly tested.
- **Mistakes or oversights made during deployment.** An example of these defects would be when a team provisions inadequate V M resources.
- **Errors in the process or policies a team uses to govern the development cycle**. These defects crop up when, for example, a team obtains signoffs or approvals without adequate design, coding or testing review.

The team can take proactive measures to fix the flaw and stop it from happening again once root cause analysis identifies the problem. If, for example, a design mistake resulted in the problem, developers can study the design and requirements papers and make modifications. If a problem was caused by a flaw in the testing procedure, developers can change the test cases and metrics.

4.1.13 Methods for Conducting a Root Cause Analysis of Software Flaws

Software teams can draw from numerous tools to approach root cause analysis tasks, including:

- Fishbone diagrams
- Five Whys
- Scatter plots
- Failure Mode and Effects Analysis (FMEA)
- Pareto charts

Fishbone diagrams and Five Whys are the most popular techniques.

Fishbone diagram. The goal of a fishbone analysis, also known as an Ishikawa diagram or cause-and-effect diagram, is to group potential causes into subcategories that branch off from the main problem in order to aid analysts in visualising a root cause. The figure that results bears a striking resemblance to a fish skeleton, hence the name. The figure that results bears a striking resemblance to a fish skeleton, hence the name. In actuality, the "head" of the fish is where the underlying issue or problem is written. The "bones" of the image are composed of the categories of potential causes. The next step is for analysts to determine the primary causes for each category; if additional secondary and tertiary causes are required, they can be added.

Fig. 4.1.1 Ishikawa diagram for the 5 Whys

- 5 Whys analysis. People might delve further into a problem by asking "why," which opens up new possibilities. The response to each why serves as the foundation for the following inquiry. The process is comparable to a toddler asking a series of why questions; each time an adult responds, the youngster builds on that response to ask a new question. The method makes use of brainstorming.

Take a look at an example with only four "whys":

- Why is the log file missing?
- The log file is not found in the expected logical unit number or folder.
- Why wasn't the log file there?
- The software application did not have the log file enabled.
- Why wasn't the log feature turned on?
- The software application's configuration was incorrect.
- Why was the software not configured properly?

A team did not finish a process to install and operate the software, or they did not properly document the application. The best solution might be to turn on the log and offer improved documentation and user instruction.

4.1.14 Programming Language and Its Uses

Without a doubt, the traditional manual testing methods used by organizations are being replaced by automation testing in the modern era. All of this is taking place because automation testing, when compared to manual testing, is more laboreffective, efficient, and scalable while also providing better performance and incurring lower operational costs. Automation testing primarily focuses on the process of developing test cases using a variety of tools and software and then executing those test cases while taking into account pre-defined actions.

Python

Without a doubt, the traditional manual testing methods used by organizations are being replaced by automation testing in the modern era. All of this is taking place because automation testing, when compared to manual testing, is more labor effective, efficient, and scalable while also providing better performance and incurring lower operational costs. Automation testing primarily focuses on the process of developing test cases using a variety of tools and software and then executing those test cases while taking into account pre-defined actions.

JavaScript

JavaScript excels at redefining customer expectations through front-end development and places a high emphasis on test automation. Numerous web applications, including Instagram, Accenture, Slack, and Air bnb, inexorably support libraries created with JavaScript automation, including instauto, ATOM Accenture Test Automation Open source Modular Libraries), Botkit, and Mavericks. Unavoidably, many web applications, including Slack, Instagram, Accenture, and Airbnb, offer JavaScript automation libraries, including instauto, ATOM Accenture Test Automation Open source Modular Libraries), Mavericks, and Bot kit.

C#

According to a Stack Over flow survey, more than 60% of users endorse the C# programming language for meeting the development and automation needs of a commercial enterprise.

C# is steadily booming thanks to its automation testing frameworks, which were developed by Microsoft's skilled developers. Microsoft's seasoned developers created C#, which is steadily booming thanks to its automation testing tools.

Ruby

Another programming language supported by the MVC Architecture is Ruby, which is growing in popularity in business sectors that need automation. It's possible that those topics will have to do with risk management, compliance, logistics, or hiring. A further intriguing aspect of Ruby is its potential to facilitate Selenium Automation Testers' work environments, enabling them to implement cross-border testing and its related processes with fewer lines of code.

Java

Larger firms are choosing Java as their second most essential technology in order to increase job prospects in the field of automation testing. Thanks to the general-purpose Java code(s) held by Oracle Corporation, more than 2 billion devices can synchronise a variety of automation benefits enabled by the "Write Once, Run Anywhere" premise of this multipurpose language.

Summary

- Test management is a procedure for controlling daily testing activities, from developing test cases through carrying them out and recording the findings.
- Test management tools are required for Managing and Tracking Test Activities, has test assigning feature, traceability and to eliminate complexity and duplication.
- Some of the best Testing Management tools are; Spiratest , Tes trail, X-ray, Zephyr Scale, Zephyr Squad, PractiTest, Testlink, Qmetry etc.
- The pre-scripted test which runs automatically is known as Automated Testing. The tests run to assert test results comparing actual result with expected result
- Latest versions of software testing tools for test cases and automated scripts are Selenium, Cucumber, Ranorex etc.
- A defect is any difference between a result that really occurs and one that was anticipated, such as when software performs as intended while working flawlessly.
- Software teams can utilise a variety of methods, such as fishbone diagrams and the Five Whys are Major, to address root cause analysis assignments.
- Compared to manual testing, automation testing is more labor- effective, efficient, and scalable, offers superior performance, and has lower operating expenses.

Activitiy

Lab Session

- In this activity, the trainer asks to pair up amongst themselves.
- Each pair will be given with a Topic on which they have to give presentation from the below listed topics from Management tools.
  - o SpiraTest by Inflectra/TestRail/XRay – Test Management For Jira/Zephyr Scale/Zephyr Squad/PractiTest/TestLink/QTest/QMetry Test Management/Kualitee
- Trainer ask to choose the issue from the list below or open to choose as per their wish:
- Encourage student for active participation.
- After completion the task they will submit it to the trainer.
- Trainer will appreciate the best filled logging sample and accolade for the best efforts.

Discussion Session

- The trainer will inquire whether the trainees comprehended the chapter.
- In this activity, the trainer will ask the trainee some questions related to the topic.
- The trainees will answer the questions according to their understanding of the topic.
- In case, the trainees have queries and confusions in their minds; they can put forth those in front of the trainer.
- The trainer will ensure that he or she can give appropriate answers to the queries placed by the trainees.

Exercise

A. Answer the Following Questions:

1. List out key programming language and its uses.
2. Explain below tools to approach root cause analysis tasks
   a) Fishbone diagrams
   b) Five Whys
3. Latest versions of software testing tools for test cases and automated scripts:
4. Explain Need For Test Management Tool.

B. Choose the Correct Answer:

1. Which test tool is used for managing scheduling, tracking and analysis
   a) Test management tool
   b) Test comparator tool
   c) Test execution tool
   d) Performance testing tool
2. What are the signs that state that a software project is in trouble
   a) The product scope is inappropriately defined.
   b) Deadlines are unrealistic.
   c) Changes are managed poorly.
   d) All of the above

Notes

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

IT - ITeS SSC
NASSCOM

5. Contribute to Quality Assurance of Projects

Unit 4.1 - Contribute to Quality Assurance of Projects

SSC/N1302

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Demonstrate the ability to identify the checkpoints that a project should comply with during each phase of the project.
2. Prepare reports on checks carried out, data and information collected, and risks identified.

UNIT 5.1: Contribute to Quality Assurance of Projects

Unit Objectives

By the end of this unit, participants will be able to:

1. Discuss the principles of effective quality testing on data.
2. Collate required data/information against key indicators using standard templates and tools.
3. Discuss the hierarchy of approval process in quality team.
4. Discuss with experts, any issues related to project data, where necessary.
5. Demonstrate regressive/progressive analysis of data to accurately identify risks to projects.
6. Apply the review process of the test plan, test cases and/or automated scripts.
7. Examine the impact of failure test or stress test of data on quality.
8. Demonstrate the use of chart/bar diagrams to project milestones to management reviewers, internal auditors, and technical reviewers.
9. Identify various test plans, test cases for monitoring automated scripts.

5.1.1 Scope of Quality Assurance in Software Testing

The testing can cover a wide range of topics, including standards and limitations for systems administration, business requirements, design requirements, programmer's code, and hardware configuration. The scope of testing may also include testing in relation to professional best practices and industry standards. As a result, testing offers a chance to validate and confirm all facets of software engineering. Software testing thus offers an organization a number of advantages. Due to the early detection of defects, time and money are saved. Testing makes sure that there is less downtime for the product, which increases customer satisfaction.

The task of the person in charge of software quality assurance is to analyze and assess the current software development procedures and find ways to make it better with the goal of avoiding the occurrence of bugs. A software quality assurance group's responsibilities and reach are broader than a software testing group's.

A quality assurance group's responsibility is to ensure that a product's quality is good, in accordance with the definition of assurance, which is "a guarantee or pledge" or "a freedom from doubt." A software project's standards and methodologies can be checked by the Quality Assurance (QA) group. It also monitors and evaluates the software development process to provide feedback solutions to the process problems and performs the testing process to decide whether the product is ready for release in the market.

Following are the major quality assurance tasks:

- To create a standardized procedure for software development and quality control and to make sure that the standards are not departed from.
- To develop quality factors, quality criteria, and quality metrics and define a complete set of quality factors.
- To establish standards at each stage of the process, such as requirement templates, design methodologies, coding standards, etc.
- To create checklists for every step of the process and verify the results of each step against the subsequent guidelines and checklists.

5.1.2 Project Milestone Review Meeting

Reviewing the overall milestone progress and accomplishments at specific project life cycle milestones/phases is the goal of the milestone review meeting. The project team can meet with higher management and other stakeholders during this meeting to discuss project-related issues, share lessons learned, and offer improvements for upcoming milestones.

Tasks

- The project manager uses the milestone review report template to calculate milestone progress and create a milestone review report.
- According to the plan, the project manager sends the meeting request, agenda, and milestone review report to the appropriate stakeholders.
- According to the agenda, the project manager conducts the meeting to review the milestones.
- Every items on the agenda to be discussed, and problems, suggestions for improvements, lessons learned, and action items are recorded in the issue management system.
- The project and customer contact person receives a milestone review report from the project manager.
- The project manager delegated the actions to the proper stakeholders and secured their commitment.
- Project Manager monitors the completion of the action items.

5.1.3 Guidelines for Milestone Review Meeting

1. The milestone review meeting may be held concurrently with the weekly higher management status meeting or at a later date after the completion of the milestones.
2. The milestone review meeting's pertinent stakeholders typically include the following, but other stakeholders may be invited as needed:
    - Development Manager
    - QA Manager
    - Process Engineering Lead
    - Configuration Controller
    - Project Team (Quality Assurance and Development)
3. Generally, the agenda for the milestone review meeting comprises the following items (but other items may be added if necessary):
    - Examine the overall development of each milestone in terms of effort and schedule variance.
    - Analysis of Customer Feedback (if any).
    - Problems encountered by the team during a milestone.
    - Lessons learned throughout milestone and improvement ideas for following milestones.

5.1.4 QA Process and Development Steps

Initially, QA contributes to the planning and management of the software development process in a manner that helps avert major problems throughout the project. To achieve this, QA engineers collaborate with other experts from the product owner, project manager, business analyst, and developers throughout the entire software development cycle as a crucial part of the software development team on the project. Depending on the nature of the work and the aims of the project, the number and order of quality assurance activities may vary from project to project.

Development process

- Analysis of requirements
- Design
- Implementation
- Verification or testing
- Maintenance

QA process

- Review of requirements
- Test planning/writing test cases
- Unit testing
- Integration testing
- System testing
- Performance testing
- Security testing
- Cross-browser testing / cross-platform testing
- Updating test cases
- Regression testing

1. **Step one:** review requirements and documentation

   Parallel to the creation of the documentation, the QA engineers begin working on the project. They examine the prerequisites and supporting paperwork for:

   - Completeness
   - Redundancies
   - Clarity
   - Consistency
   - Executability
   - Verifiability

The objective is to investigate system architecture and technologies for inconsistencies.

Key benefits for the development process:

- Errors cost less when discovered early on.
- Better documentation results in a higher quality project with less labour input and more precise estimations.

Consider using specialized software for reviewing documentation for this step, such as Confluence. You can compile all relevant materials that are used for the entire project and keep an internal knowledge base. When any requirements or documents are adjusted, added, updated, or removed, all team members are notified immediately.

2. **Step two:** Plan and prepare test cases

Once the specifications have been established, it's time to start developing test cases, which specify the procedures QA engineers use to ensure the software works as intended. You can also use specialized tools like TestRail or Zephyr for writing test cases if the volume of these cases ends up being quite large. Both programs enable the creation and modification of tests and the use of metrics to monitor outcomes.

3. **Step three:** design test cases

The QA team runs the test cases after the development phase is complete. The primary objective of this phase is to check if the solution was technically produced correctly and if it meets the original product owner's specifications.

The primary QA activities and their respective objectives are detailed below.:

- Smoke testing. QA engineers verify that the software or its module operates as intended. When passed, additional inquiry will commence.
- Integration testing. Verification that distinct elements function as a unified system.
- Performance testing. Test the system's behaviour under normal and predicted peak loads.
- Stress testing establishes the critical load beyond which the system fails.
- Security testing. Ensuring the solution has an adequate level of security.
- Cross-browser testing/cross-platform testing. Verifying an application's compatibility with several browsers (Chrome, Mozilla, and Safari) or platforms (Android, iOS, Windows Phone). This is particularly crucial for web and hybrid apps.
- Regression testing. Identifying problems within previously tested code. Typically required when adding new features or updating an old system. Again, it is possible to automate testing (e.g., unit testing, regression testing). Rule of thumb: the longer a project lasts, the more crucial automated testing becomes.

4. **Step four:** Report and measure

When a quality assurance engineer finds a bug, they log it in a bug tracking system, which doubles as a project management tool. You can use Redmine or Jira for this, both of which are highly configurable programs. All team members can view real-time task updates, and they make it simple to track issues of any severity, from a broken login form to security issues. This makes team communication easier and aids in maintaining a clear picture of the improvement process.

Each issue is assigned a priority level ranging from high to low, and the development team resolves them based on available time and resources.

5. **Step five:** Verifying fixes

The responsible QA engineers are notified when a developer resolves a problem, and they confirm it. When no problem is found, the bug tracking system closes the ticket. According to this rule, no bug can be marked as repaired until it has been validated.

Hints and tips to improve your QA process

A Waterfall development methodology is compatible with all of the testing steps mentioned above. In complex, long-term projects and/or sensitive industries, such as healthcare or logistics, where problems and errors are likely to arise, it is appropriate to devote time and energy in producing thorough document ation.QA processes such as Review of Requirements and Test Planning/Test Case Writing are widely employed..

When time and resources are limited, it is preferable to focus on software stabilisation and improvement than than producing lengthy documentation and exhaustive test procedures..Everyone wants their project to be successful. However, when it's time for a project audit, we snub it. However, conducting a project management review is a great way to show off your abilities and your level of project control. Your client, sponsor, and stakeholders will feel more secure as a result. Additionally, it's a way to learn and encourage your team and project to achieve true excellence.

Software Review

Software review is the systematic study of software performed by one or more people in the early stages of the Software Development Life Cycle (SDLC) in order to discover and correct software flaws and defects. Software review aids software engineers in validating the software's quality, functionality, and other key features and components as part of the Software Development Life Cycle (SDLC). It is a comprehensive process that includes testing the software product to ensure that it satisfies the client's requirements.

Typically, manual software reviews are conducted to validate a number of documents, such as requirements, system designs, codes, test plans, and test cases.

Objectives of Software Review:

The objective of software review is:

- To increase the development team's output.
- To accelerate and lower the cost of the testing procedure.
- To create software with fewer flaws in the final product.
- To make the deficiencies disappear.

Process of Software Review:

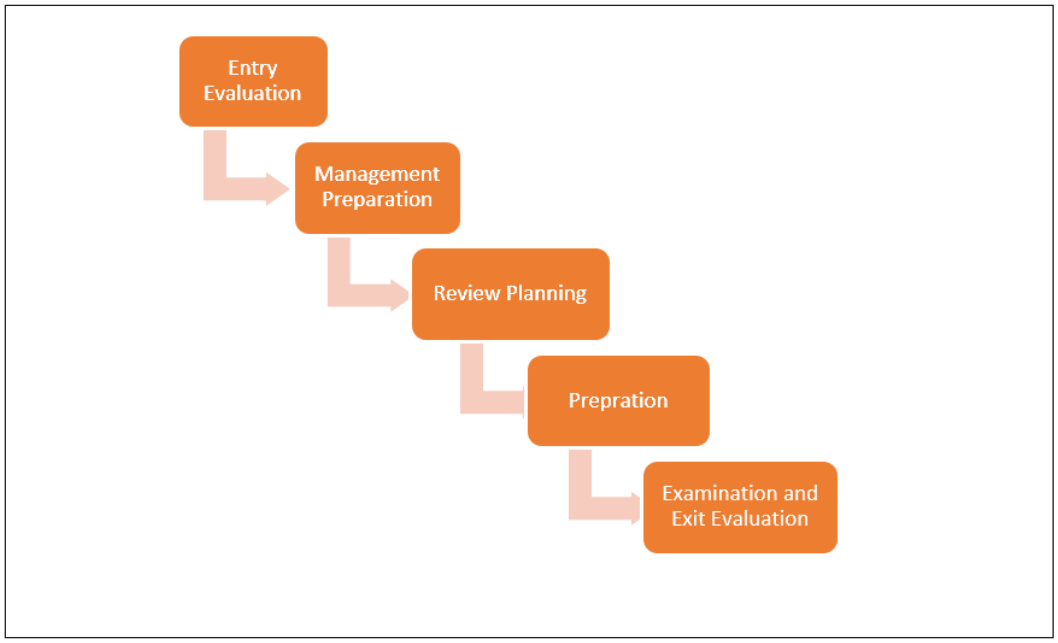


Fig. 5.1.1 Software Review Process

Types of Software Reviews:

There are primarily three types of software reviews:

1. **Software Peer Review:**

   Peer review is the process of evaluating the technical content and quality of a product, and it is often undertaken by the product's author and other developers. Peer review is a process that evaluates the technical content and quality of the product, and it is typically carried out by the work product's author in collaboration with a few other developers.

   Peer review is carried out to examine or fix software flaws, the quality of which is also verified by other team members.

   There are several types of peer review:

   - **Code Review:** Computer source code is systematically analysed.
   - **Pair Programming:** It is a code review in which two developers collaborate on the same platform to create code.
   - **Walkthrough:** Members of the development team are directed by the author and other interested parties, while participants pose questions and provide feedback on flaws.
   - **Technical Review:** A team of highly qualified persons checks the software product for its client's use and detects any technical problems based on the product's specifications and standards.
   - **Inspection:** In inspection, reviewers use a well-defined procedure to identify flaws.

2. Software Management Review

The work status is evaluated by the software management review. Decisions are made in this section regarding activities that will occur downstream.

3. Software Audit Review

"Software Audit Review" is a type of external review in which one or more critics who are not part of the development team set up an unbiased look at the software product and its processes to see if they meet the stated requirements and standards. Managerial employees are responsible for this.

Advantages of Software Review:

- Defects can be detected early in the development process (especially in formal review).
- Earlier inspection also minimises the cost of software maintenance.
- It can be used for technical author training.
- It can be used to eliminate process deficiencies that promote faults.

5.1.5 Conducting Technical Expert Review

Software engineers can find logical and functional errors early on by performing a formal technical review (FTR). At an FTR, participants in specific roles ensure that the software being developed complies with the established standards and requirements.

Software engineers do a formal technical review (FTR) to identify functional and logical flaws at an early stage. At an FTR, participants in designated roles guarantee that the developed software conforms with the stated standards and requirements.

When your product is fully developed, that is the ideal time to run an FTR. However, it varies with the kind of review. A team of engineers with designated responsibilities as speakers, reviewers, or producers is necessary for a typical FTR. A review report should be prepared following each meeting to address the following issues:

- What was reviewed?
- Who reviewed it?
- What findings and decisions were made?

The FTR indicates a category of reviews that consists of the following:

- The author of a product presents their work at a formal evaluation or review meeting. The primary purpose is to introduce the product to the remaining reviewers. Consequently, all participants must accept the output, offer adjustments, and discuss deadlines.
- A walkthrough is a meeting in which reviewers evaluate the source code, design, and stated requirements of the referred-to product. A walkthrough meeting is organised to identify code defects. Frequently, the code's author is present to answer queries.
- An inspection is a review session that determines the product's extra properties in accordance with the specifications. While formal reviews and walkthroughs are used to uncover flaws, inspections are conducted to expand initial standards or to ensure that earlier bugs have not returned.
- Formal technical reviews aid in error prevention and lower the likelihood of logical and implementation mistakes in advance. In addition, it assists a production team in observing the entire product's characteristics, making development more controllable.

5.1.6 RCA (Root Cause Analysis)

RCA (Root Cause Analysis) is a technique for identifying the root cause of defects. We use brainstorming, reading, and digging to determine whether a defect was caused by "testing miss," "development miss," or "requirement or design miss."

Accurate RCA helps to stop defects from appearing in later releases or phases. If it turns out that a defect was caused by a design error, we can review the design documentation and take the necessary action. Similar to this, we can review our test cases or metrics and make the appropriate updates if we discover that a defect was caused by a testing miss.

Testing only the defects should not be the sole focus of RCA. We can also perform RCA on production flaws. We can improve our Test Bed and add those production tickets as Regression Test cases based on the RCA's conclusion. This will prevent the defect or other defects of a similar nature from happening again.

The method of conducting a Root Cause Analysis resembles that of a physician treating a patient. The physician will initially identify the symptoms. Then, in order to determine the underlying causes of the ailment, he will conduct laboratory tests.

If the underlying reason of the sickness is unknown, the physician will refer for scan tests. He will continue to diagnose and do research until he identifies the root cause of the patient's condition. In any industry, the same concept applies to Root Cause Analysis.

Consequently, the objective of RCA is to identify the root cause as opposed to treating the symptom, using a defined set of processes and accompanying instruments.It differs from defect analysis, troubleshooting, and other problem-solving techniques because these techniques look for a fix for a particular problem, whereas RCA looks for the root of the problem.

5.1.7 Practicing Effective Documentation

Let's start with why effective documentation is so important. What benefits will your company experience in exchange for prioritizing the development of team documentation skills? There are four specific benefits that we want to cover in this article:

1. **Increased Productivity**

   Effective documentation boosts productivity and efficiency at work. Think about this: New hires can learn their jobs more quickly with clear documentation. Additionally, if important company documents are consistently stored in the same secure location, your team won't have to waste time looking for them.

   Although there are countless other examples we could use, these two nicely illustrate our point. Teams work more effectively when documents are accessible and understandable.

2. **Better Progress Monitoring**

   You'll be able to keep track of how your team is doing on specific projects and initiatives if you maintain effective documentation. It will be simple to determine Stuart's level of involvement with the fresh content marketing initiative and the number of applications Jenny has been able to sort through.

   Your team will remain focused and motivated to work toward the following goal if you track their progress in this way. It will also help you manage deadlines better.

3. **Improved Customer Service**

   If you are creating documentation for the customers of your business, you will be able to provide first-rate customer support. Instead of typing out lengthy emails when a customer emails your team with a question, they can just point them in the direction of a specific document.

   This makes giving both your team and customers customer service easier and more effective. To put it another way, everybody wins.

4. **Boosted Company Value**

   Lastly, effective documentation can increase the value of a company. If you own the company that you run, you may wish to sell it one day and cash in on your efforts. This will be difficult to accomplish if you have not kept accurate records along the way.

   Any prospective purchaser will be interested in the techniques you've built to accomplish success. If you can prove these facts, you can use them to negotiate a higher selling price.

5.1.8 Test Plan

A test plan is the fundamental document referred to when testing software. A project's comprehensive test plan is the duty of the test manager. Test planning reveals the entire project's testing schedule and approach. Hence, it is important to know the amount of effort and the kind of information that goes into a test plan. A good test plan will help to efficiently communicate and document the test effort with well constructed test plans, test cases, and test reports. It will also help the testers in achieving the desired quality for the product being tested.

5.1.9 Test Planning

Throughout the development process, testing is performed. For testing to be conducted, it is necessary to plan, design, and define performance measures. The activities involved in the testing process can be divided into phases, which begin in the design stage and end when the software is installed at the customer's site.

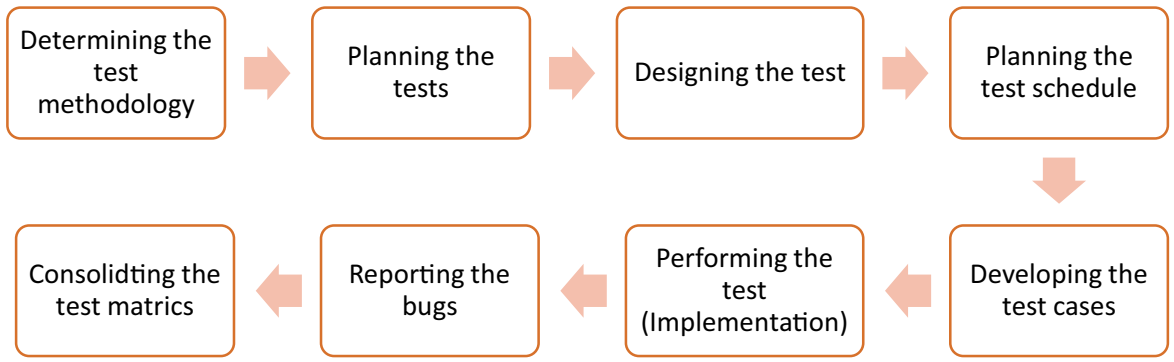


Fig. 5.1.2 A Typical Planning

The main purpose of implementing testing is to test the effectiveness and efficiency of the software. It is an effort to reduce the number of undetected errors present in the system or software being tested. Despite all measures taken to identify and remove errors, obtaining software that is free of defects is still an unrealizable goal. This challenge requires the testers to maintain high quality of testing in the software. Two ways adopted to improve the testing process are to upgrade the effectiveness of the test cases applied during testing and to develop automatic software testing tools. A tester can successfully complete the task of accurately communicating and documenting the test activity with the aid of well-written test plans, test cases, and test reports.

5.1.10 Writing a Test Plan

1. **Learn Software**

   Before evaluating the software, it is essential to learn everything possible about it. To find out more about how it was created, its intended use, how it functions, and to gather details that may help you comprehend its functionality, ask questions about how it was created. You can develop test cases that are pertinent and helpful for evaluating your product by properly understanding your software.

2. **Define the Scope of Testing**

   Making test documentation that is longer than the product itself is useless. Establish what will be tested specifically during the process, which modules or functions need to be covered in detail, and any other important details you should be aware of before you do anything else.

3. **Create Test Cases**

   Making test cases is one of the key steps in developing a software testing document. A test case is a written account of the procedures followed during your testing. It should contain details like:

   - What requires testing
   - How will it be evaluated?
   - Who will conduct the test?
   - Expected outcomes

Test Case Type	Description	Test Step	Expected Result	Status
Functionality	Area should accommodate up to 20 characters	Input up to 20 characters	All 20 characters in the request should be appropriate	Pass or Fail
Security	Verify password rules are working	Create a new password in accordance with rules	The user's password will be accepted if it adheres to the rules	Pass or Fail
Usability	Ensure all links are working properly	Have users click on various links on the page	Links will take users to another web page according to the on-page URL	Pass or Fail

Table 5.1.1 Spreadsheet for Setting up a Tet Case

4. **Develop a Test Strategy**

   The test strategy outlines your testing implementation strategy. Make sure that everyone on the team is aware of what they are supposed to be doing at any given time. Your testers should be operating according to the same game plan.

5. **Define the Test Objective**

   A test objective should be connected to each test case. The goal makes sure every action is pertinent and helps increase the value of your software for customers. Test goals may include the following:

   - Testing known features
   - Testing newly implemented features
   - Performing exploratory tests
   - Ensuring stability throughout the product life cycle

6. **Choose Testing Tools**

   To carry out your testing tasks, you must ensure that you have the appropriate software testing solution. Some of these tools might be computer programs, while others might need actual hardware, like test machines. It's crucial to select the proper tools for each individual task rather than relying on a generalized approach.

7. **Find Bugs Early**

   In your planning document, include time for "bug fixing" sessions. These enable you to spot software issues early on before they become too difficult or expensive to fix. This makes dealing with them simpler and less expensive. Verify any app security features, utilize all features, and look for anything that doesn't function properly.

8. **Define your Test Criteria**

   Even though it should be a part of the test case, it's still a good idea to describe it in more detail. Your objectives are essentially divided into smaller components called test criteria. They provide detailed information on how each objective will be accomplished, making it easier for you to keep track of your testing progress.

   There are standards that must be met prior to suspending testing. For instance, you might want to stop testing when a specific number of bugs have been discovered or when the software can't be used because of performance problems.

   Exit criteria are requirements that must be satisfied before testing is complete. For instance, the test case should be completed once all goals have been achieved and all bugs have been fixed.

9. **Resource Planning**

   Include a resource strategy detailing the number of individuals needed for the testing process in your software testing document. This should include a description of each individual's responsibilities and any necessary training.

Fig. 5.1.3 Resource Planning in Project Management

10. Plan your Test Environment

Include details about the environment where testing will occur in your test plan, such as:

- Test equipment necessary for product testing.
- Requirements for the sizing of software and servers.
- Platforms that the product supports.
- Additional environment-related details that could affect your testing procedure.

11. Plan Test Team Logistics

One of the most crucial elements of process implementation is test management. Your testing document won't be as helpful as it may be if you can't effectively communicate with your testers. This will hinder their development.

12. Schedule & Estimation

Include a timeline that enables you to specify specific testing milestones and deadlines in your test plan. Milestones could include the first public beta test, internal testing sessions, the product's initial release, or any other crucial junctures where your team needs to concentrate its efforts on testing.

13. Test Deliverables

A list of each deliverable needed for testing should be included in your testing documentation. To ensure that everyone is aware of the specific time at which they must be ready to act, these should be linked to the programme steps.

14. Test Automation

- Consider software test automation if your software is especially complicated and needs a lot of test cases.
- By automating the process, testers can complete more work in less time, increasing productivity and sharply lowering testing costs as a whole. Even a mobile bot may be useful for accelerating testing tasks.
- An effective test strategy is a crucial component of the project briefing. Your testing plan should be clear, succinct, adaptable, and take into account changes in your environment or schedule.
- To have the necessary resources available when you need them, planning and preparation are necessary in many testing-related aspects. Some resources, including people and places, could need a lot of planning. These resources are described in the test plan, along with the testing requirements.
- Planning and preparation are essential in many elements of testing in order to have the resources available when you need them. Some resources, such as people and locations, may require extensive planning. The test strategy outlines these resources as well as the testing specifications.
- Just keep in mind that no test plan is flawless, but with practise, creating test plans gets simpler.

Summary

- Testing assures the product's stability and minimises downtime, resulting in increased customer satisfaction.
- According to the definition of assurance as "a promise or pledge" or "freedom from question," the job of a quality assurance group is to ensure that the product's quality is high.
- The purpose of the milestone review meeting is to review the overall milestone progress and accomplishments at selected project milestones/phases.
- Throughout the whole software development cycle, QA engineers are a vital member of the software development team, working alongside other professionals such as the product owner, project manager, business analyst, and developers.
- Software review is the systematic inspection of a software during the early phases of the Software Development Life Cycle by one or more individuals who collaborate to detect and fix faults and problems (SDLC).
- Software engineers do a formal technical review (FTR) to identify functional and logical flaws at an early stage.
- RCA (Root Cause Analysis) is a method for identifying the root cause of defects. We use brainstorming, reading, and digging to determine whether a defect was caused by "testing miss," "development miss," or "requirement or design miss."
- Effective Documentation Practice has four distinct benefits:
- Enhanced Productivity, Improved Progress Tracking, Enhanced Customer Service, and Enhanced Company Value.

Activitiy

Lab Session

- In this activity, the trainer will divide the class into 10 equal pairs.
- All the groups will need to brief the class on Quality Assurance Process. Each pair will be given with below topic in which they need to tell the benefits as well.

  Team 1: Review of requirements

  Team 2: Test planning/writing test cases

  Team 3: Unit testing

  Team 4: Integration testing

  Team 5: System testing

  Team 6: Performance testing

  Team 7: Security testing

  Team 8: Cross-browser testing / cross-platform testing

  Team 9: Updating test cases

  Team 10:Regression testing

- The group completing the given task in the best way will be declared as the winner and will be appreciated in the class with accolades.

Discussion Session

- The trainer will inquire whether the trainees comprehended the chapter.
- In this activity, the trainer will ask the trainee some questions related to the topic.
- The trainees will answer the questions according to their understanding of the topic.
- In case, the trainees have queries and confusions in their minds; they can put forth those in front of the trainer.
- The trainer will ensure that he or she can give appropriate answers to the queries placed by the trainees.

Exercise

A. Tick the correction options agaist each statement.

1. Who identifies, documents, and verifies that corrections have been made to the software?

   a) Project manager b) Project team

   c) SQA group d) All of the mentioned

2. Software quality assurance consists of the auditing and reporting functions of management.

   a) True

   b) False

3. What does QA and QC stand for?

   a) Quality Assurance and Queuing Control b) Quality Adjustment and Quality completion

   c) Quality Assurance and Quality control d) Quality Adjustment and Queuing control

4. What is QA?

   a) It is the measurement of degree to which a product satisfies the need

   b) Any systematic process used to ensure quality in the process

   c) Process of identifying defects

   d) It is a corrective tool

5. Arrange the steps of QA in ascending order?

   a) Review requirements and documentation, plan and prepare test cases, design test cases, report and measure, verifying fixes

   b) Verifying fixes, plan and prepare test cases, design test cases, review requirements and documentation, report and measure

   c) Review requirements and documentation,verifying fixes, plan and prepare test cases, design test cases, report and measure

   d) Report and measure , review requirements and documentation,verifying fixes, plan and prepare test cases, design test cases,

B. Answer the below questions:

a) What does FTR stands for?

b) Define RCA (Root Cause Analysis)?

c) List out the steps of Software Review Process.

Notes

Scan the QR Code to watch the related videos

https://www.youtube.com/watch?v=TAjwPW2iULY

Discuss the scope of quality assurance for a Test Engineer

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

IT - ITeS SSC
NASSCOM

6. Key Indicators for Software Applications

Unit 6.1 - Key Indicators for Software Applications

SSC/N1303

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Identify the primary sources of key indicators for quality testing.
2. Examine the purpose of data or information provided to third party for quality check.

UNIT 6.1: Key Indicators for Software Applications

Unit Objectives

By the end of this unit, participants will be able to:

1. Identify the key factors impacting the quality assurance of projects and why projects must comply with these.
2. Discuss the types of risks and their indicators like unproven technologies, user and functional requirements, application, and system architecture, etc.
3. Define the checkpoints for third party performance in quality assurance.
4. Explain the factors that impact quality of software such as inadequate third-party performance, litigation in protecting intellectual property, obsolete software, and wrong software functionality.
5. Analyse how management reviewers, internal auditors, technical reviewers impact the purpose of data/information shared for software applications.

6.1.1 Basics of Impact Analysis in Software Testing

Software development is a continuous process where we continuously add new features or enhance existing functionality. However, every modification made to a product may have an effect on a specific component or even the entire product. However, every modification made to a product may have an effect on a specific component or even the entire product. In addition, tracking the effects of changes we make to a product gets harder as we make more of them.

Therefore, it might not be sufficient to simply test your updated product in accordance with industry best practices to guarantee its proper security and faultless performance. Some code sections may need to be verified twice, given a more thorough examination, or tested using a different strategy. A different testing strategy, a deeper analysis, or double-checking may be necessary for some sections of code. And conducting a thorough impact analysis is the best way to distinguish these components of your product.

A software testing strategy called impact analysis aids in defining all risks connected to any alterations made to the test product.

An impact analysis should be conducted whenever:

- A request has been made for the product to be changed.
- The specifications for products have changed.
- Current modules or features have undergone changes
- You want to put new modules or features in place.

Although this procedure might raise the overall cost of product development, the extra costs are easily justifiable.

What is impact analysis in testing?

There are three types of impact analysis that concentrate on various elements of this procedure and pursue various objectives:

- The main goal of dependency impact analysis is to identify dependencies, which can be potential side effects of changes or product components that need to be redesigned as a result of these changes.
- Experiential impact analysis aims to estimate the risks associated with product changes in terms of the whole development process, including the need for extra time and resources for development.
- Traceability impact analysis, according to the definition by ISTQB Glossary, assesses what must be changed at different documentation levels in order to implement a particular change to the product.

Calculating the TEI (Test Efficiency Indicator)

A defect severity point system and total defects counts in System Testing vs. UAT are used to calculate the Test Efficiency Indicator. Each defect that is logged is given a severity rating, which corresponds to a point ranking.

- Critical = 4
- Serious = 3
- Medium = 2
- Low =1

The total number of flaws discovered throughout the System and UAT testing cycles is then tallied, and the corresponding point ranking is determined. Any flaws discovered during UAT that cannot be reproduced in the initial testing setting are taken out of the analysis. After that, we divide the total number of defects discovered during system testing by the total number of defects discovered during system and UAT testing. The final percentage represents the testing procedure's effectiveness as a whole.

TEI equation: SIT Points / SIT + UAT Points = TEI

Generally, good testing processes have roughly a 90% TEI, with only 10-12% defect leakage. However, as stated above, QA Mentor aims higher than that for a 5% defect leakage.

6.1.2 The Basisof Software Testing Metrics

Software testing metrics, also referred to as software test measurement, show the breadth, depth, height, and capacity of a software process and attempt to immediately increase its effectiveness and efficiency. The most effective way to gauge and keep track of the various testing tasks carried out by the team of testers throughout the software testing life cycle is through the use of testing metrics. Additionally, it aids in communicating the outcome of a prediction based on a combination of data. As a result, software engineers all over the world use the following various software testing metrics:

Complying the Software Quality Indicators

Metrics and indicators usually play an important role in software quality and maturity. They provide a quick overview on the project and the direction it is moving. With metrics it is simpler to target improvements and asses them. However, metrics can have also negative impact on developers and the code itself. The problem occurs when people forget their original goal and fall into false proxy trap (quoted below). Typically, it is impossible to measure directly quality of a product, so we choose to measure something much easier - e.g. number of defects found. This measure becomes an approximation of the original measure we targeted for.

Management concerns can also be addressed with the help of quality indicators. Some of the recommended quality indicators are:

- **Progress:** It measures the amount of work done by the developer in each phase.
- **Stability:** It evaluates whether the products of each phase are adequately stable to proceed to the next phase.
- **Process Compliance:** It measures the developer's obedience with the development procedures approved during the start of the project.
- **Quality Evaluation Effort:** It measures the developer's effort that is being spent on internal quality evaluation activities.
- **Defect Detection Efficiency:** It measures how many defects were detected in a particular phase
- **Defect Removal Rate:** It measures the total number of defects detected and resolved over a period of time.
- **Defect Age Profile:** It measures the total number of defects that have not been resolved over a period of time.
- **Defect Density:** It identifies the defect-prone parts of the system.
- **Complexity**: It measures the complexity of the code in a particular program.

6.1.3 Software Testing Performance Indicators(KPIs) for Quality Assurance

A type of performance measurement, Organizations and testers both utilise performance indicators, or KPIs, to collect data that may be measured. KPIs are the exact specs that the software testing team measures and examines to make sure the process complies with the company's goals. Additionally, they support the team in taking any necessary actions if the product's performance falls short of the predetermined goals.

In a nutshell, key performance indicators (KPIs) are significant metrics that software testing teams compute to make sure the project is progressing properly and effectively attaining the target, which was established during the planning, strategy, and/or budget sessions. These are some of the several crucial KPIs for software testers:

Active Defects: Active Defects: This straightforward but crucial KPI helps determine whether an issue is new, open, or fixed and enables the team to take the appropriate action to address it. These are assessed using the team's threshold, and if they exceed it, they are flagged for quick action.

1. **Automated Tests:** It's crucial for the test manager to recognise the automated tests while tracking and examining the key performance indicators. Tricky enables the team to keep track of the number of automated tests that can be used to find and stop the critical and high priority flaws that are being introduced into the software delivery stream

2. **Covered Requirements:** The team can monitor the percentage of requirements that are covered by at least one test with the help of this key performance indicator. To guarantee complete test and requirement coverage, the test manager keeps an eye on this KPI every day.

3. **Authored Tests:** A further significant KPI, authored tests are examined by the test manager and test engineer in order to assess the test design activity of their business analysts and testing team.

4. **Passed Tests:** The team assesses/measures the proportion of passed tests by keeping track of the execution of each and every configuration contained within a test. This aids the team's comprehension of how well the test configurations detect and capture flaws during the testing process.

5. **Test Instances Executed:** The team uses this key performance indicator to show what proportion of the total instances in a test set have actually been executed. It is tied to the speed of the test execution plan. This KPI, however, does not provide information about the build quality.

6. **Test Executed:** After the test cases are selected, the team watches the various test execution methods, including manual, automated, and so forth. Just like test instances executed, this is also a velocity KPI.

7. **Defects Resolved Per Day:** By monitoring this KPI, the test manager may keep track of the number of defects fixed each day as well as the team's efforts to address these problems. Additionally, it enables them to follow the project's development and testing operations.

8. **Direct Coverage:** This KPI supports human or automated feature or component coverage and guarantees that all features and their functionalities are exhaustively evaluated. A component will be deemed incomplete if it is not tested during a given sprint and will not be moved until it is.
9. **Percentage of Critical & Escaped Problems:** Software testers should pay close attention to this crucial KPI, which measures the percentage of critical and escaped defects. It guarantees that the team's testing efforts are concentrated on fixing the product's most important problems and flaws, which in turn aids in ensuring the calibre of both the product and the overall testing process.
10. **Time to Test:** The goal of this key performance indicator is to provide the software testing team with information about how long it takes a feature to go from the "testing" stage to the "done" stage. It helps in understanding the complexity of the feature being tested as well as calculating the testers' effectiveness and efficiency.
11. **Defect Resolution Time:** This metric is used to assess how long it takes a team to identify software defects and to test and confirm a repair. In addition, it measures and qualifies the tester's responsibility and ownership for their bugs while keeping track of the resolution time. In other words, this KPI guarantees everything from tracking the defects to ensuring that they are corrected as intended to promptly closing the issue.
12. **Successful Sprint Count Ratio:** Although a software testing metric, once all the successful sprint statistics have been gathered, software testers utilise this as a KPI. They can use the following calculation to compute the proportion of sprints that are successful: Successful Sprint Count Ratio: (Successful Sprint / Total Number of Sprints) x 100.
13. **Quality Ratio:** The quality ratio, which is employed as both a software testing metric and a KPI, is based on the passed or failed rates of all the tests that the software testers conducted. This is calculated using the following formula: Quality Ratio = (Successful Test Cases / Total Test Cases) x 100.
14. Test case quality is a software testing statistic and KPI that aids in assessing and rating written test cases in accordance with the established standards. By creating high-quality test case scenarios or with the aid of sampling, it makes sure that all test cases are evaluated.

Moreover, to ensure the quality of the test cases, certain factors should be considered by the team, such as:

- They ought to be reviewed for errors and flaws.
- Complete test and requirement coverage is required.
- The regions that the flaws affect should be pointed out and specified.
- Test data must be accurate and comprehensively represent all potential scenarios.
- Both success and failure situations ought to be included.
- Expected results must be stated in a precise and understandable manner.

- **Defect Resolution Success Ratio:** The software testing team can determine how many bugs have been closed and then reopened by calculating this KPI. 100% resolution success is attained if none of the flaws are reopened. The following formula is used to evaluate the defect resolution success ratio: Defect Resolution Success Ratio = [(Total Number of Resolved Defects) - (Total Number of Reopened Defects) / (Total Number of Resolved Defects)] / (Total Number of Resolved Defects) x 100.
- **Process Adherence & Improvement:** This KPI can be used to reward the software testing team's efforts if they come up with any concepts or fixes that make the testing process easier to understand, more agile, and more precise.

These play a significant part in the software development lifecycle, from validating the product's quality to guaranteeing the accuracy of the numerous tests carried out by the testers. Consequently, you may improve the efficiency and accuracy of your testing efforts and achieve remarkable quality by implementing and using these software testing metrics and performance indicators .

6.1.4 Risk In Software Project Management

- **Five primary risk impact areas can be identified for the majority of software development projects:**
  - Emerging, untested technologies
  - User and functional requirements
  - System and application architecture
  - Performance
  - Organizational
- Emerging, untested technologies. The use of new technologies is a requirement for the majority of software projects. The likelihood that technology risks will materialize in practically any significant software engineering effort increases due to the constant change in tools, techniques, protocols, standards, and development systems. Training and knowledge are crucial, and the improper application of new technology frequently results in project failure.
- Specifications for users and functionality. All user needs in relation to the features, functions, and level of service provided by the software system are captured by the software requirements. The process of defining requirements is far too frequently drawn out, tiresome, and complicated. Additionally, requirements typically alter as a result of integration, prototyping, and discovery activities. Modifications to user requirements might not translate to changes in functional requirements, and changes to elemental requirements will probably spread throughout the entire project. These setbacks frequently result in one or more serious failures of a poorly thought out software development project.

- Architecture for applications and systems. A platform, component, or architecture can go horribly wrong if it is developed in the wrong way. It is crucial that the team consists of experts who are familiar with the architecture and have the capacity to make wise design decisions, just as it is with the technological risks.
- Performance. Any risk management strategy should take performance expectations from partners and users into account. To make sure the work products are heading in the right direction, benchmarks and threshold testing must be taken into account throughout the project.
- Organizational. Project management must make plans for the project's effective execution and strike a balance between the demands of the development team and those of the clientele. Project management must make plans for the project's effective execution and strike a balance between the demands of the development team and those of the clientele. Of course, adequate staffing includes choosing team members with skill sets that are a good match with the project.

6.1.5 Risks in Software Project

"Risks are unforeseeable future events with a high likelihood of suffering a loss. Software risks can be effectively analyzed, which will aid in risk analysis and assist in task planning for the future. We will comprehend the various risk types in this article. Risk-based testing is used in software development.

One may end up taking a fair amount of risk if they do not test the software with all of the potential inputs because there is a chance they will skip some of the inputs that are correct. One runs the risk of failing at this point, which could result in monetary loss, a loss of security, or even a loss of life. A software tester is under extreme pressure as a result of this. It is believed that software testing is a risk-based regime of practice, where one can find that:

- Excessive testing can raise the price of development.
- An organization may incur significant costs if the developed software fails due to insufficient testing.
- The overall cost of testing is higher due to missed bugs, overtesting, and undertesting.

Development risks: Development risks are inherent in any project. To understand their pervasiveness, and how they can be controlled, we should first define the concept. A development risk is “a state or property of a development task or environment, which, if ignored, will increase the chances of project failure”.

- **Technological gaps:** Lack of adequate and sufficient professional knowledge and experience to carry out the demands of the development contract.
- **Staff shortages:** Unanticipated shortfalls of professional staff.
- **Interdependence of organizational elements:** The likelihood that suppliers of specialized hardware or software subcontractors, for example, will not fulfill their obligations on schedule.

The risk management process includes the following activities:

Risk Identification, Risk Evaluation, Planning of Risk Management Actions (Rmas), Implementation of RMAS, and Monitoring of RMAS. Software RMAS are incorporated in the development plan.

Several lists of potential software development risks ("software risk items" or SRIs) are mentioned in the literature. Ropponen and Lyytinen (2000) have classified software risk items into the following six classes:

- Scheduling and timing risks
- System functionality risks
- Subcontracting risks
- Requirement management risks
- Resource usage and performance risks
- Personnel management risks.

Sr. No.	Software Risk	No.	Software Risk Item	Description
1	Personnel	1	Personnel shortfalls	Lack and turnover of management risks
2	Scheduling and timing	2	Unrealistic schedules	Incorrectly estimated (too low) development time and budget
3	System functionality	3	Developing wrong software functions	Development of software functions that are not needed or are incorrectly specified
		4	Developing wrong user interface	Inadequate or difficult user interface (GUI)
4	Requirement management	5	Gold plating	Addition of unnecessary features ("whistles and bells") due to professional interests, pride, or user demands
		6	Continuing stream of requirement changes	Uncontrolled and unpredictable changes in system functions and features
5	Subcontracting	7	Shortfalls in externally furnished components	Poor quality of externally delivered system components
		8	Shortfalls in externally performed tasks	Poor quality or unpredictable accomplishment of externally performed tasks
6	Resource usage and performance	9	Real-time performance shortfalls	Poor system performance
		10	Straining computer science capabilities	Inability to implement the system due to lack of technical solutions and/or computing power

Table 6.1.1 The top 10 software risk items

6.1.6 Performance Testing Difficulties and Strategies for Over Coming Them

1. **Selection of Wrong Performance Testing Tools**

   This is a frequent problem, and frequently the best performance testing tool is not chosen. The choice of tool is influenced by a number of variables, including the application communication protocol, the application technology stack, the performance tester's skill level, and the tool's licensing fees.

   The selection of the incorrect performance tool could result in lost testing time trying to get the test scripts to run, so it's crucial that it can identify the controls of the application being tested.

   Solution:

   The application under test (AUT) and the associated licensing costs must be carefully evaluated by the QA manager and QA team before selecting the best performance testing tool to guarantee the success of the testing process.

2. **Third-party performance -3rd Party Application Dependency**

   The performance of web applications is now significantly influenced by third-party web components. Your system may completely malfunction if it is dependent on data from a third party who becomes unavailable. The performance of your system may suffer as a result of the third party's subpar performance.

3. **Outsourcing Challenges**

   The primary causes of unsuccessful software development outsourcing projects include inadequately defining the project's requirements and scope, selecting the incorrect vendors and development methodologies, and failing to involve stakeholders and end users from various organizational levels in all stages of the development process. Inadequate milestone and progress tracking, a dearth of appropriate and proactive supplier-customer communication, and a shoddy definition of the work team's organizational structure are additional factors (Selleo, 2016a).

   Solution:

   Therefore, the goal of this research is to develop a flow of steps for organizing and managing outsourcing processes in order to close this gap. The analysis and classification of the aforementioned processes and best practices led to the development of the following steps for outsourcing management: planning, supplier selection, relationship management, development, and reconsideration.

4. **Improper analysis of performance test outcomes:**

   This is a major challenge for many testers because in-depth analysis of the performance test results requires substantial system and application knowledge.

Solution:

The testing process should be carried out by an experienced performance tester who can evaluate the scenarios, continuously improve the tests, and keep adding tests to make them consistent. Application architecture should be well-known to the performance tester.

The performance tester should also have knowledge of networking, data structures, OSI model, client-side and server-side performance, as well as OS concepts, web architecture, and OS concepts. These performance specialists will have quick access to the test results' analysis.

6.1.7 Third Party QA Testing

The importance of quality assurance and software testing in software development is comparable to that of the code itself. Yet, the software quality assurance (QA) procedure is typically performed last, right prior to the project's release.

Despite the fact that the QA testing process is frequently tedious and time-consuming, it plays a crucial role in ensuring the success of the entire testing effort, hence facilitating the delivery of reliable software and the best products. If well-planned and performed, the quality assurance process may ensure not only the success and high quality of the product, but also the continuity of business operations.

One of the smartest methods to ensure a favourable outcome from third-party QA testing is to employ a few test management best practises.

Best Methods for Third-Party QA Testing

The only way to produce a high-quality product is to implement an efficient QA procedure. Here are some third-party QA testing best practises that will help you achieve higher success in quality assurance:

1. **Prior to testing, develop a robust product and third-party application expertise**

   During testing, third-party application concerns are frequently confused with true product flaws. This can have a substantial impact on testing and your time-to-market.

   Before conducting test scenarios, it is crucial that the QA team has a comprehensive understanding of all domains (APIs, OS, mobile, etc.).

2. **Plan the testing and quality assurance procedure in depth**

   Planning, defining, and documenting the software testing process is one of the most effective approaches to increase the overall product quality.

Good documentation is an efficient means of fostering good communication within a software team and enables the maintenance of quality and testing plans for a project. Among the most important papers that can be shared with clients or stakeholders are:

- Test plan
- Supported systems
- Change log Test cases and checklists

3. Focus on Quality Enhancement

The primary purpose of any quality assurance process is to provide assurance to all stakeholders that the procedures and activities employed in the creation of your software are meticulously planned to preserve the product's high quality.

Therefore, QA testing should focus mostly on enhancing the software development process in order to optimise the quality of final products.

4. Utilize Efficient Methodologies

An efficient QA testing technique can be essential in minimising the entire cost of the software's lifespan. It assures that the software complies with all requirements and standards, which are vital to the creation of life-critical goods.

In addition, it is essential to identify the appropriate processes to be employed in the development of the product and to verify that these processes are followed precisely and without variation.

5. Give Risk Management precedence above Quality Assurance

Contrary to the common misconception, quality assurance is a considerably broader word than testing. To ensure the quality of your product, a vast array of processes, activities, and risk management are required.

In reality, risk management coupled with a quality check is one of the pillars of effective quality assurance.

6. Employ Test Methodologies Based on Requirements

To be able to conduct various sorts of tests, such as security, mobile, agile, big data, and analytics, you must employ various testing strategies that correspond to the particular type of test management framework.

Simply expressed, it is essential to follow a requirements-based testing strategy that can aid in system/application validation and the estimation of testing efforts.

7. Test Early and Frequently

The objective of quality assurance testing is to incorporate testing at each stage of a project's lifecycle in order to uncover problems early on. Within each level of testing, QA engineers test and retest the product to determine the viability of newly added features. This enables them to discover and address any issues that may have been introduced during this procedure.

Early and frequent testing requires constant process monitoring to ensure that all agreed-upon standards and procedures are being adhered to throughout the development process.

8. Automation

It is always a good idea to find novel approaches to automate repetitious test cases. This in no way implies investing a great deal of money to automate everything.

The optimal solution is to acquire the services of a test automation service provider and install servers capable of efficiently managing automated tests.

9. Practice to Achieve Perfection the First Time

Mistakes are inevitable in any delivery project, regardless of how strict the specifications. In software testing, getting it right the first time implies that all such potential flaws must be eradicated.

The objective is to define with a certain degree of precision what you are attempting to accomplish in the project, design a solution that clearly reflects your desired outcome, and ensure that any variations are recognised and accepted.

Summary

- Testing your updated product in accordance with industry best practices might not be enough to ensure its proper security and flawless performance. A different testing strategy, a deeper analysis, or double-checking may be necessary for some sections of code.
- A software testing technique called impact analysis helps to identify all risks associated with any alterations made to the test product.
- A defect severity point system and total defects counts in System Testing vs. UAT are used to calculate the Test Efficiency Indicator.
- Software testing metrics, also referred to as software test measurement, show the breadth, depth, height, and capacity of a software process and attempt to immediately increase its effectiveness and efficiency.
- Organizations and testers both use performance indicators, or KPIs, to collect data that can be measured.
- We can categorize the five main risk impact areas for the majority of software development projects as follows: new, unproven technologies, user and functional requirements, application and system architecture, performance, and organizational.

- Risks are unknown events in the future with the chances of loss and easy occurrence. Software risks can be effectively analyzed, which will aid in risk analysis and assist in task planning for the future.
- Risk evaluation, risk identification, planning of risk management actions (RMAs), execution of RMAs, and monitoring of RMAs are all components of the risk management process. Software RMAs are incorporated in the development plan
- The performance of your system may suffer as a result of the third party's subpar performance.

Activitiy

Discussion Session

- The Trainer asks the Trainees the following question: “What are the key performance assurance indu cators?”
- In this activity candidate need to raise their hands to volunteer and speak.
- The language spoken during the session should be known by the majority of Trainees in the class.
- While the session goes on, the Trainer should jot down the crucial points on the Whiteboard with the help of a marker.
- The best answer shall be appreciated by the Trainer in front of the whole class.

Discussion Session

- The trainer will inquire whether the trainees comprehended the chapter.
- In this activity, the trainer will ask the trainee some questions related to the topic.
- The trainees will answer the questions according to their understanding of the topic.
- In case, the trainees have queries and confusions in their minds; they can put forth those in front of the trainer.
- The trainer will ensure that he or she can give appropriate answers to the queries placed by the trainees.

Exercise

1. Choose the Correct Answers:

I. What is QA?

a). Process of identifying defects

b). It is a corrective tool

c). The degree to which a product satisfies a need is gauged by this metric.

d). Any methodical procedure applied to guarantee process quality

II. To identify issues and address them in order to raise the caliber of a project is the goal of _____.

a). Software bug

b). Software complexity

c). Software testing

d). Software development

III. Which is an example of an indicator?

a). Number of tests

b). Number of staff-hours

c). Actual versus planned task completions

d). Defects per thousand lines of code

IV. Reducing ____ is software assurance's main objective.

a). Risks

b). Time

c). Quality

V. FTR stands for ____.

a). File Transfer

b). Formal Telephonic Review

c). Formal Technical Review

d). Formal Telegraphic Review

VI. The factors influencing project management is/are ____.

a). Time

b). Cost

c). Scope

d). All of the above

2. List the performance challenges in testing.

Notes

Scan the QR Code to watch the related videos

https://www.youtube.com/watch?v=I1uT7mhalio

The purpose of impact indicator, efficiency indicator in testing

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

IT - ITeS SSC
NASSCOM

7. Technical Skills for Manual Tests

Unit 7.1 - Technical Skills for Manual Tests

SSC/N1303

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Design test suites or cases during the testing phase with 100% test coverage.
2. Examine the importance of source coding standards, and utilities/tools for handling quality assurance.

UNIT 7.1: Technical Skills for Manual Tests

Unit Objectives

By the end of this unit, participants will be able to:

1. Discuss latest changes, procedures, and practices in the field of designing test suites.
2. Examine the purpose of software testing elements, like static testing, dynamic testing, white box testing, black box testing, etc.
3. Discuss the use of different value and data to determine the correct action in manual testing.
4. Apply correct usage of information technology to input and/or extract data accurately.
5. Demonstrate static testing, dynamic testing, white box testing, black box testing, etc.
6. Use applications like .Net, SQL, Java, Oracle, VB Script, etc. for comprehending code written in respective coding language.
7. Monitor, access and validate automated alerts and test service requests.

7.1.1 Introduction

Software development and systems are a sector of the global economy that is increasing rapidly. Computers are inoperable without software. If this software and the hardware it works on are not maintained, entire civilizations could lose electricity and their basic needs for living.

7.1.2 Test Suite

Test suite is a container containing a collection of tests that aids testers in executing and reporting the status of test execution. It may exist in any of the three stages Active, In-Progress, or Completed.

Multiple test suites and test plans can include a test case. After generating a test plan, test suites including any number of tests are developed.

Test suites are generated in accordance with the cycle or the scope. It may include both functional and non-functional tests.

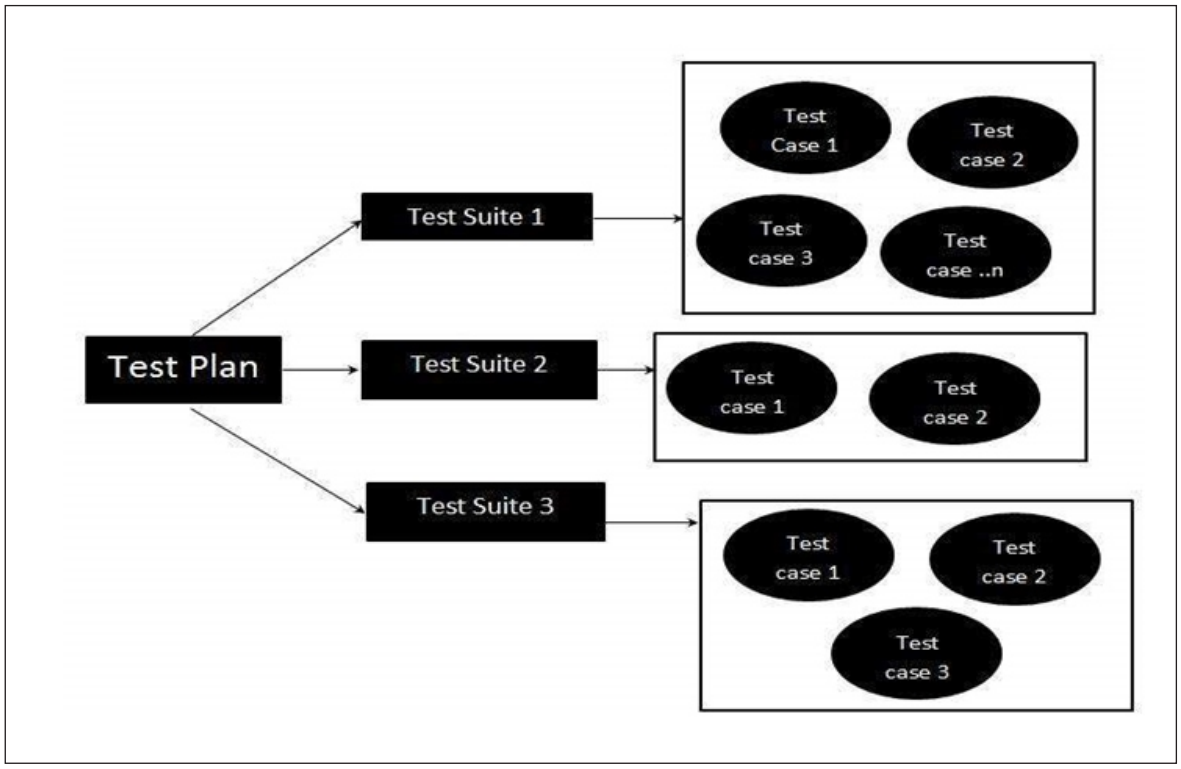


Fig. 7.1.1 Test Suites

7.1.3 Information Storage and Retrieval

A digital document is typically used when saving a document in a document management system (DMS). Instead of filing cabinets, we employ electronic storage today. It saves time and room. When a document is stored, it is usually marked with a set of metadata to designate a particular type or categorization. Then it is placed into the digital file.

The process of connecting a requester with the necessary material is known as document retrieval (s). A professional may help with the search and recovery of the record when dealing with specialised documents and management systems.

Information storage and retrieval is the process of systematically gathering and cataloguing data so that it can be located and shown on demand. Computers and data processing techniques have enabled the rapid, selective retrieval of vast quantities of data for government, commercial, and academic applications.

There are various types of information storage-and-retrieval systems.

- Document retrieval systems hold full documents, which are often retrieved using the document's title or associated keywords.
- In certain systems, document text is kept as data. This enables full text searching and retrieval based on any term in the document. In other cases, a digital image of the document is saved, typically on a write-once optical disc. Database systems store information as a sequence of discrete records divided into discrete fields (e.g., name, address, and telephone number); records can be searched and retrieved based on the content of the fields (e.g., all people who have a particular telephone area code). The information is saved on the computer, either in primary storage or secondary storage, for easy access.
- Reference retrieval systems retain references to documents as opposed to the actual documents. In response to a search query, these systems provide the document names and, frequently, their actual locations.

7.1.4 Backup and Restore

Periodic backups should be carried out when storing electronic documents, including electronic archives. The same guidelines for accessing archived electronic records should be applied to the backup(s), even though these backups do not qualify as archived records and may need to be restored in the event of a system failure. The long-term readability of these archives is typically not a problem because backups are typically intended for short-term storage rather than long-term storage or archiving; however, the restoration of the backups should also be checked on a regular basis. The facility's backup and restore policy should also cover any data created during the study's experimental phase.

Data extraction is the process of obtaining data from different data sources for later processing and analysis to derive insightful business information or for storage in a central data warehouse. Unstructured, semi-structured, or structured data can be gathered from various sources.

Data is frequently extracted by organizations, people, or companies for use in BI tools for analysis, migration to a repository, or replication as a backup.

Understanding Data Extraction and Generation

- The first step in the ETL (Extract, Transform, and Load) process is data extraction. You can only transform and load the data into the destinations you want to use for further data analysis after properly extracting the data.
- Data extraction, to put it simply, is the process of obtaining data from a source system in order to use it in a data warehouse environment. The Data Extraction process can typically be divided into three phases:
    - I**dentify Changes: You must monitor any updates to your data.** A new table or column might be included, for instance.

- **Specify the Data to be Extracted**: You should decide which portions of your data need to be extracted and then specify those portions. The entire set of data is extracted all at once using the full extraction method.
- **Process Data Extraction:** At this point, you have finished all the prerequisite steps and are prepared to extract data using either manually created scripts or automated tools.

7.1.5 Methods of Data Extraction

Data is gathered from various data sources during the data extraction process for later analysis by various data extraction tools.

The data sources can be physical or digital, including printed or physical media like books, newspapers, invoices, spreadsheets, and databases. For the analysis, many data extraction tools use some or all of the sources.

The process of extracting data from physical sources is typically laborious, expensive, and time-consuming when done manually, but modern technologies like optical character recognition (OCR) have made it possible to automate this process.

7.1.6 Types of Data Structures used in Data Extraction

The Data Extraction methods can be mainly divided into Logical and Physical. Additionally, there are numerous types of these, as follows:

1. Logical Data Extraction

The most popular data extraction technique is logical extraction. It is further separated into two groups:

- **Full Extraction:** This process typically occurs during initial load. Here, all of the data is directly extracted from the source at once. Since this extraction reflects all of the data that is currently available on the source system, there is no need to keep track of changes to the data source after the most recent successful extraction.
- **Incremental Extraction:** The delta changes in the data are the focus of this method. You must first apply complex extraction logic to the source systems as a data engineer and keep track of data updates and changes. The updated data extraction timestamps are recorded using this technique.

2. Physical Data Extraction

Data extraction using logical extraction from outdated data storage systems may be challenging. This information can only be obtained through physical extractions. It can also be divided into two categories:

- **Online Extraction:** You can extract data from the data source directly into the desired Data Warehouse using this process. The extraction tools must link directly to the source system for this method to function. Instead of connecting it directly to the source, you can connect it to the transitional system, which is an exact duplicate of the source system but with more structured data.
- **Offline Extraction:** Instead of being directly taken from the original source, the data is explicitly staged outside of it in this method. In this process, the data is either structured or can be structured using routines for data extraction. A flat file, a dump file, or a remote extraction from database transaction logs are a few of the file structures it takes into account.

Time is money for any organization. Therefore, you should think about using Data Extraction Tools that can improve your workflows and save you time. When used properly, data extraction tools can save your team time and free up staff members to work on higher priority tasks.

7.1.7 Generating Data

Here you will learn abut how to generate, manage, and use test data to achieve the best results.

- **Manual Test Data Generation**

  Manual test data generation involves producing test data independently, with the assistance of your QA team, or by developers. With the manual approach, you must first prepare a list of the things to test and then produce sample data for each of those things.

  It is the simplest and most direct method of gathering test data. The manual test data is frequently created at the start of a project's implementation and is done so in a way that includes all potential input-output combinations. This method is the most time-consuming and is typically applied to edge test cases and input/output combinations that are project-specific and cannot be automatically generated.

- **Automated Test Data Generation**

  Automated test data generation is done with the help of software tools that automate the whole process from start to finish. The main benefits of this method are vast generation speed, as well as the accuracy of generated data. It is an effective way to reduce the time taken to develop, maintain, and execute tests compared to using manual test data, which is prone to human error.

- **Managing the test data to achieve the best test results**

  The first step is only to create the sample test data. Managing the test data gathered from running each test case in each regression cycle from regression to release is the real challenge.

  Effective software testing is the result of efficient test data management, but it's not simple. Only about 45% of testing teams in a recent study said it was difficult to manage test environments and data.

The best test data management tools include:

- Solix Enterprise Data Management Suite
- DATPROF
- CA Test Data Manager (Datamaker)
- Compuware
- Informatica
- Microfocus Data Express
- IBM Test Data Management
- Appsurify
- PractiTest
- Quality

7.1.8 What is an Anomaly?

An anomaly in software testing is a result that differs from what was anticipated. This behavior may be the result of a document or the tester's perceptions and experiences.

As the test ware may behave in accordance with the specification but still have usability room for improvement, an anomaly can also refer to a usability issue. A defect or bug may also be used to describe the anomaly.

What is an Anomaly Report?

Anomaly Report: An anomaly report, also known as a bug report or defect report, is used to inform the project team or development team about anomalies that have been found and typically includes the following components:

- - Defect Identifier: A unique number assigned to each discovered defect.
- Describes the defects in detail in the defect description.
- **Defect Status:** The condition of the defect right now.
- Steps to duplicate the errors
- Severity.
- Priority.
- Date of Defect Reporting.
- Testing Conditions.
- Documents, screen captures, and other attachments

Data flow anomalies: An anomaly is denoted by a two-character sequence of actions. For example, ku means that the object is killed and then used, where as dd means that the object is defined twice without an intervening usage. What is an anomaly is depend on the application. There are nine possible two-letter combinations for d, k and u. some are bugs, some are suspicious, and some are okay.

1. DD: Probably not harmful, but unsettling. Why define the object twice when there isn't a usage in between? In the absence of a usage, why define the object twice?
2. DK : ?Why define an object if you're not going to use it?
3. DU : The normal case. The object is defined and then used.
4. KD: Normal situation. An object is killed and then redefined.
5. KK: Harmless but probably buggy. Did you want to be sure it was really killed?
6. KU: A bug. the object doesnot exist.
7. UD: Usually not a bug because the language permits reassignment at almost any time.
8. UK: Normal situation.
9. UU: Normal situation

7.1.9 Anomaly Detection Methods

Types of Anomalies are:

- **Point anomalies:** A single instance of knowledge is abnormal if it is too distant from the remainder.
- **Business use case:** Detecting MasterCard fraud based on "amount spent.
- **Contextual anomalies**: The abnormality is based on the specific context. This kind of anomaly can be found in time series data. company use case: Spending $100 on gas every day during the workdays is typical, but doing so while on vacation will be considered strange.
- **Collective anomalies:** A set of knowledge instances put together helps in detecting anomalies. Ex: Somebody is attempting to repeat information type a far off machine to an area host unexpectedly, An anomaly that will be marked as a possible cyber attack.

7.1.10 Use of Common Applications Like .Net, SQL, Java, Oracle, V B Script in Testing Are

1. NET?

Microsoft. NET, formerly known as. NET Core, was introduced in 2016 and is an open-source, cross-platform replacement for the earlier. NET Framework.

A wide range of applications, including desktop, mobile, game, web, and IoT devices, can be created using the. NET framework. .NET can use a variety of libraries, languages, and editors because it is open-source.

What Does NET Framework Do?

The NET Framework initially offered a standardized software development framework to aid Microsoft Windows development when it was first conceived and released. The. NET product now offers a multichannel open-source development framework. The .NET Framework, created by the. NET Foundation and made available under the MIT License, is designed to make developing simpler, quicker, and more reliable.

A vast ecosystem is associated with. NET, including:

- WPF (Windows Presentation Foundation). a user-interface design tool for the Windows operating system and desktop programs.
- Windows Forms. The Windows desktop applications can use the NET Framework GUI Library.
- ASP NET Forms. a web application framework that aids in creating web applications that are both secure and accessible.

What Is NET Used For?

- NET has several important parts that make development easier, including:
- Frameworks and libraries used in applications.
- Language compilers and runtime components.
- support for languages like Visual Basic, F#, and C#.
- software programs like Visual Studio
- Desktop applications, web applications, mobile apps , cloud-native applications, games, and Internet of Things (IoT) gadgets can all be made with NET.
- The incredible breadth of NET makes it a popular choice across a wide range of industries. Programmers who specialize in NET will be in high demand.
- NET is only a framework, though, at the same time. A developer must be familiar with one of the NET specific programming languages, such as C#, F#, or Visual Basic, in order to create an end product.

2. SQL (Structured Query Language)

- Both manual testers and automated testers must possess SQL knowledge.
- A domain-specific language called SQL (Structured Query Language) is used to store, manipulate, and retrieve data in databases.
- The SEQUEL (Structured English Query Language) was created by IBM researchers Raymond Boyce and Donald Cham berlin as a result of their study of E F Cod's paper. In 1970, they both created the SQL at IBM Corporation's San Jose Research facility.
- The preferred language for Relational Database Systems is SQL. SQL is the common database language used by all Relational Database Management Systems (R D MS), including My SQL, MS Access, Oracle, Sybase, Informix, Postgress, and SQL Server.

3. Uses of SQL

- **Data definition:** It describes the organization and structure of the data that is stored, as well as the connections between the data items that are stored.
- **Data retrieval:** Data retrieval can also be done using SQL.
- **Data manipulation:** SQL also offers the ability for users to add new data, remove data, or modify existing data.
- **Access control:** By limiting a user's ability to access, add, and modify data, SQL can safeguard stored information from unauthorized access.
- **Data sharing:** SQL is used to manage data sharing between concurrent users, preventing accidental erasure of changes made at nearly the same time by two different users.

4. Java

- Java is the most popular, widely used object-oriented programming language. The security feature of Java makes it popular and widely used. It used by many Java enthusiasts for different purposes. By using Java, one can develop a variety of applications such as enterprise applications, network applications, desktop applications, web applications, games, android app, and many more.
- The most widely used and well-known object-oriented programming language is Java. Java's popularity and widespread use are largely due to its security feature. Many Java enthusiasts use it for various purposes. Java can be used to create a wide range of applications, including enterprise apps, network apps, desktop apps, web apps, games, Android apps, and many more.
- Java Programming Language Applications
- The statement 3 Billion Devices Run Java, which is displayed during Java installation, attests to the Java programming language's extensive use. Java offers a comprehensive and rich set of APIs that aid developers in creating applications. Java allows us to create a variety of applications for various uses. We can create the following applications using Java technology.

- Development of mobile apps
- Applications for Desktop GUI
- Web-based Applications for the Web
- Applications for games
- Technologies for Big Data
- Various Applications
- Applications utilizing the cloud
- Applications of IoT
- Building mobile applications

5. Oracle Cloud Infrastructure (OCI) application

The globally distributed Oracle Cloud can be built, managed, and automated by developers using the Oracle Cloud Infrastructure (OCI) application development services. Customers can upgrade application servers like Oracle Web Logic, modernize existing apps using Kuber netes and containers, and create new applications using micro services, AI, autonomous databases, and many other technologies.

The advantages of Oracle Database

The main benefits of an Oracle database are as follows:

- **Oracle's Procedures and Guiding Principles** help us achieve high levels of database performance. Using performance optimization techniques in its database, we can speed up operations and query execution. This method facilitates quicker data retrieval and modification.
- **Being more Portable than any of Its Rivals,** the Oracle database can be used on a variety of platforms. This database can be used on more than 100 hardware platforms and about 20 networking protocols. By making secure changes to the OS and hardware, this database makes it simple to write an Oracle application.
- **Backup and Restoration:** It is always advisable to properly backup all of your Oracle online backup and restoration data. Utilizing the Oracle database makes it simple to complete recovery quickly. RMAN (Recovery Manager) functionality. During downtime or power outages, it can recover or restore database files. It can be used for archived backups, continuous archiving, and online backups. In order to perform user-managed recovery, we can also use SQL* PLUS.
- **PL/SQL:** Support for the PL/SQL extension for procedural programming is one of the main advantages of using the Oracle database.
- **Multiple Database:** Oracle database allows several database instances management on a single server. For controlling CPU allocations on a server hosting database instances, it offers an instance caging strategy. For controlling CPU allocations on a server hosting database instances, it offers an instance caging strategy. The database resource management and instance caging can work together to manage services across multiple instances.

Flashback Technology: The most recent Oracle version offers this benefit. It enables us to retrieve information that has been accidentally deleted or lost due to human error, such as dropping a table or accidentally deleting important information.

V B Script

- V B Script is a scripting language.
- uages Programming languages that are interpreted include scripting languages.
- Microsoft's Visual Basic programming language serves as a foundation for V B Script.
- It provides a potent tool that can be used to add interaction to web pages even though it doesn't offer the same functionality as Visual Basic.

V B Script uses

- V B Script is used to give web pages functionality and interaction.
- Client-side scripting can be done using V B Script.
- The drawback of using V B S for client side scripting is that only Internet Explorer can understand it. Microsoft's Internet Explorer is necessary for V B Script, a Microsoft product.
- Additionally, server-side scripting with V B S is an option. This calls for the use of an Active Server Pages (ASP)-compatible packaging and a Microsoft web server like Personal Web Server (PWS) or Internet Information Server (IIS).

7.1.11 Process of Conducting Static Testing, Dynamic Testing, White Box/black Box/Grey Box Testing

1. Static Testing

Static testing is a method of testing software that doesn't involve running the entire program or having the software executed. Static testing is done early on in the development process so that bugs and errors can be found and fixed before moving on to dynamic testing. Compared to dynamic testing, static testing might require less time and resources. The use of static testing may also uncover some errors that dynamic testing might not be able to detect as easily.

There are essentially two types of static testing:

- One is carried out manually, i.e., the analysis of the code is performed manually. A code review or a simple review are other names for this procedure.
- The automated analysis, in which we run the test using tools, is another.

Importance of Static Testing: There are a number of reasons why static testing is done and why it greatly enhances dynamic testing. The use and significance of static testing are illustrated in some of the points below.

- This testing method is used to find errors before they become serious, not just after dynamic testing.
- This testing is done to lessen the number of errors that were anticipated to be discovered during the dynamic testing, which will ease the workload in the next stages of development.
- The cost is optimised because of these cost and resource efficiencies, and the total development period is also expedited by using the static testing technique.

Being able to see mistakes with less effort and experience increases productivity.

Different components or entities that will be examined during the static testing procedure.

- Unit Test Examples.
- Documents Outlining Business Requirements
- A working version of the software.
- Test Data - User manuals, instruction manuals, or any other relevant documents
- Script for a performance test
- Testing Cases
- Requirements for Functions
- Documents with a traceability matrix
- Test strategy

The concept of testing review:

Finding potential mistakes or flaws in any program's design is a process known as review in static testing. The entire team is informed of the process's progress and every risk that might exist during the testing phase.

It is therefore conceivable to make a judgement that is significantly superior than any individual decision made by an expert because all the various mindsets may work together on the same platform and are free to express their opinions.

It should be mentioned that there are four main categories in which reviews can be placed. As follows:

- **Formal Review:** In a formal review, the entire procedure adheres to a formal and predetermined agenda. Additionally, it features a well-organized and regulated process that must be followed after each stage of the software life cycle.
- **Walkthrough:** During the walkthrough, the development team is required to review the software product and note any inconsistencies they encounter.
- **Technical Review:** Technical in nature, this means that only those who have received technical training are permitted to use their knowledge and expertise to identify errors and bugs in the software.

  As the name implies, technical, so only the technically trained persons are allowed to find errors and bugs within the software using their skills and experience.

Inspection: During this process, a team of experts and developers uses a set of standards, rules, and guidelines to evaluate the effectiveness of the software's features as well as to identify bugs and other flaws that can be fixed.

Participants in the testing's review phase:

- **Moderator:** This team manages the entry review, manages the rework, coaches the team, plans meetings as required, and completes all other pre-test review duties.
- **Scribe:** They perform the deficiency logging, attend the review meeting, and note all the necessary things that need to be fixed. The primary responsibility of a scribe is to take thorough notes while a test is being reviewed.
- **Reviewer:** This is the person or people who inspects and looks for flaws in the material.
- **Manager:** This person ensures that all reviewing protocols have been followed and that all requirements and objectives have been met after the test review has been completed.
- **Author:** This person is in charge of correcting any errors discovered and also has to guarantee that the document's quality has increased.

2. Dynamic Testing

- One type of software testing called dynamic testing completely examines the dynamic behavior of the software's code. In a nutshell, verification and validation make up the entire test, with the latter—the validation process—being referred to as dynamic testing.
- Let's look at an example of a log-in page, where there are two fields for entering the password and email. Let us say that we just have to add the local part, such as star and the address domain @gmail.com will be automatically added.
- Additionally, for the password to be accepted as a password, it must contain at least one symbol or an uppercase letter when registering. Dynamic testing is used to examine this class of dynamic functionality.
- Dynamic testing's primary goal is to test every component of the software by manually entering values into its UI or APIs and determining whether the values are accepted and whether the desired output is produced in accordance with the input.
- The system is tested with input from end users, and this is entirely their perspective.
- When working with the real system, dynamic testing can be carried out, allowing for real-time input and output validation. Simply put, this testing is carried out on the system itself to identify any flaws or errors while the software and the system are in use.
- Dynamic testing, as we have learned, is the process of verifying all the applications and features of the software by ensuring that real inputs are made and the software is maintained in a real working environment.

Dynamic Testing Process

- Dynamic testing typically follows a predetermined procedure once the approach and test implementation performances have been decided, at which point the team can proceed to carry out the various testing activities.
- The team can identify any irregularities in the approaches and strategies by using this process, which also aids in the display of all the testing steps.
- There are various tasks involved in the dynamic testing process in the STLC. Additionally, the success of the earlier task in the testing process is necessary for all of the functions in the dynamic testing process.

The steps that make up the dynamic testing process are as follows:

- Test case design
- Test environment step-up
- Test case execution
- Test analysis and evaluation
- Bug Reporting

In the life cycle of software testing, the actual dynamic testing process starts with test case design. Now that we have a thorough understanding of the dynamic testing procedure, we will go over each step individually.

Step1: Test Case Design

- In the life cycle of software testing, the actual dynamic testing process starts with test case design. Now that we have a thorough understanding of the dynamic testing procedure, we will go over each step individually.
- In this step, we can determine the features that require testing, as well as create the test conditions, acquire the test cases, and extract the coverage items.

Step2: Environment Setup

- Because testing is carried out directly on the software product, we will ensure that the test environment is always parallel to the production environment during this phase.
- Installing the test environment, which aids us in succeeding in the test machines, is the main goal of the dynamic testing process in this step.

Step3: Test Execution

- The test cases created during the initial phase of dynamic testing will be executed after the test environment has been successfully installed.

Step4: Analysis & Evaluation

- Following the completion of the test cases, we will analyze and assess the results. And we'll contrast those results with what was anticipated.
- We will consider those test cases to have failed and will log the bug in the bug repository if the expected and actual results are not the same after execution.
- Following the completion of the test cases, we will analyze and assess the results. And we'll contrast those results with what was anticipated.
- We will consider those test cases to have failed and will log the bug in the bug repository if the expected and actual results are not the same after execution.

Step5: Bug Reporting

- After examining the test cases, we will notify the appropriate party of any bugs or discrepancies between the actual result and the anticipated result. Additionally, the concerned party will ensure that the problem has been resolved and that a high-quality product is delivered.
- a case study of dynamic testing
- Let's use a simple example to demonstrate how dynamic testing functions.
- We will therefore comprehend the login module of any application, such as www.Twitter.com, for this purpose.
- Let's say we want to create a brand-new account with a strong password, in which case the password field must adhere to some pre-established guidelines.
- Also required are eight characters, all capital letters, and at least one special character in the password.
- If we are testing this functionality, we would take all the input conditions to test this and then verify the output.If we were to test this functionality, we would use all the possible input scenarios and then confirm the results.
- We can also impose restrictions that won't work, like requiring users to enter 4-character passwords, and check to see if an error actually occurred.

3. White Box Testing

- White box testing is a software testing technique where, the internal structure, design, coding and all the events are tested. As implied by the name, white box software allows the tester to see whatever is inside the box. As the name says, white box, which means whatever in the box that is, the software is visible to the tester.
- This method of testing focuses on confirming how the input flows into the output. The analysis of the data flow through the various modules or functions also focuses on enhancing the software's security, usability, and furnishing.
- There are several names for white box testing. For instance, open box testing or clear box testing. This means that the tester can see or access all of the internal modules. A team of developers performs white box testing.

Common Steps of White Box Testing

- Design all test scenarios, test cases and prioritize them according to high priority number.Create all test cases and scenarios, then rank them in order of highest priority.
- This step entails analyzing code while it is running to look at resource usage, code that isn't being used, how long different methods and operations take, and other factors.
- Internal subroutines are tested during this step. Internal subroutines, such as nonpublic methods and interfaces, can handle all types of data correctly or incorrectly depending on their design.
- This step focuses on testing conditional and loop statements to ensure that they are accurate and efficient for a range of data inputs.
- The final stage of white box testing includes security testing to look for any potential security flaws by examining how the code manages security.

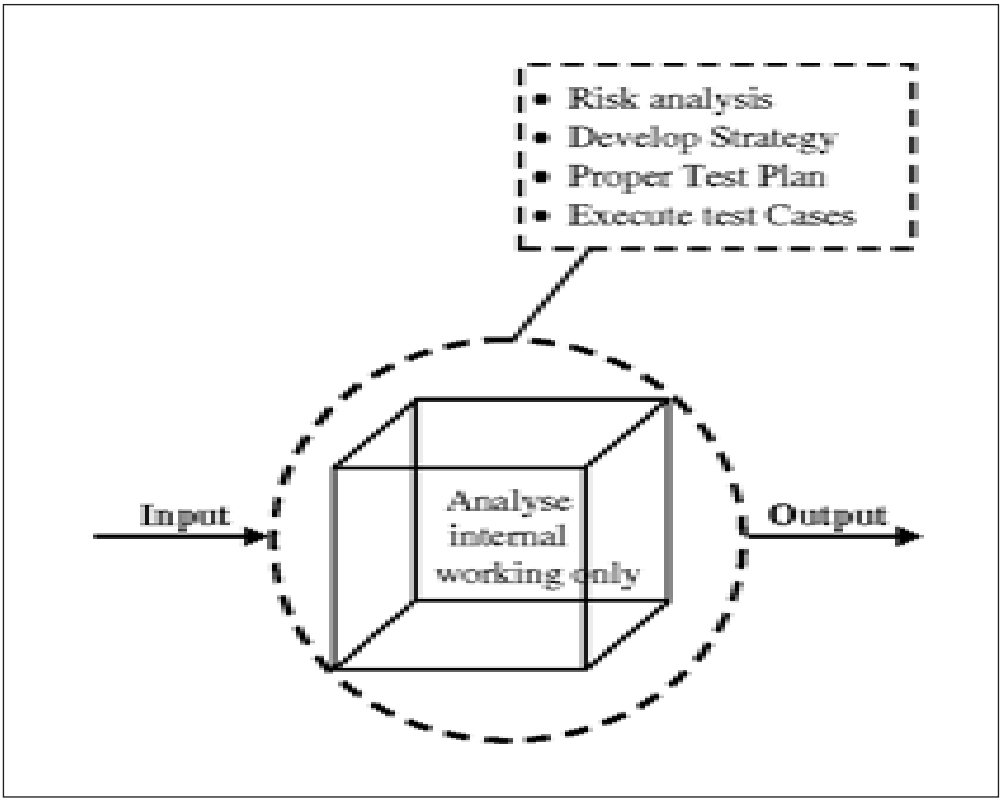


Fig 7.1.2 White Box Testing

Techniques for White Box Testing

Data Flow Testing	In order to explore the sequence of variables in relation to the sequence of events, a group of testing techniques known as data flow testing looks at the control flow of programs.
Control Flow Testing	Control flow testing establishes the program's statement or instruction execution order using a control structure. A test case for a program is created using the control structure of the program. In this method, the tester chooses a specific section of a sizable program to set the testing path. The program's control graph serves as a representation of test cases.
Branch Testing	All branches of the control flow graph are covered using the branch coverage technique. Each condition of the decision point is covered at least once for both true and false outcomes.
Statement Testing	White box test cases are created using the statement coverage technique. This method calls for running every source code statement at least once. Out of all the statements present in the source code, it is used to calculate the total number of executed statements.
Decision Testing	This method reports whether the results of Boolean expressions are true or false. Every time a statement, such as a do while statement, an if statement, or a case statement, has the potential to have two or more outcomes, it is referred to as a decision point because there are only two possible outcomes: true or false.

Table 7.1.1 White Box Testing Techniques

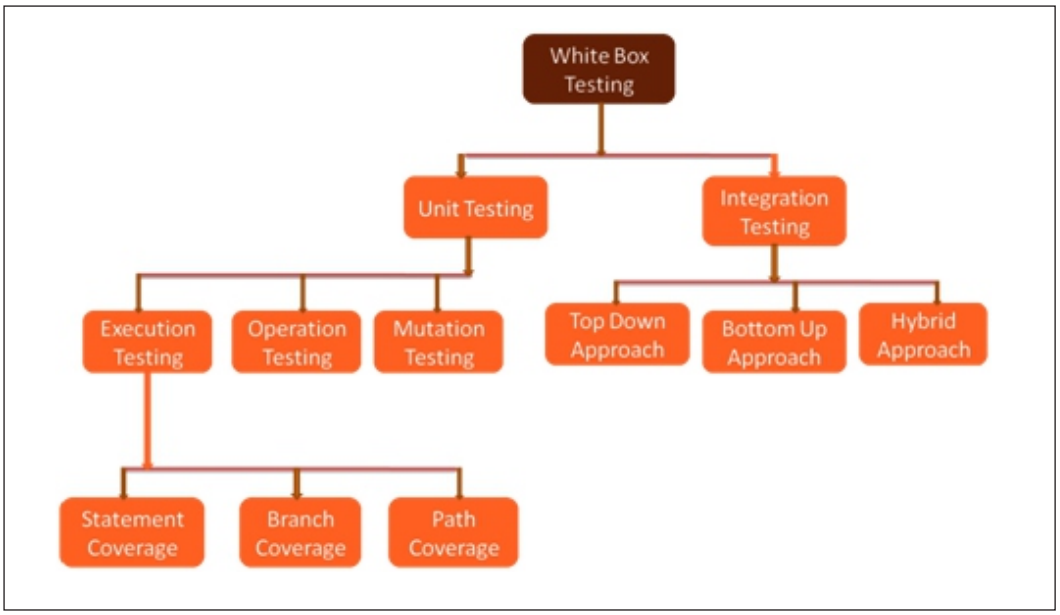


Fig 7.1.3 Types of White Box Testing

4. Black Box Testing

It is a method of testing in which the inner workings of the application are not known. It only looks at the system's fundamental components and has little to no bearing on the internal logical organization of the system.

Types of Black Box Testing

Black Box Testing comes in a variety of forms, but the following are the most common.

- Software testers conduct functional testing, a type of black box testing that relates to a system's functional requirements.
- Non-functional testing - This kind of black box testing focuses on non-functional requirements like performance, scalability, and usability rather than testing for specific functionality.
- Regression testing - Regression testing is carried out following code upgrades, fixes, or any other system maintenance to ensure that the new code has not adversely affected the existing code.

To make sure the code satisfies functional and nonfunctional requirements, black box testing involves testing external interfaces. The following are the various steps involved in black box testing:

Black Box Testing Steps

- Develop test plans. To prepare for black box testing, make prioritized test plans.
- Examine the external interfaces. Using automated test suites, such as NUnit suites and custom prototype applications, test the external interfaces for different types of inputs.
- Perform load testing. Run a load test on the application block to examine how it behaves under different loads. This guarantees that it satisfies all performance requirements that have been set forth.
- 4Apply stress testing. To analyze various bottlenecks and to spot any problems that would only be apparent under extremely high load conditions, such as race conditions and contentions, stress test the application block.
- Perform security testing. Test deployment scenarios for potential threats. Deploy the application block in a mock target environment and attempt to hack the application by taking advantage of any application block flaws that may exist.
- Conduct a globalization test. Execute test cases to ensure the application block can be integrated with programs intended for locales other than the one set as the default in the program.

5. Grey Box Testing Methodology

This technique, known as "White box + Black box = Grey box," is used to test an application while having only a basic understanding of how it operates internally.

The tester is not required to create test cases when performing gray box testing. Instead, algorithms that assess internal states, program behavior, and application architecture knowledge are used to create test cases. The tester then executes these tests and analyzes the outcomes.

The following actions are taken when conducting gray box testing:

Gray Box Testing Procedures

A structured evaluation based on the information currently known about the system being tested is known as a gray box assessment. It should proceed as follows:

- Determine inputs using black-box and white-box testing methods
- Determine the outputs that these inputs ought to produce in light of the documentation provided.
- Determine the key control flows that need to be tested.
- Determine which crucial sub-functions need to undergo in-depth testing.
- Determine a sub-function's inputs.
- Determine the outputs the sub-function should produce in response to the supplied inputs.
- Create and run a test case for this subsection of the function.
- Check to see if the sub-function delivers the desired outcome for the test case.
- For all sub-functions, repeat steps 4–8.

6. Comparison of "Black Box" testing, "White Box" testing, and "Grey Box" testing

Index	Black Box Testing	White Box Testing	Grey Box Testing
1	Knowledge of internal working structure (Code) is not required for this type of testing.For test cases, only the GUI (Graphical User Interface) is necessary. Only GUI (Graphical User Interface) is required for test cases.	This type of testing necessitates an understanding of the internal workings of software (software coding).	It is necessary to have some knowledge of how things work internally.
2	The terms functional testing, data-driven testing, and closed box testing are all variations of black box testing.	The terms structural testing, clear box testing, code-based testing, and transparent testing are also used to describe white box testing.	The term "grey box testing" or "translucent testing" refers to testing where the tester has only rudimentary coding knowledge.

3	Due to the fact that testers are not required to understand the internal software coding, the testing methodology includes trial and error methods.	Since there is no internal coding knowledge gap, White Box Testing is followed by a verification of the system boundaries and data domains built into the software.	The software is then validated for its internal system boundaries and data domains if the tester has coding knowledge.
4	The largest of all testing spaces is the testing space for input tables, which contain the inputs needed to build test cases.	In comparison to Black Box testing, the testing space for tables of inputs (inputs to be used for creating test cases) is smaller.	Tables for inputs (inputs used to build test cases) have a smaller testing space than Black Box and White Box testing.
5	Because hidden software errors may be caused by internal functioning that is unknown to Black Box testing, finding them can be very challenging.	Due to the possibility that they may result from internal functioning, which is thoroughly examined in White Box testing, hidden errors are easily found.	Difficult to discover the hidden error. User level testing may reveal it. Might be found in user level testing.
6	It is not taken into account when testing algorithms.	It is a good fit and is advised for testing algorithms.	It is not taken into account when testing algorithms.
7	The amount of time needed for Black Box testing depends on whether the functional specifications are available.	Due to the extensive code, designing test cases for white box testing takes a lot of time.	Designing test cases can be completed quickly.
8	Testers, developers, and end users can all participate.	The only people who can participate in testing are testers and developers; end users are not allowed.	Testers, developers, and end users can all participate.
9	Of all the testing processes, it takes the least amount of time.	The testing process as a whole takes the longest of all the testing processes.	Taking less time than White Box testing.
10	Black Box testing covers resilience and security against viral attacks.	White Box testing does not cover resilience and security against viral attacks.	.Grey Box testing does not cover resilience and security against viral attacks.
11	External expectations and unknown internal behavior form the basis of this testing.	Coding, which is in charge of internal operation, serves as the foundation for this testing.	High-level database and dataflow diagrams are used for testing.

Table 7.1.2 Comparison of "Black Box" testing, "White Box" testing, and "Grey Box" testing

Summary

- Information storage and retrieval is the process of systematically gathering and cataloguing data so that it can be located and shown on demand.
- Data extraction is the process of obtaining data from different data sources for later processing and analysis to derive insightful business information or for storage in a central data warehouse.
- The first step in the ETL (Extract, Transform, and Load) process is data extraction.
- The two main categories of data extraction techniques are logical and physical.
- Manual test data generation involves producing test data independently, with the assistance of your QA team, or by developers.
- A result that differs from what was anticipated is referred to as an anomaly. This behavior may be the result of a document or the tester's perceptions and experiences.
- NET can be used to build a wide variety of applications, including desktop apps, mobile apps, games, web apps, and IoT devices.
- The domain-specific language SQL (Structured Query Language) is used to store, manipulate, and retrieve data in databases.
- Static testing is a method of testing software that doesn't involve running the entire program or having the software executed.
- One type of software testing called dynamic testing completely examines the dynamic behavior of the software's code.
- The internal structure, design, coding, and all events are tested as part of the software testing technique known as "white box testing."
- It is a method of testing wherein the inner workings of the application are not known.
- White box + Black box = Grey box is a technique used to test an application while having only a cursory understanding of how it operates internally and a general understanding of the system.

Activitiy

Con-Vid Session

- In this session, the trainer will play a videos.
- The video will be learning about data extraction and its uses.
- The YouTube link for the video is https://www.youtube.com/watch?v=O6GFM_uGVFY
- The trainees will observe the video with pin drop silence.
- They can note down pointers from the video that they may find relevant.
- Trainees will maintain decorum in the class and not talk, whisper or discuss in the class.

In case of any queries or confusions, trainees will write those down in their notebooks.

Discussion Session

- The trainer will inquire whether the trainees comprehended the chapter.
- In this activity, the trainer will ask the trainee some questions related to the topic.
- The trainees will answer the questions according to their understanding of the topic.
- In case, the trainees have queries and confusions in their minds; they can put forth those in frontof the trainer.
- The trainer will ensure that he or she can give appropriate answers to the queries placed by the trainees.

Exercise

A. Answer the Following Questions

1. Differentiate between Black Box Testing vs. White Box Testing vs. Grey Box Testing.

2. Brief static testing & dynamic testing.

B. Fill in the Blanks

a) _____ is the 1st phase of the ETL (Extract, Transform, and Load) process.

b) _____ is a software testing technique in which the software is tested without the entire program or the software is being run or executed.

c) _____ refers to a result that is different from the expected one.

d) _____ is a domain-specific language used for storing, manipulating and retrieving data in databases.

Notes

Scan the QR Code to watch the related videos

https://www.youtube.com/watch?v=URmCWIL0IA0

Latest Changes, procedures and practice in the field of designing test suite

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

8. Carry Out Manual Tests on Software Products/ Applications/Modules

IT - ITeS SSC
NASSCOM

Unit 8.1 - Carry Out Manual Tests on Software Products/Applications/Modules

SSC/N1303

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Demonstrate understanding of the nature of testing to be carried out and usage of the test management tool to be used.
2. Evaluate the use of adhering conformance to usability guidelines in case of usability testing.

UNIT 8.1: Carry out Manual Tests on Software Products/ Applications/Modules

Unit Objectives

By the end of this unit, participants will be able to:

1. Select the latest versions of the test cases and automated scripts.
2. Select correct alternative solution from software specification document.
3. Identify the correct versions of the application and data to be used for testing.
4. Analyse requirements from the software specification document.
5. Demonstrate the process of creating a test plan and developing test cases.
6. Develop test progress report, results and defects discovered, using the agreed test management tool.
7. Demonstrate contingency plans to monitor risk triggers during the project.
8. Analyse results to identify the defects and track the same in defect tracking system.

8.1.1 Manual Testing

Software testing that uses manual execution of test cases rather than an automated tool is known as manual testing. All test cases are manually run from the perspective of the end user by the tester. The application's compliance with the requirements outlined in the requirement document is determined by this. To finish almost 100% of the software application, test cases are created and implemented.

One of the most fundamental testing techniques is manual testing, which has the ability to find both outward and internal software flaws. A fault is when the output produced by the program differs from what was intended. Before giving them back to the tester for another round of testing, the developer fixed the bugs.

Any newly created software or product must first undergo manual testing before being subjected to automated testing. Although it takes a lot of time and effort, this testing makes sure that the final product is bug-free. While automated testing software does not require any prior knowledge, manual testing does.

8.1.2 What is the Importance of Manual Testing?

Although there are still many reasons for manual testing, software professionals are favoring automated testing more and more. Few are:

Human Perspective: Humans can quickly assess the app's fundamental usefulness and appearance by giving it a quick glance. A tester may spot usability issues and user interface flaws when they interact with software in the same way that a user would. These issues cannot be found by automated test scripts.

A larger view of the System workflows: A more comprehensive view of the app is always provided by manual verification. Instead of being in a coding mode that repeats processes, the human mind is constantly exploring. So, it will cover the greater ground for system validation.

- **Human Perspective:** Humans can quickly assess the app's fundamental usefulness and appearance by giving it a quick glance. A tester may spot usability issues and user interface flaws when they interact with software in the same way that a user would. These issues cannot be found by automated test scripts.
- **A larger view of the System workflows:** A more comprehensive view of the app is always provided by manual verification. Instead of being in a coding mode that repeats processes, the human mind is constantly exploring. So, it will cover the greater ground for system validation.
- Automation costs money because it frequently results in false positives and false negatives during testing. By incorporating a human touch throughout the testing process, these errors are avoided.

Scenarios that aren't automatable or don't offer obvious confidence in user behavior when testing with automation, for example, "Tap & Pay" on mobile devices, have distinct behaviors when automated using tools vs. manually verified. Despite this, manual testing still holds a prominent place in the quick-moving validation stage of the software development life cycle. Additionally, there are instances where manual verification is the best choice.

Automation is expensive, often Testing that is automated might produce false positives and false negatives. By incorporating a human touch throughout the testing process, these errors are avoided.

8.1.3 Platform and Application Versions

The task of the management team team is to select the target platform and compatible applications because the operating system, Web browser, compatible versions, and other features are designed based on customer perspective. The platform and application versions enable the development and testing teams to decide on what is to be done.

Example: On most of the software packages or start up screens you may come across the following:

- Works best with
- Requires Windows XP or greater
- For use with Unix or Linux 2.6.10 only

Compatibility testing involves checking the compatibility of an application or website with several browsers, operating systems, and hardware. This testing is conducted on an existing environment either manually or on an automated basis.

1. Testing Multiple Versions

- To test various versions of platforms and software applications is a challenging task. We will now consider a situation where a compatibility test is to be done on a popular operating system. The programmers have fixed several bugs and have also improved performances by adding new features to the existing code. There are thousands of existing programs for the present version of the operating system. The ultimate aim of the project is to ensure 100% compatibility. Equivalence partitioning is appropriately applied to reduce the job of testers.
- The task of compatibility testing starts with the equivalence partitioning of all possible combinations of the software. This is done to ensure that the equivalence sets verify the accuracy of the interaction between the software. Although one can test all the possible software programs on the operating system, only the most important ones are finalized and tested.
- Baseline versions, intermediate versions, and revisions are thought to be the three primary types of software configuration releases.

2. Baseline Versions

Early in a system's development or operation, baseline software configuration versions are planned. They undergo review, testing, and approval as part of the procedure, along with their SCIs. Baseline versions represent the starting point for further system development and act as milestones in the life cycle of the software system.

3. Intermediate Versions

When problems arise that require immediate attention – such as the need to correct defects identified in an important SCI, or perform immediate adaptations as defined in a contract with a new customer – an intermediate version of the software is often prepared. Ordinarily, intermediate versions are only used by a portion of a company's clients and only for a short time before being replaced by a new baseline version. Ordinarily, intermediate versions are only used by a portion of a company's clients and only for a short time before being replaced by a new baseline version. Naturally, we can anticipate that these versions won't receive the focus and effort typically put into the publication of baseline versions. An intermediate software configuration version can thus serve as a “pilot” or springboard to the next baseline version.

An interim version of the software is frequently created when issues arise that demand immediate attention, such as the need to fix errors found in a crucial SCI or carry out quick adaptations as required by a contract with a new client. Usually, intermediate versions are only used by a small percentage of a company's customers for a brief period of time before being replaced by a new baseline version. Naturally, we can anticipate that these versions won't receive the focus and effort typically put into the publication of baseline versions. Thus, an intermediate software configuration version may act as a "pilot" or launch pad for the following baseline version.

8.1.4 Numeration Conventions for Identification of SCI and Software Versions

The most widely used numeration convention for identifying SCIs is decimal numeration, which indicates the escalating version and revision numbers and is registered accordingly. DD-7 Ver.1.0, DD-7 Ver.1.1, DD-7 Ver.2.0, DD-7 Ver.3.0, DD-7 Ver.3.1, DD-7 Ver.3.2, etc., where the first number denotes the version and the second the revision, are examples of SCI design documents that may have multiple versions and revisions.

Use of Version Control System:

- **Repository:** It can be compared to a database of modifications. It includes all revisions and previous iterations (snapshots) of the project.
- **Copy of Work (sometimes called as checkout):** It is the individual copy of each project's files. When finished, you can commit your changes to a repository and continue editing this copy without affecting the work of others.
- **Working in a group:** Imagine that you are employed by a company and are required to work on a current project. Since the main code is already in use and cannot be changed without causing the user any inconvenience, you must work with your team as a whole to collaborate and incorporate their changes. Utilizing version control will assist you in merging various requests into the main repository without making any unintended changes. You don't have to download and set up every time to test the functionality; you can simply pull the changes, make the changes, test them, and then merge them back. It might be pictured as.

Imagine that you are employed by a company and are required to work on a current project. Since the main code is already in use and cannot be changed without causing the user any inconvenience, you must work with your team as a whole to collaborate and incorporate their changes. You can merge various requests into the main repository with the assistance of version control without making any unintended changes. You don't have to download and set up every time to test the functionality; you can simply pull the changes, make the changes, test them, and then merge them back. It may be visualized as.

8.1.5 Version History

Every component in your Enterprise Tester repository comes with a version history. Versioning is a feature of Enterprise Tester that comes standard and, like Baselines, is not dependent on a Version Control Repository. An initial version is produced whenever a new element, such as a requirement or test script, is added.

A new incremental version of an element is automatically created as you make changes. By selecting the Versions tab when viewing your requirement or test script, you can view all versions and contrast the components of a chosen version with the current version. You can quickly reinstate an earlier version at any time by choosing to go back to a previous version of an element at any time. A new version is created as the most recent version when you roll back to an earlier version, and all previous versions—including those made after the version you're rolling back to—are kept as well. This gives you a very strong mechanism for regulating repository changes.

8.1.6 Writing Manual Test Case

Test cases help guide the tester through a sequence of steps to validate whether a software application is free of bugs, and working as required by the end-user. The ability to write clearly, pay attention to detail, and have a solid grasp of the application under test (AUT) are all necessary when learning how to write test cases for software. Learning how to write test cases for software requires basic writing skills, an attention to detail, and a good understanding of the application under test (AUT).

A test suite is typically made up of the test cases for a particular module or section of an application. There are typically multiple specific scenarios to be tested, so a test session will typically include many test cases.

Any tester should be able to comprehend and carry out the test using a well-written test case.

It's crucial to consider the user's perspective when writing test cases and to include all necessary information. You will save time and effort in the long run by making the extra effort to write quality test cases up front. A well-tested application can be distinguished from one that has not been thoroughly tested by having well-written test cases.

Writing test cases – especially high volumes of them at once – can be a time-consuming task. Writing and maintaining organized, high-quality test cases is simple with TestLodge. Writing effective test cases and keeping them organized are simple with TestLodge. Here, we've provided some guidelines for writing test cases. At the end of this article, we've also included a sample test case.

8.1.7 How to Write Test Cases for Software

- **Use a Strong Title**

  Strong titles are the foundation of a good test case. As a best practice, it's advisable to name the test case after the module you are testing. For instance, if you're testing the login page, the title of the test case should say "Login Page." If the tool you're using doesn't already do so, it might make sense in some circumstances to include a distinctive identifier in the test case's title as well so that it can be referenced rather than a lengthy title.

- **Include a Strong Description**

  The tester should be made aware of what they will test in the description. Other relevant information, such as the test environment, test data, and preconditions and assumptions, may occasionally be included in this section. A description ought to be simple to read and convey the test's main objective right away.

- **Include Assumptions and Preconditions**

  Include any preconditions that must be met before the test can be executed, as well as any assumptions that apply to the test. This information may include the page on which the user should launch the test, environment dependencies, and any unique setup requirements that must be fulfilled before launching the test. These details aid in keeping the test procedures succinct and to the point.

- **Keep the Test Steps Clear and Concise**

  Simple test cases are ideal. Remember that not necessarily the person who writes the test case will also run the test. The necessary data and instructions on how to conduct the test should be included in the test steps. This is possibly the most significant component of a test case. Do not omit any crucial information, but keep this section succinct and clear. Create the test case in a way that allows anyone to execute the test.

- **Include the Expected result**

  The expected outcome describes what the tester should encounter as a result of the test procedures. The tester uses this method to decide whether the test case is a "pass" or "fail".

- **Make it Reusable**

  A good test case is reusable and offers the software testing team long-term value. Consider this when drafting a test case. Reusing the test case as opposed to writing it from scratch will save you time in the future.

- **Sample Test Case**

Here is an example of a test case:

- Title: Login Page – Authenticate Successfully on gmail.com
- Description: A registered user should be able to successfully log in at gmail.com.
- Precondition: the user must already be registered with an email address and password.
- Assumption: a supported browser is being used.

Test Steps:

- Navigate to gmail.com
- In the 'email' field, enter the email address of the registered user.
- Click the 'Next' button.
- Enter the password of the registered user
- Click 'Sign In'

8.1.8 Test Progress Report

Test Report is a document which contains a summary of all test activities and final test results of a testing project. A test report evaluates how well the testing is carried out. A test report evaluates how well the testing is carried out. Based on the test report, stakeholders can evaluate the quality of the tested product and make a decision on the software release.

For instance, if the test report reveals that the product still has a number of flaws, stakeholders may decide to hold off on the release until every flaw has been addressed.

The test report serves as a channel of communication between the stakeholder and the test manager. The stakeholder can comprehend the project situation, the product quality, and other things through the test report.

The scenario that follows demonstrates the need for a quality test report

A good test report should include the following:

- **Detail:** You should give a thorough description of the testing activity and list the tests you conducted. Avoid including abstract information in your report because the reader will find it confusing.
- **Clarity:** The test report should contain only concise, easily understood information.
- **Standard:** The test report should adhere to the prescribed format. In many projects, it is simple for stakeholders to review and confirm the consistency between test reports.
- **Specific:** Avoid composing an essay about the project activity. Focus on the main point while describing and summarizing the test result specification.

8.1.9 Types of Test Reports

Test planning, test specification, and test reporting are all covered by the documents outlined in the IEEE Standard of Software Test Documentation.

Test reporting covers four document types:

- If a formal start to test execution is desired, a Test Item Transmittal Report identifies the test items being sent from development to the testing group for testing.
    - Details to be included in the report - Purpose, Outline, Transmittal-Report Identifier, Transmitted Items, Location, Status, and Approvals.
- A Test Log is used by the test team to record what occurred during test execution
    - Details to be included in the report - Purpose, Outline, Test-Log Identifier, Description, Activity and Event Entries, Execution Description, Procedure Results, Environmental Information, Anomalous Events, Incident-Report Identifiers.

- **Details to be included in the report -** Purpose, Outline, Test-Incident-Report Identifier, Summary, Impact.

- A test summary report summarizes the testing activities associated with one or more test-design specifications.
  - **Details to be included in the report -** Purpose, Outline, Test-Summary-Report, Identifier, Summary, Variances, Comprehensiveness Assessment, Summary of Results, Summary of Activities, and Approvals.

8.1.10 Defect Resolution

Defect Resolution in software testing is a step by step process of fixing the defects. Defect resolution in software testing is the process of gradually fixing flaws. The first step in the defect resolution process is assigning the defect to the developer. After scheduling the defect fix according to priority, the defect is fixed, and then the developer sends a report of resolution to the test manager. This process helps to fix and track defects easily.

You can follow the following Steps to Fix the Defect

- **Assignment:** Changed the status to Responding and assigned the problem to a developer or other technician.
- **Setting the schedule:** This phase is handled by the developer side. Depending on the priority of the defects, they will develop a schedule to fix them.
- **Fix the defect:** The Test Manager monitors the process of fixing defects in comparison to the above schedule while the development team is fixing the defects.
- **Report the resolution:** When bugs are fixed, request a report of the resolution from the developers.
- **Verification:** The testing team confirms that the defects have been fixed after the development team fixed and reported them.
- **Closure:** A defect is changed to its closed status once it has been fixed and verified.

8.1.11 How to Measure and Evaluate the Quality of the Test Execution?

Test managers prepare and send defect reports to the management team during the defect reporting process for software testing in order to receive feedback on the defect management procedure and the status of the defects. The management team then reviews the defect report and sends feedback or additional assistance as needed. Defect reporting facilitates clearer communication, detailed tracking, and explanation of defects.

The management board has a right to be informed of the status of defects. To assist you with this project, they must be familiar with the defect management process. Therefore, in order to receive feedback from them regarding the current defect situation, you must inform them.

8.1.12 How to Measure and Evaluate the Quality of the Test Execution?

Every test manager is interested in knowing the answer to this question. There are the following 2 parameters you can take into consideration:

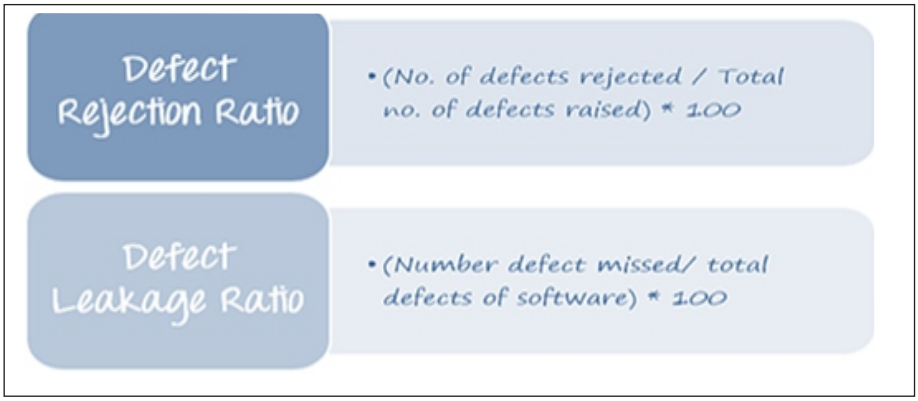


Fig. 8.1.1 Report Documentation of Defects

8.1.13 What is a Test Report?

A test report is a well-organized summary of the goals, procedures, and outcomes of the test. It was developed and is used to assist stakeholders (product manager, analysts, testing team, and developers) in determining whether a feature, product, or defect resolution is ready for release.

An important document called a test summary report is created at the conclusion of a testing project, or rather, after the testing cycle has been completed. The main goal of this document is to inform the relevant stakeholders, such as Senior Management, Clients, etc., in detail about the Testing activities carried out for the Project. Additionally, it shows how well-rounded the application's quality is.

What to include in a Test Summary Report?

A test summary report that is instructive should be succinct and pertinent. There are numerous test summary report template examples online, but not all of them may be applicable in your situation. Therefore, it is crucial to adjust our report after conducting a thorough analysis in accordance with the characteristics of our test project.

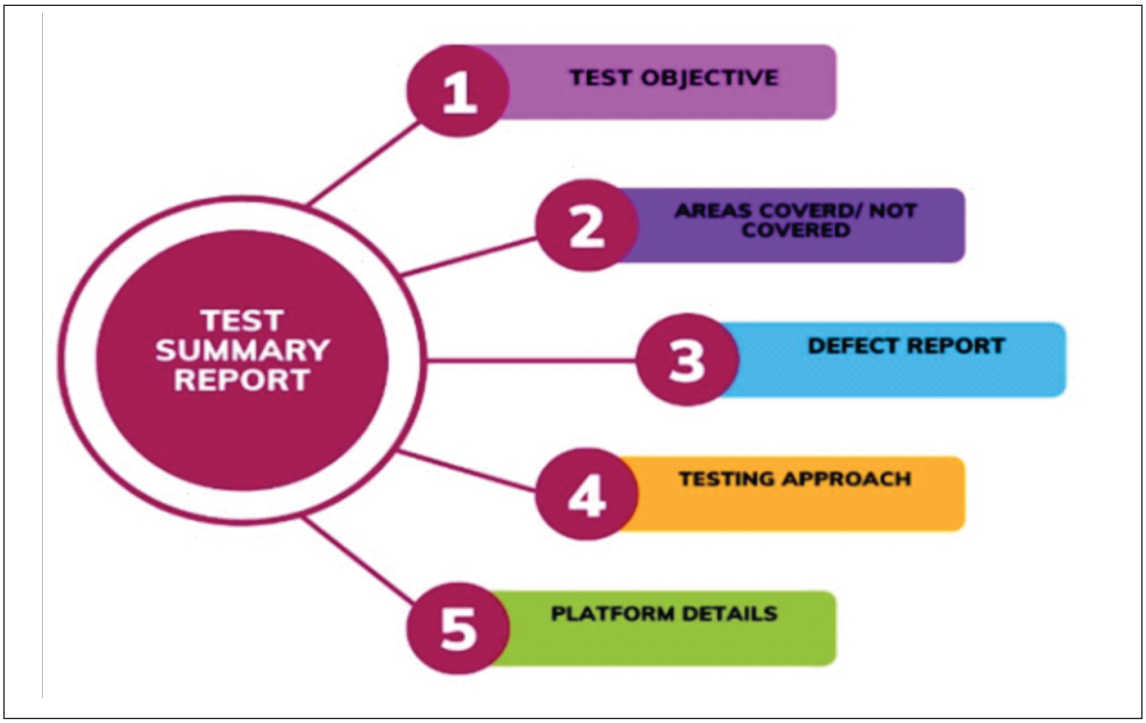


Fig. 8.1.2 Test Summary Report

1 **Test Objective** – Include the testing's objective to demonstrate that the team's understanding of the test object and requirement was complete.

2. **Areas Covered–** List the components of the product that were examined during testing. It need not include every test scenario in great detail, but it should cover all the major bases.

3. **Areas Not Covered–** It is very essential to capture the areas of the product that were not covered in testing. Any areas not tested can cause concern on the client's end, so we need to make sure that anything untested is noted and that expectations are set appropriately. Any areas not tested can cause concern on the client's end, so we need to make sure that anything untested is noted and that expectations are set appropriately. Also, ensure that each point has an associated reason, for example, limitation of access to the availability of devices.

4. **Testing Approach**– This section needs to be covered because it describes the nature and methods of the testing. Make sure to describe in detail the actions taken and the different testing strategies that were used to complete the task.

5. **Defect Report**– Although a defect report is typically included in a bug report, including one in the test summary report can give your report an edge.

6. **Platform specifics**– Products are currently tested on a variety of platforms. Due to the increased demand, testing now encompasses not only various hardware configurations and browsers but also various software releases. Let's make sure to include information about each platform and environment that the product was tested in.

8.1.14 Defect Tracking System

A software program that tracks reported software issues is known as a bug tracking system or defect tracking system. Based on the overall significance and likelihood that the bug will recur in the future, each bug in the system may be given a value for its urgency. Low-urgency bugs are minor and should be fixed as soon as time allows. Other information about bugs includes the client who reported it, the submission date, in-depth descriptions of the issue, tried fixes, and other pertinent data. The technician updates the system with fresh data as they fix that bug. Any effort to address the issue should be recorded in the bug system because each bug keeps a history of all changes.

The Important Elements of a Bug Tracking System are as Follows

The first and foremost thing we require is knowledge of the precise location of the program's bug. Location provides details such as the program's method, class, and line number where the bug is located. This makes it easier for developers to find problematic code. This information can be obtained by using software development environments (IDEs).

A database that keeps track of information about known bugs is a key part of a bug tracking system. The time a bug was reported, its severity, the incorrect program behavior, information on how to reproduce the bug, and the names of any programmers who may be working on fixing it are all examples of relevant facts. The user information, login information, data code information, and reports that are stored in a database will all be handled by the bug tracking system. These services are offered by our project using the JSP and HTML languages. A user interface for login, signup, etc. is provided by HTML. JSP, on the other hand, is used to connect the HTML to the server. To store the user information in this, we used a MySQL database. As your starting point for the day, our user-friendly interface includes a Dashboard view that displays issues that are "Assigned to Me," "Unassigned," "Reported by Me," and many other common filters.

Defect tracking is the process of identifying flaws in a product. It can take the form of discrete systems where information about the flaw and the progress of work to fix it is stored, or it can be a component of an integrated set of configuration management tools where the defect's status can act as a key or trigger for other system events.

8.1.15 Effective Defect Reporting

Effective defect reporting is crucial to prevent needless time and effort being spent trying to comprehend and reproduce the defect. Here are some recommendations:

Be specific:

Give more details, not fewer. In other words, avoid being sluggish. The information you give developers may or may not be used, but they are not going to come to you and ask for anything you missed.

Be objective:

Make no judgments about the quality of the application, such as "This is a lousy application" or "You fixed it real bad."

Keep to the facts and stay away from your feelings.

Reproduce the defect:

Do not report a defect right away if you find one out of haste. At the very least, repeat it once more to be certain. (If you are having trouble reproducing it, try remembering the precise test condition and keep trying. The report should be submitted for further investigation if you are finally unable to replicate the defect after numerous attempts while also including any evidence you may have gathered to support your claim.)

Review the report:

Do not immediately press "Submit" after finishing the report. Review it at least once. Remove any typos.

Example: Employee Login Page Defect Report:

- Defect id: D001
- Project name: MyASP
- Module Name: Login
- Sub Module Name: Employee Login
- Type of Defect: Missing
- Status: New
- Seriousness: High
- Priority :
- Failure to Open Employee Login Page
- Description :
- Enter Url
- Select "Employee Login" from the menu.
- The employee login page should load as expected.
- Actual Results: Employee login page does not get open
- Reported By: ABC Tester
- Assign To: XYZ Developer
- Date & Time: 09/11/2019

8.1.16 Impacts of Test Data on the Manual Test

Data that will be used to test a specific piece of software is known as test data. While some data is used to obtain results that are confirmed, other data may be used to question the software's capabilities. Appropriate test data can be acquired for system testing in a variety of ways. The test data for testing a specific system can be generated by a tester or a program.

The testing team might, for instance, want to check whether the software yields the desired outcome or not. The system would be fed the data, and it would run. After analyzing the outcome, it would decide if the desired outcomes had been attained. The software should at the very least operate without error and produce the desired results. It must accomplish this because that was the main reason for creating it in the first place.

What are the types of test data?

Below is the list of some common type of test data:

Boundary Test Data

This kind of information aids in the elimination of connected defects that arise during the processing of boundary values. A combination of boundary values that are enough to handle the application makes up the data in this data type. Additionally, if the tester goes further, the application might be broken.

Valid Test Data

These data types are legitimate and the application accepts them. These aid in confirming how the system works, and when an input is given, they aid in obtaining the desired result.

Invalid Test Data

Unsupported data formats are included in these data types. The teams use the data to assess whether or not the application is functioning properly. By entering invalid values, the app should display the appropriate error message and let the user know that the data is unfit for use.

Absent Data

Files without any data are referred to as blank files or no data files. By entering blank or no data into the software, the app's response can be tested to see how it reacts.

Summary

- Software testing that uses manual execution of test cases rather than an automated tool is known as manual testing.
- Any tester should be able to comprehend and carry out the test using a well-written test case.
- Test cases aid in directing the tester through a series of actions to verify whether a software application is free of bugs and performing in accordance with end-user expectations.
- In software testing, defect resolution is the process of correcting flaws step by step.
- Test managers prepare and send defect reports to the management team during the defect reporting process for software testing in order to receive feedback on the defect management procedure and the status of the defects.
- A test report is a well-organized summary of the goals, procedures, and outcomes of the test. It was developed and is employed to assist stakeholders (product manager, analysts, testing team, and developers) in understanding product quality and determining whether a product, feature, or defect resolution is on schedule for release.
- A database that keeps track of information about known bugs is a key part of a bug tracking system.
- It is essential that you report defects effectively so that time and effort is not unnecessarily wasted in trying to understand and reproduce the defect.

Activitiy

Chart Paper

- In this activity, the trainer will conduct a session on creating Test Report
- Each of the candidate will be provided with a common Test Report Template.
- Each of the candidate has to fill it for prerelease testing.
- It is important that the trainees present their answers not only rich in information but also supported by a flow chart.
- The group, which can present their answers in the best way within 30 minutes will be awarded appreciation and accolades.

Discussion Session

- The trainer will inquire whether the trainees comprehended the chapter.
- In this activity, the trainer will ask the trainee some questions related to the topic.
- The trainees will answer the questions according to their understanding of the topic.
- In case, the trainees have queries and confusions in their minds; they can put forth those in frontof the trainer.
- The trainer will ensure that he or she can give appropriate answers to the queries placed by the trainees.

Exercise

A. Answer the Following Questions:

1. What is a Test Report?
2. What should a test summary report contain?
3. Enumerate the Importance of Manual Testing?

B. Fill in the Blanks

Version , Defect, Manual testing ,Test

1. A _____ report is an organized summary of testing objectives, activities, and results.
2. Software testing that uses manual execution of test cases rather than an automated tool is known as _____.
3. _____ history is provided on all elements within your Enterprise Tester repository.
4. A software program that tracks reported software bugs is known as a bug _____ tracking system.

Notes

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

IT - ITeS SSC
NASSCOM

9. Manage Your Work to Meet Requirements

Unit 9.1 - Follow Instructions and Company Policies

Unit 9.2 - Work Ethics to Follow in an Organization

SSC/N9001

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Discuss the importance of following work instruction and complying with company policies.
2. Recognize proper work ethics to follow in an organization.

UNIT 9.1: Follow Instructions and Company Policies

Unit Objectives

By the end of this unit, participants will be able to:

1. Discuss and agree to work requirements with appropriate people.
2. Demonstrate how to obtain guidance from appropriate people, where necessary.
3. Identify the methods so that the work meets the agreed requirements.
4. Recognize how to use resources correctly and efficiently.

9.1.1 Work Accurately

Relevant data is important in every professional field. Finishing tasks accurately and on time is essential in the workplace. When employees finish their work correctly, and within the stipulated time, they directly influence the company's growth, profit and efficiency.

Establish and agree on work requirements with appropriate people

For a Test Engineers, it is crucial to meet the deadline. The various requirements to be met are:

- Activities (what is required to be done).
- Deliverables (output of work).
- Quantity (the amount of work to be done).
- Standards (acceptable performance coupled with compliance and service level agreement).
- Timing (the stipulated time for completion of task/s).

A Test Engineers must obtain valuable information from:

- **Subject Matter Expert (SME):** SME is the point of contact in case a Test Engineers faces an issue during operations
- **Floor Manager:** Manages a team and looks after the operations as well as HR-related issues of the employees under him/her. Associates get placed under a Floor Manager who, in most cases, becomes their reporting person.
- **Process Manager:** A Process Manager has more than one team under him/her supervision. The profile is more like an Operations Manager who has the entire ops team to look after. In an organization, there might be five teams under separate Floor Managers. However, these five teams will be looked after by the Process Manager.

- **Floor Supports:** Floor Supports are coordinators who mainly look after the smooth coordination between various teams. For example, a Floor Support must coordinate with on-floor teams to get a job done without hassle. Whereas SMEs are only the experts in the operations, Floor Support must be well-versed with various processes, including the core operations.
- **Team Lead (TL):** Team Lead is appointed under each Floor Manager to look after the operations. While the Floor Manager must take care of every issue of the employees under him/her, Team Lead looks after the operations. Team Leads are often empowered with authority to perform HR-related tasks, such as leave approval, grievance handling, etc. However, in most organizations, TLs are placed under the Floor Manager to help him/her look after the team.
- **Manager:** A manager is one rank higher than the Process Manager. Under one Manager, multiple teams, such as Ops Team, Caller Team, Chat Team, and Email Team, are allotted. The manager looks after all the aspects of an organization, including operations, HR and Sales.
- **Immediate Seniors:** Immediate Seniors are the seniors in the same team. A team is made of Junior Associates and Senior Associates to maintain a balance. Senior Associates can be SMEs as well (depending on the decision of an organization). It is advisable for Junior Associates to consult an SME or a Senior Associate first to mitigate an issue.

It is important for a Test Engineers to stick to the Service Level Agreements (SLAs)

- Service level agreement is a crucial element of the service contract.
- SLA is the time considered to finish a job as per the contract. For instance, internet service providers are likely to consider service level agreements where the time range of the net service will be considered.
- SLA consists of two basic components: Mean Time To Recovery (MTTR) and Mean Time Between Failures (MTBF).
- Simply, SLA is the negotiated agreement between parties where one group is the customers, and the other is the service provider.

Fig 9.1.1 Team Meetings or Open Discussions Improve the Quality of Performance

Obtain guidance from appropriate people, where necessary

The principal goal of the Test Engineers is to learn from experienced seniors and other employees. A Test Engineers is aware of his/her job role. Taking help from seniors and experienced professionals will only help increase the confidence of the Test Engineers.

A Test Engineers must keep these things in mind:

Dos and Don'ts while obtaining guidance from appropriate people:

- When you approach a person, ensure that the company protocol is followed. For example, if an associate requires some assistance from someone of higher authority, he/she must know the right person in the hierarchy to approach. Initially, a Test Engineers should ask for guidance from the immediate boss or the Team Lead. If the problem is not resolved, he/ she should seek help from the manager or someone immediate in the hierarchy.
- Asking for assistance directly from someone at the top of the hierarchy without addressing the same to the immediate boss is considered unprofessional too.
- Always carry pens and a notebook to document the main points and try to avoid forgetting things. Approaching a person for the same issue is both annoying and redundant. The importance of keeping Minutes of Meeting must be considered. Minutes of Meeting comprises the gist of the discussion. It helps keep a written documentary of the discussion and may be used as a future point of reference.
- It is advisable to fix an appointment with the concerned person beforehand, if possible.
- Behave professionally while you are taking someone's assistance. Do not interrupt with your personal opinion. If you want further clarification, ask questions after he/she finishes speaking. Raise hands, and then ask the question for clarification.
- Always thank a person after receiving assistance from them.

Receiving guidance is the first step toward continuous learning. However, the emphasis is on the application of the learning outcomes at work. A Test Engineers must try to apply the guidance to increase the quality of work. This helps one grow as a learner and boosts the confidence level.

Ensure work meets the agreed requirements

One must learn the importance of pre-planning and preparing an assignment so that it can be completed in a very organized manner. The importance and benefits are:

- Planning helps in making speedy and right decisions by providing a person with adequate guidelines.
- Planning helps in preparing a person for unexpected situations and complex outcomes, thus helping the person in control of the situation in a better way.
- Preparing and planning help in optimally dividing resources like finances, raw materials, time and manpower.
- Preparing and planning help in defining, identifying, and quantifying goals so that appropriate methods can be adopted to finish the assignment on time and in an organized manner.

Work Requirement	How to Ensure that the Requirement is Met
Activities	• Ensure that you have a clear idea about the work requirements • Seek guidance from colleagues/seniors concerned persons in case of doubts • Follow company policies and processes while planning activities
Deliverables	• Deliverables are considered the output of the work • Plan in accordance with the timeline given to you to complete the work so that the deliverables are not hampered • Always measure the required output and devise a process to achieve the same
Quantity	• Quantity is the amount of work that needs to be done to attain the deliverables • Don't let the quality be affected due to quantity and vice versa • The balance between quality and quantity must be maintained
Standards	• Understand the client requirements as per SLA • Ensure that you know the acceptable standards of performance and perform accordingly • Keep the compliance-parameters in mind and do not violet the rules while working on a project
Timing	• Be aware of the deadline before starting a work so that you can plan your work accordingly • Set small and achievable targets. For example, decide how much work should be done on the first day and achieve it on that very day • Do not compromise on the quality to meet the deadline. For this, you need to plan things before and execute the same.

Methods Adopted to Gather Requirements and Analyze

1. While working as Test Engineers or an entry level employee:

   - Organize, plan and prioritize the work order and jobs received
   - Conduct and organize technical assignments in optimal manner
   - Plan to utilize time and equipment efficiently
   - Concentrate on tasks and finish them within time limits
   - Assist in record keeping and proper documentation
   - Work on the completed project based on the feedback received
   - Ask for help from domain experts and seniors if issues arise during work
   - Organize and plan own work in a way that all the tasks are completed on time and as per specifications
   - Organize work as per job specifications
   - Organize and plan maintenance and cleaning activities

Requirements of Decision Making

Decisions at workplace must be taken swiftly, based on the best researches, prior experience and data gathered from relevant field experiences. Such data provide 'Evidence' and help a person in making suitable decisions.

The requirements of efficient decision making process are given below:

- One must have adequate yet controlled access to accurate and consistent data.
- One must stick to appropriate processes in analyzing data. For example, data analysis based on the set parameters of the organization. Ensure the reliability of the source of the data.
- One must take decisions swiftly, based on the analysis of collected data.
- Apart from analyzing the researched data, one must rely on practical experiences while taking decisions.

The role and responsibilities of Test Engineers include the following:

- Ability to troubleshoot everyday issues .
- Analyze critical points in daily tasks through observation and experience and identify control measures to solve the issue. For example, if you are working on a particular domain on a regular basis, know the weaknesses and strengths of the point. Thorough knowledge helps to solve issues quickly.
- Find smart ways to finish a job.
- Decide whether to accept or reject particular process flow (technical aspects) based on quality parameters. For example, if process A is fit to achieve a specific goal, it does not mean that process A should be followed in every case. Consider varied processes to accomplish different goals.
- Take decisions of one's own roles and responsibilities.

Steps involved in the Decision Making Process

The steps involved in the general decision making process are mentioned in the diagram below:

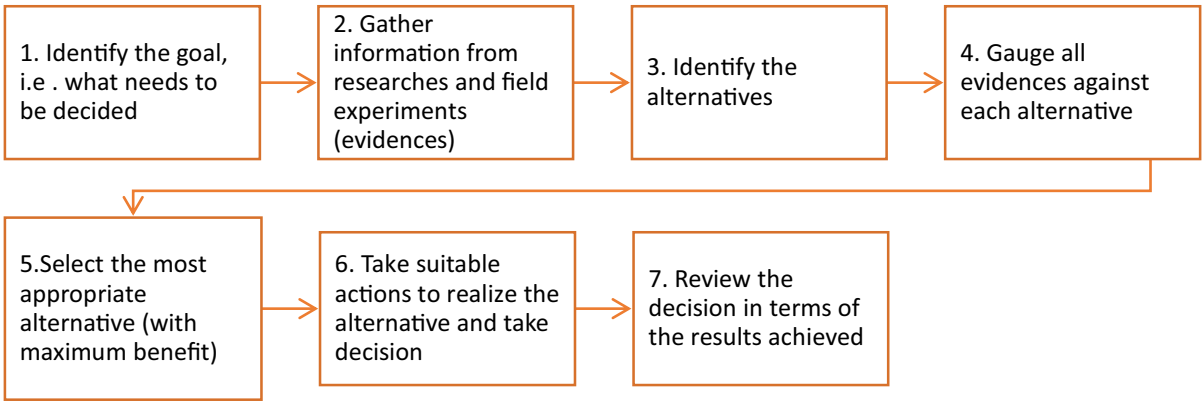


Fig 9.1.2 Steps Involved in Decision Making

Notes

UNIT 9.2: Work Ethics to Follow in an Organization

Unit Objectives

By the end of this unit, participants will be able to:

1. Demonstrate how to keep your immediate work area clean and tidy.
2. Explain the process of utilizing time effectively.
3. Discuss the importance of treating confidential information correctly.
4. Practice working in line with organization's policies and procedures.
5. Recognize how to work within the limits of job role.

9.2.1 Utilize Time Effectively

According to the popular saying, time is money. Not only are there various reasons for the need of time management, but there are also several rewards. Implementing effective time management skills enables one to finish more work in less time, resulting in increased free time, decreased stress, and enhanced job quality. Each advantage of time management improves another part of your life, and the cycle continues indefinitely.

Fig 9.2.1 Time is Money

Here, we will discuss several tips for effective time management that may enhance your work and you as an individual.

- **Time is Limited:** Everyone gets the same amount of time each day, and it's limited, therefore it's imperative to make the most of your time if you ever want to be efficient at the workplace.
- **Be Decisive:** There are many options in life to choose from. When you practice good time management, you have more time to breathe; this allows you to control which choices are the best to make. You're more likely to jump to conclusions and not fully consider the different options when you feel pressed for time and have to make a decision; this leads to poor decision making.
- **Accomplish More in Lesser Time:** You're able to stay focused on the task at hand by taking control of your time. This leads to higher efficiency because you never lose momentum. Imagine running a mile where you stop every 5 seconds; this would cause you to become tired very quickly and take much longer to complete the run.
- **Learn More in Shorter Period of Time:** When you control your time and work more efficiently, you're able to learn more and enhance your experience faster. There is a reason some students graduate before the others, so imagine applying time management throughout your entire career. You will not only stand out from the rest, but you will gain experience much quicker and be able to move up in life a lot sooner.
- **Enjoy More Success:** Time management is the key to success; it lets you to take control of your life rather than follow the flow of others. You achieve more, you make better decisions, and you work more efficiently; this leads to a more successful life.
- **Reduce Stress:** One of the main causes of stress is that people get rushed. The phrase "I have so much to do and so little time to do it" is spoken with frustration which leads to stress. With good time management, you know how much time you have, how long it will take to get your tasks done, you achieve more, and have more free time. This gives you more time to collect yourself, which reduces the feeling of being rushed and which in turn leads to less frustration and stress.
- **Higher Quality of Work:** We all need some free time to unwind and relax but, unfortunately, many of us don't get much free time since we're too busy to keep up with our daily activities and work load. By applying time management skills, you can get more done in a shorter period of time leading to more free time.
- **More Discipline in Life:** When you practice good time management in your life, you are less likely to leave work for later. Time management leads to higher efficiency and leads to a disciplined life.

The three most important tools for good time management are given below:

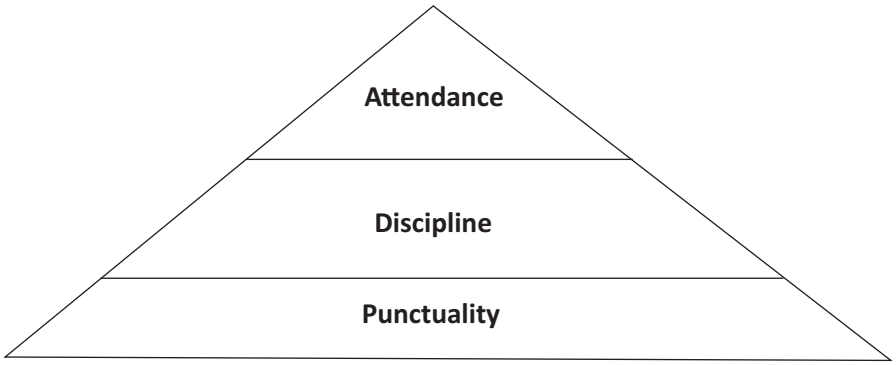


Fig 9.2.2 The Pyramid of Effective Time Management

Tips to manage time effectively:

- Adhering to one's plan
- Keeping a daily To-Do list
- Dividing one's time wisely
- Sticking to and respecting deadlines
- Not wasting time

9.2.2 Basic Work Ethics to be Followed in an Organization

Keep immediate work area clean and tidy

Keeping one's work area tidy and clean carries the following benefits:

- Avoiding accidents at the work area and the corresponding injuries and damages
- Lessening an employee's exposure to Occupational Hazards
- Permitting easy flow of materials
- Enhancing productivity at work
- Enhancing the employee's control on the various tools and equipment

The work area can be kept tidy and clean in the following manner:

A. Keep work area in a tidy and organized state

- **Clear Clutter and Spills:** Clearing clutter and spills helps prevent Slips, Trips and Falls.
- **Control Dust and Debris:** Get rid of the debris every day to avoid untidy workstation.
- **Store Tools and Equipment Appropriately in their Designated Storage Locations:** CD/DVDs should be kept in the correct place. Similarly, for any other device like pen drive, which is used daily to test/ store works, the same step should be taken.
- **Abide By a Written Set of Guidelines:** Keeping one's workstation/desk clean, tidy and organized, one must follow the SOPs (Standard Operating Procedures).
- **Store and Maintain Personal Protective Equipment (PPE) Appropriately:** Anti-glare spectacles, earmuffs are important for Test Engineers. Anti-glare goggles help reduce the stress of looking at the screen for a long time. Ear muffs negate the noise.

B. Keep work area safe

- Promoting the practice of looking out for signs like "Wet Floor" or "Cleaning under Progress"
- Promoting the practice of reading Directions of Use and MSDS sheet before using any chemical.
- Reporting each incident of spill (oil, grease, chemical, etc.) to the housekeeping staff with immediate effect.
- Avoiding Fire and Electrical Hazards.
- Avoiding storage of heavy objects at high and elevated areas.

Treat confidential information correctly

As a Test Engineers, one works in projects with confidential information. Even in the SLAs, upholding confidentiality is a major clause and the violation of the same leads to termination.

Every client wants to see that certain information is kept confidential. Information leaks can cause a huge loss and thus, Test Engineers must be cautious while working. To prevent information leaks which may be implied as security breach, one must install anti-spam, anti-spyware and anti-phishing tools.

To understand how to treat confidential information,

- Maintain that all the new employees accept and sign the Confidentiality Agreement
- Recognizing the relevant organizational confidential information and details must be taken into consideration. These may be in the form of:
  - Verbally disclosed information
  - Written information
  - Slides and Handouts
  - Visual information
  - E-mail and file documents
  - Carefully reviewing the NDAs from 3rd Parties

Work in line with organization's policies and procedures and Work within the limits of job role

Abiding by compliance is a must. Additionally, the company or brand image also gets affected due to work accuracy. A meticulous work helps in the development of company image as well as the ethical views in front of the professional community.

The new joinees will never understand the importance of submitting precise work if they are unaware what the expected benchmarks are. So, it is important that trainers of the particular company lay down the objectives to the Test Engineers. One of the best ways to do so is to take the help of "SMART."

- S – Specific
- M – Measurable
- A – Achievable
- R – Relevant
- T – Timely
    - o Slides and Handouts
    - o Visual information
    - o E-mail and file documents
    - o Carefully reviewing the NDAs from 3rd Parties

1. **S - Specific:** It stresses on specifying a specific goal rather than a general one. A specific goal will usually answer the "Why", "What", "Who", "Which" and "Where" questions. While carrying on with the daily responsibilities, the Attendant must take care and abide by these 5 questions, to remain clear about his/her goals and if, they are aligned towards the interests of the organization. The operator should not only work to achieve success for oneself, but also for the betterment of the organization as a whole.
2. **M - Measurable:** It stresses on measuring the progress towards the attainment of goal. A measurable goal usually answers the "How many?" and "How much?" questions. Whatever activities the operator should perform daily, his/her achievements must be expressed in volume, workload or quantity. This should be reported to the immediate supervisor or the Line Manager, who would be appraising the Attendant at the end of the year. A Daily Work Report (DWR) should be maintained and emailed to the supervisor daily.
3. **A - Accurate & Attainable:** Achieving all predefined metrics and abiding by the Service Level Agreements (SLAs). Every task should be error-free.
4. **R - Relevant and Reporting Real-time:** All activities and achievements, relevant to the job role, should be reported to the Line Manager, as and when accomplished.

5. **T - Timely & Target-oriented:** Activities should be prioritized according to long term and short term goals. Short term goals should be divided into small, achievable, measurable and time-bound steps. This, in turn, would induce a sense of urgency, promptness and ownership towards one's duties.

Fig 9.2.2 Implementation of SMART Module Enhances Quality of Performance

Brainstorming

There are several ways that can improve the quality and effectiveness of brainstorming. They are:

- **Be attentive to everyone's ideas:** People pay more attention to their own ideas; however brainstorming calls for equal exposure to the thoughts produced by others.
- **Avoid face-to-face groups:** Using face-to-face groups affects assessment apprehension, social loafing, production blocking, and social matching.
- **Include both individual and group approaches:** The process that helps members mix their ideas into the group is brain writing. Here, the members write their ideas on a piece of paper and then pass it along to others who add their own ideas.
- **Take breaks:** Permit silence sessions during group discussions so that members have time to think things through.

There is no fixed timescale for work completion for any profession. The duration to complete a project depends entirely on the set rules of a company.

To ensure your work meets the agreed requirements, consider these points:

- Type of the project
- Time period allocated for research
- The time to acquire relevant data and gather resources
- Outlining the direction and the flow of work
- Analyzing data for creating test cases/prototypes

Implications of Brainstorming Sessions

Timescale and deadlines are entirely based on the complications in a project. Where efficiency of an employee, technical aspects, and accuracy in work are constantly checked, submitting the work to the clients at the designated or promised deadline is also a must. The end date or the deadline of a project is planned as per rough calculations.

In case the work or the project is not delivered on time, there are high chances of the company to bear the brunt on the stricter side. Some of the implications of a missed deadline can be:

- Weaker commitment from burnt out employees
- High-stress level
- Low morale
- Huge penalties imposed by the client
- Loss of revenue for the company
- Loss of opportunities and potent chances for business growth
- Negative effect on the brand reputation of the company

The most common factor that leads to task inaccuracy at the workplace is a misunderstanding. During OJT (On-the-Job-Training) in most companies, workers are left at the mercy of employees, who belong to other departments or different projects. Now, as these employees are in charge of assisting the new trainees at work, they provide all necessary information to prepare them well for future endeavours.

It is important that the trainers should belong to the department where the employees are placed and provide precise details. The same goes for the employees who need opening up to their trainers and leaving behind the inhibitions to finish their work with precision.

Summary

- Relevant data is important in every professional field.
- Implementing good time management skills allows one to complete more work in a shorter time, which leads to more free time, reduced exposure to stress, and improved work quality.
- When you control your time and work more efficiently, you're able to learn more and enhance your experience faster.
- One of the main causes of stress is that people get rushed.

Activitiy

Activity 1: Interactive Session

- The Trainer asks the Trainees the following question: "How to achieve SMART goals in an organization."
- The Trainees are expected to raise their hands to volunteer and speak.
- The language spoken during the session should be known by the majority of Trainees in the class.
- While the session goes on, the Trainer should jot down the crucial points on the Whiteboard with the help of a marker.
- The best answer shall be appreciated by the Trainer in front of the whole class.

Activity 2: Prepare a List

- The Trainer divides the class into few teams, depending on the batch strength.
- Each team will have an even number of Trainees.
- The Trainer will instruct the Trainees that each of the teams must prepare a list of Resources that are required for the job role of a Test Engineers.
- After the team of Trainees create the list, they will submit it to the Trainer.
- The Trainer will evaluate the 5 lists and select the best one.
- The Trainer will read out the appreciated list in front of the class.
- The most extensive and detailed list shall be appreciated with accolades.

Exercise

A. State whether the following statements are CORRECT or INCORRECT

1. A project or a task which is both important and urgent is basically deadline driven. ☐
2. The end date or the deadline of a project is designed on the basis of certain rough calculations. ☐
3. Proper engagement and open communication amongst employees make the working process in a company more effective. ☐
4. Sharing of information or updating the section which you are working with your team members will help them get a better idea of your progress. ☐

B. Answer the Following Questions

1. How should one obtain guidance from seniors or supervisors?
2. What are the benefits of time management?
3. Write a short note on the "Triple Constraint Triangle".

Notes

Scan the QR Code to watch the related videos

https://www.youtube.com/watch?v=z_SeOWtpJVc

Work Ethics to Follow in an Organization

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

IT - ITeS SSC
NASSCOM

10. Work Effectively with Colleagues

Unit 10.1 - Team Work and Communication

Unit 10.2 - Significance of Healthy Team Bonding in Ideal Work Culture

SSC/N9002

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Discuss the importance of team work and communication.
2. Analyze the significance of healthy team bonding in ideal work culture.

UNIT 10.1: Team Work and Communication

Unit Objectives

By the end of this unit, participants will be able to:

1. Explain the principles of clear communication.
2. Outline the importance of being a good listener and adhering to the commitments.

10.1.1 Different Types of Information Required by Colleagues

In all companies, departments like operations, management, and recruitment play a great role. This may be in relation with finance, operations, marketing or any other department. As there are different teams present in an office, so are the types of information.

Communicate With Colleagues Clearly, Concisely and Accurately

Effective communication allows us to comprehend the problems that our colleagues are facing, and asking them to portray it clearly. To impart clear and precise information, we need to:

- Spread positive attitude
- Ensure proper understanding regarding strategy and goal accomplishment
- Ensure that everybody complies with the company's regulatory bodies

Respect Colleagues through the Actions

- Assisting, working and asking for help from colleagues to show trust and respect.
- Increasing the morale for an effective teamwork.
- Increasing open communication between co-workers.
- Working with your colleagues to increase productivity.

Seek guidance from appropriate people to agree the analysis to be performed on the data

- One's supervisor is supposed to be one's mentor at work.
- Ask questions to clarify doubts.
- Assistance and direction must be sought from the supervisor whenever necessary.

When necessary, inquire about and request clarification on work-related duties.

A Technical Support Executive, who has joined a new project, might face issues regarding technical aspects and process. It is always preferable to ask inquiries to clarify doubts than to brood over them in silence.

There are designated departments for each aspect of an organization. For example, HR, transports, security, operations are departments that handle different issues. In case of any doubt regarding HR policy, one must seek assistance from HR personnel. If an executive gets stuck with a technical query, he/she must approach someone in operations.

- Question/s must be asked to clarify doubt and to narrow down communication gaps with one's supervisor.
- Obtain and seek clarifications on policies and procedures, from the supervisor or other authorized personnel.
- Report and identify any possible deviations to appropriate authority.
- Any doubt/s can approach the supervisors or other authorized personnel, if the Technical Support Executives has doubts about the organizational policies and SOPs.
- Precisely receive information and instructions from the supervisor related to one's work.
- This must be done to get a proper idea about the responsibilities expected by one's supervisor.
- Having a proper idea about one's tasks helps in fulfilling targets successfully.
- Address the problems efficiently and report if required to immediate supervisor appropriately.
- Receive instructions clearly from superiors and react effectively on the same.

10.1.2 Effectively Communicate with Clients

Outsourcing plays instrumental role in generating revenue. Multiple projects run at a time in a company and each project might have different clients with different requirements and expectations. Therefore, Client communication is a vital thing to know for a Technical Support Executive.

- Communicate and politely, clearly, precisely
- Empower the client by putting adequate value to his / her views
- Recognize the client as a "Partner" and not just "customer"
- Stay honest in dealing with customer
- Maintain contact and inform the client of current and upcoming deals and offers.

- Exceed client's expectations through impeccable deals and service
- Negotiate fairly, politely but firmly
- Know and learn about the client
- Resolve service issues and concerns promptly
- Empathize with the client and apologize, in case of grievance and complaint

Review the analysis results with appropriate people and implement modifications.

The job role of Technical Support Executive is dependent on the different technical aspects and the technology changes quite rapidly. To keep up with the pace, executives must take vital updates from concerned people. Seeking and following up for feedback is another way to increase the quality of work.

Good feedback help the organization increase the service provided. Moreover, implementation of good feedback earns the respect and trust of the clients.

It might happen that there are certain flaws and inaccuracies in the work done that need to be taken care of. Project Manager, Quality Assessor and immediate supervisor like the Team Lead are the most appropriate persons to review the performance.

A performance report is generated on a weekly or monthly basis. The parameters vary from company to company on which executives are evaluated. The report is then shared with the executives.

As a Technical Support Executive, one should accept the feedback in a positive way and work on the areas of weakness. The main reasons behind review results are mentioned below:

- Introducing the Technical Support Executive to the process flow so that he/ she gets familiar with the common or organizational practices.
- Marking the probable areas of weaknesses. However, the executives should view it as a “scope of improvement” on which he/ she must work on to develop into a stronger professional.

The parameters of review are:

- Grooming (Verbal and Non-verbal)
- Discipline and Integrity
- Time Management
- Team Work (how the executive is as a team player)
- Attendance and Absenteeism

Each company maintains a specific mode of evaluating performances of the employees. SMART (Star, Medium, Average, etc.) is used for performance reviews. Star performers are recognized and the persons who lag behind are equipped with facilities to enhance their performance. The employee should comprehend the process and requirement of the company and then groom himself/ herself accordingly. In some cases, SMEs or Subject Matter Experts are appointed to evaluate the performance of the employees.

Notes

UNIT 10.2: Significance of Healthy Team Bonding in Ideal Work Culture

Unit Objectives

By the end of this unit, participants will be able to:

1. Identify challenges and pain points related to work distribution while working in a team.
2. Explain the importance of distributing and sharing workloads.
3. Demonstrate how to carry out commitments they have made to colleagues.
4. Identify any issues they have working with coworkers and take the initiative to address these issues.
5. Discuss the importance of following the organization's policies and procedures for working with colleagues.

10.2.1 Understanding a Colleague's Problems and Providing Support

Inform Colleagues in Advance

Certain things, at times, lead to time constraints. For example, owing to the work pressure, one might need to stretch work hours. It not only becomes your responsibility but a humane approach to assist your colleagues when they are stuck at any project.

When you comprehend the perspective of your colleagues, you create good communication and mutual bonding between yourself and your colleagues.

It isn't essential that you can do every task that is given to you. There may be times when you will be shifted to another project based on priority. In such cases, it is better to let your colleagues know what is happening. This lets your colleagues to find an appropriate replacement that can carry on with the work you are initially supposed to do.

Time Management is about managing your time. It is about making a commitment to be more organized, uphold your focus and use your time to your advantage.

Identify concerns with coworkers and fix them

A Problem can be defined as a difficult or unexpected situation, regarded as unwelcome and needing to be dealt with and overcome. Problems can take the form of intricate puzzles and riddles.

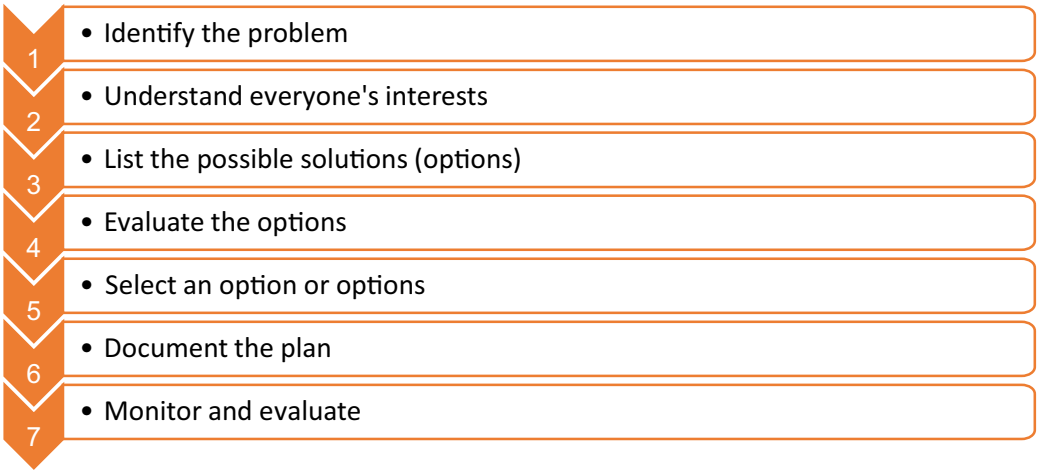


Fig 10.2.1. Stages of Problem Identification and Solving

Solving Problems while ensuring health and safety at workplace

- Analyzing the situation and taking suitable actions while dealing with team members
- Analyzing, assessing and deploying the information gathered from observation, experience, reasoning, or communication to act efficiently
- Identifying flaws in software, tools and equipment and ways to resolve them on time
- Ensuring timely correction of errors for reducing rework

Follow the organization's guidelines for collaborating with coworkers.

A Technical Support Executive must connect and coordinate not only with clients, but with seniors and supervisors in the organization as well. The elements of communicating effectively with clients, peers / colleagues and supervisors are given below:

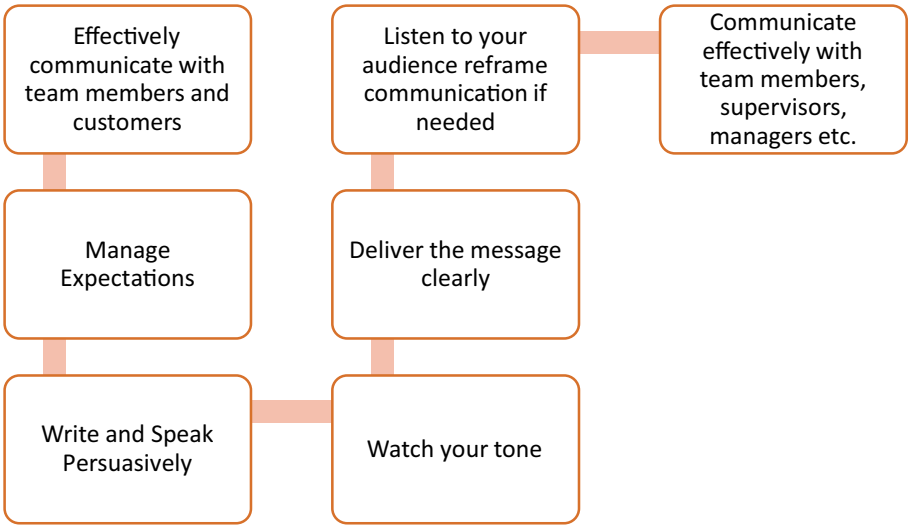


Fig 10.2.2. Essential Steps of Effective Communication

A. Cooperate and coordinate with coworkers to accomplish work objectives

- Listen actively with minimal barriers
- Build trust, but do not get too casual
- Participate and coordinate
- Watch your body language
- Share best practices with peers
- Be aware of your tone
- Ask questions to clarify
- Discuss task lists, schedules and activities

B. Effectively Communicate with Supervisors

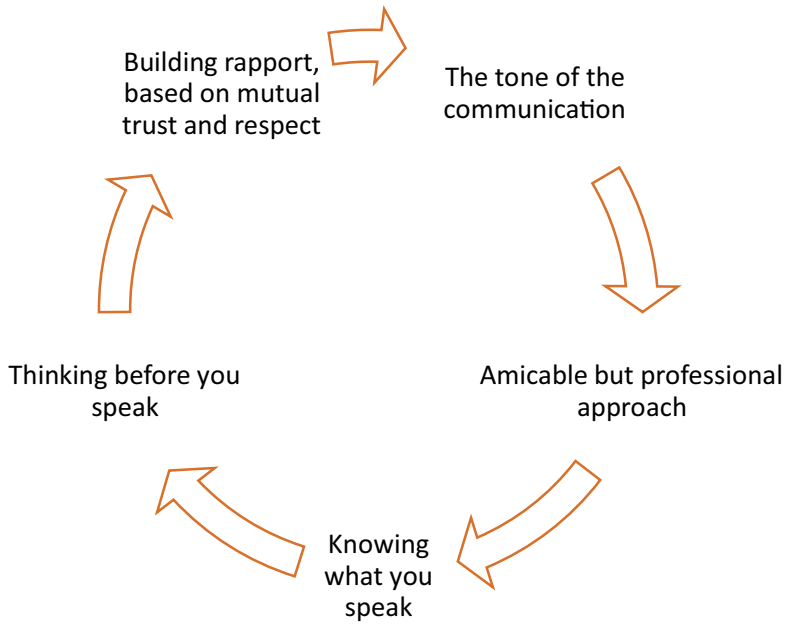


Fig 10.2.3. Essential Qualities for a Good Communication

According to the dictionary, a grievance is "a complaint or a strong feeling that one has been treated unfairly."

A. Follow the escalation matrix in the event of any complaint

- Before complaining about and expressing grievance, be very clear of the objectives, i.e. why do you need to complain and what do you want to accomplish in the long run
- Adhere to the Escalation Matrix for Internal Grievance Resolution

- Write an email to the designated official at each level of the matrix, according to the guidelines and formats provided.
- Follow up with the concerned official, if the grievance or complaint is not addressed within the standard TAT at that escalation level.
- Document all records of emails and phone calls until the issue is duly addressed and closed.
- If the concerned official, at a certain level, does not address the grievance within the TAT, escalate and carry forward the issue to the next level.
- Repeat process from 2-6.
- On resolution of the grievance, phone or email the concerned authority to thank them.

B. Addressing Worker's Grievance

- Hold an official yet private meeting with the worker.
- Acknowledge the grievance and empathize.
- Sustain confidentiality of the entire matter.
- Invite witnesses, if deemed absolute essential.
- Depending on the intricacy of the grievance, continue with further investigation.
- Obtain information to support your decision.
- Take the final decision.
- Escalate the matter to the immediately next level, if the worker is not satisfied with the decision taken.

C. Addressing Client's Grievance

- Do not contradict with or avoid the client from talking.
- Listen actively and patiently.
- Apologize and empathize with the client.
- Listen to the grievance / complaint with an open mind.
- Promise that you will get back to him / her with a permanent solution without delay.
- Keep your promise and respond to the client with a solution within the standard TAT.
- If you cannot address the issue on your own, escalate it to the next level.
- Follow up with the concerned officials till the grievance is addressed and the issue resolved.
- Let the client know over email or phone that his / her grievance has been taken care of.

Summary

- In all companies, departments like operations, management, and recruitment play a great role.
- One's supervisor is supposed to be one's mentor at work.
- Certain things, at times, lead to time constraints.
- A Technical Support Executive must connect and coordinate not only with clients, but with seniors and supervisors in the organization as well.

Activitiy 

Activity 1: Con-Vid Session

- In this session, the Trainer will play 2 videos.
- The first video will be about 10 ways to build good relationships with your co-workers.
- The YouTube link for the video is: https://www.youtube.com/watch?v=VLRMnPRJK6c
- The other video will be about how to communicate with your co-workers.
- The YouTube link for the video is: https://www.youtube.com/watch?v=B6h9QMBC9cw
- The students will watch the video attentively with pin-drop silence.
- They must note down crucial and relevant points from the video.
- Students will maintain decorum in the class and will not talk, whisper or discuss in the class.
- In case of queries or doubts, students will write those down in their notebooks.
- After watching the videos, the students can ask questions to clarify doubts.
- The students will raise their hands before asking questions.
- The Doubt Clarification session will be in the form of a discussion round, where the answers can be given by either the Trainer or any of the students knowing the answer.

Activity 2: Role Play Session

- In this activity, the Trainer will divide the class few groups, depending on the batch strength.
- The first group will enact and showcase the incorrect method of communicating with colleagues.
- The second group will enact and showcase the correct method of communicating with colleagues.
- The third group will enact and showcase different methods of building a good work relationship with colleagues.
- The group portraying the given role in the best way will be declared the winner and will be appreciated in the class.

Exercise

A. Fill in the Blanks

1. _____ Communication allows us to understand the problems that our colleagues are facing.
2. In case of IT companies, _____ plays instrumental role in generating revenue.
3. Seeking and following up for _____ is another way to improve the quality of work.
4. A Problem can be defined as a difficult or _____ situation.
5. The employees can share the information through secure _____.

B. Answer the Following Questions

1. Why is it important to know the process of communication with clients?
2. How can a Technical Support Executive improve his/ her performance based on review?

Notes

Scan the QR Code to watch the related videos

https://www.youtube.com/watch?v=R-YZonX_GSg

Importance and Understanding Problems from Your Colleague's Prospective and How to Provide Support

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

IT - ITeS SSC
NASSCOM

11. Maintain a Healthy, Safe and Secure Working Environment

SSC/ N9003

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Analyze different types of breaches in health, safety, and security.
2. Explain the evacuation procedure.
3. Identify how to summon medical and emergency services.
4. Identify the method of health, safety and accident reporting.
5. Identify the Government agencies in the area of safety, health and security and their norms and services.

UNIT 11.1: Breaches in Health, Safety and Security

Unit Objectives

By the end of this unit, participants will be able to:

1. Analyze different types of breaches in health, safety, and security.

11.1.1 Breaches in Health, Safety and Security and Accident Report

Some important points in the operations related to reporting and response related to safety breach are:

- Safety breaches in the designated premises are "Incidents" that need to be reported and duly responded to.
- Reporting a safety breach is done by providing an Incident Report.
- An Incident Report must comprise the following aspects:
  - The person/s involved (details of the offender/s)
  - What exactly happened
  - Number of casualties
  - Where it happened (location of the incident)
  - When did it happen (Exact time, when the incident took place)
  - Why it happened (factors that caused the incident; the holes and gaps in the existing security system)
  - Description, features, peculiar features and condition of the affected people, vehicles, properties, and goods

The common format of the Incident Report is given below:

INCIDENT REPORT

Day/Night: _____ **Date:** _____ **Time:** _____

Report Writer (Print) Name: _____

Co No: _____

Company: _____ **Section:** _____

Telephone no: _____ **Extension:** _____

INCIDENT: (Summary: (Who, what, where, when, why, how, etc.)

ACTION TAKEN BY SECURITY:

RECOMMENDATIONS/COMMENTS/REFERENCES:

Copy to Client: _____ **Date:** _____

Fig 11.1.1. Sample Incident Report Form

Notes

UNIT 11.2: Evacuation Procedures for Workers and Visitors

Unit Objectives

By the end of this unit, participants will be able to:

1. Demonstrate the process of evacuation.

11.2.1 Fire Safety

Perform Fire Evacuation Steps

The full form of EHS is Environmental Health and Safety. It is a discipline that studies and deploys the practical aspects of environmental protection and safety at work. Simply, it is what organizations and workshops must do to ensure that their actions do not cause harm to anyone.

The EHS commands that there must be specific escape routes or safety evacuation points. This includes thorough plans or blueprint of the building which is understandable to anyone.

Each floor of the workshop or building must have the Safety Evacuation Map. These are mainly applicable for cases of Fire outbreaks or natural calamities like Earthquake, Flood, etc.

The sequence of an Evacuation situation is given below:

1. Detection
2. Decision
3. Alarm
4. Reaction
5. The movement to an area of refuge or an Assembly station
6. Transportation

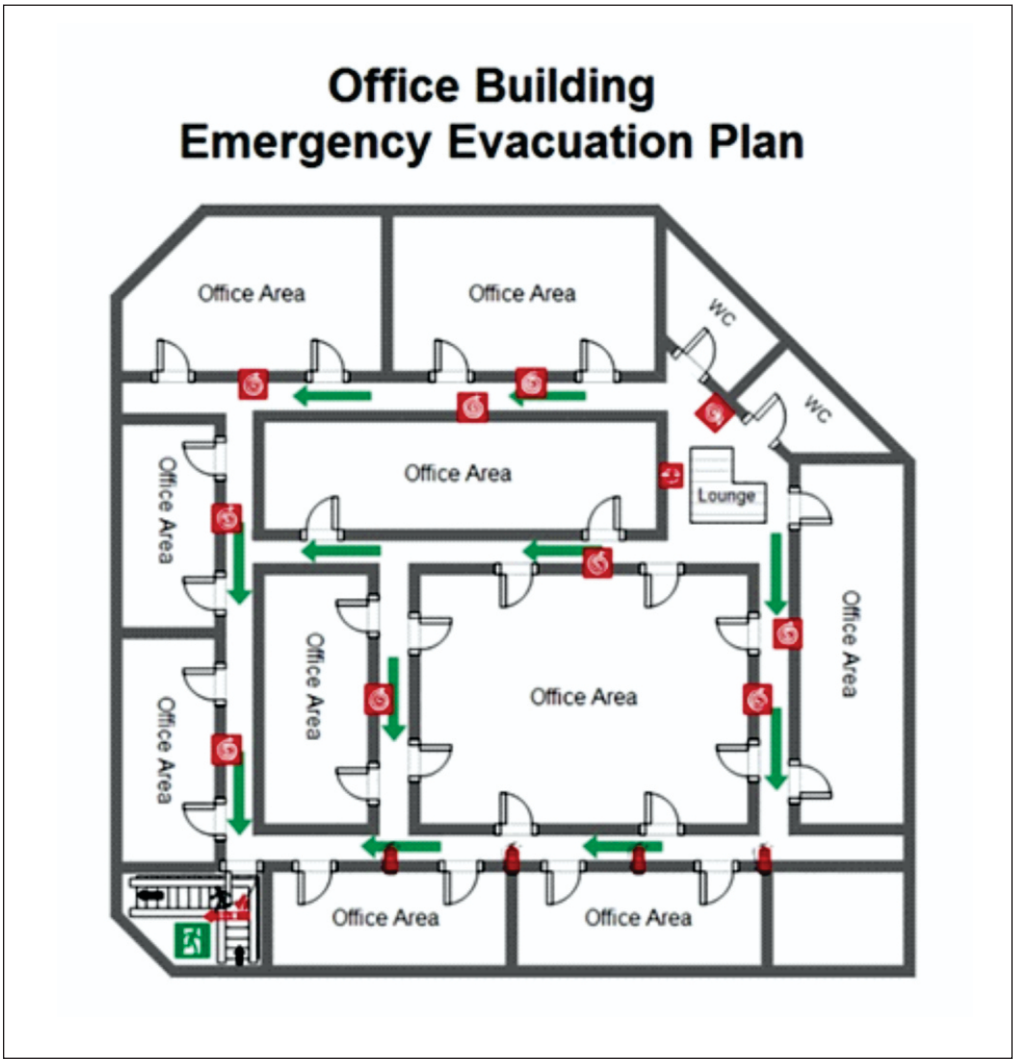


Fig 11.2.1 Fire Escape Plan

Carry out Evacuation of Casualty and Premises Tasks

Briefing and Guidance for Fire Fighters

There are basically three methods with the help of which people can be rescued from a building engulfed in fire. To ensure on-site reception, here are two of the important steps that must be taken into consideration. These come under the best safe carrying and lifting practices.

Conventional Technique

- This is a good method if there is an open area nearby.
- The first rescuers will make the victim sit reach under their armpits and grab their wrist.
- The other rescuer will cross the ankle (victim) and pull up that person's legs on his shoulder.
- Finally, on the count of 3, both will lift the person up and move out.

What to do during Bomb Threat?

- Don't be panic-stricken and try to keep your calm.
- Open the emergency exit gate so as to propagate the evacuation process.
- Think ahead and consider places where a bomb can be planted.
- Don't assemble in the common assembly area because terrorists want to kill as many people as possible. The common assembly area is the place where the evacuees assemble and therefore the possibility of killing a maximum number of people is in the common assembly area. Do not consider the common assembly area during the evacuation at the time of bomb threat. Always assemble at a place which is not premeditated.
- Inform the local police immediately.
- Evacuate immediately after receiving a bomb threat and don't wait until something is found after investigation.
- Document everything and submit the documentation to the concerned authority.
- If anything suspicious comes into sight, barricade it with red ribbon maintaining a diameter of 100 meters. Ensure that no one comes within the boundary. Bring sandbags and put them around the barricade to minimize the effect of the blast.
- Don't try to touch any suspicious object and wait for the police to arrive at the spot to diffuse it.

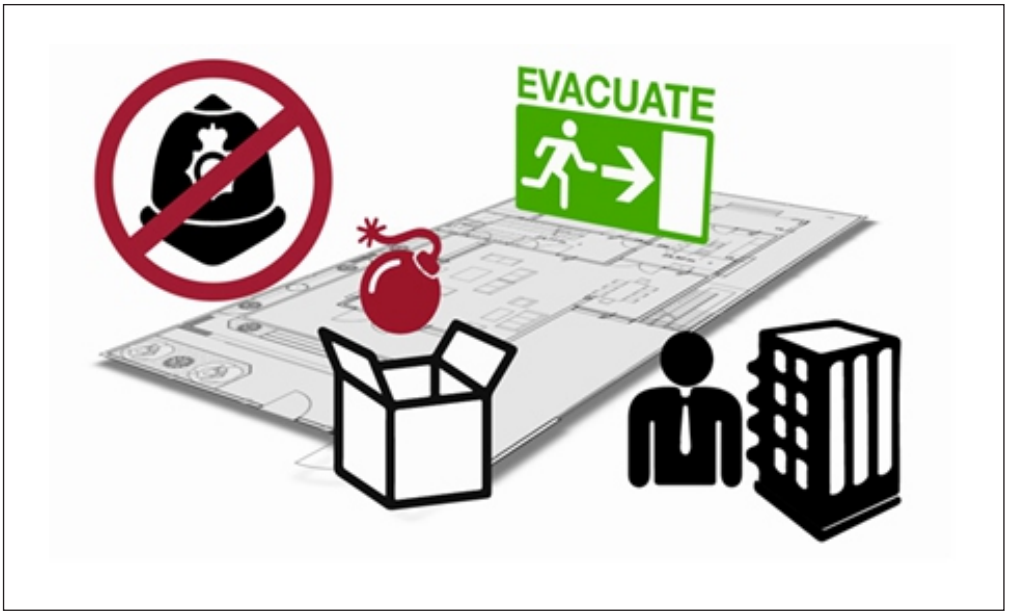


Fig 11.2.2 Proper Evacuation Procedures During Bomb Threat

For Fire Outbreak:

The emergency and evacuation procedures are given below:

- A clear passageway must be present to all escape routes.
- Signage like escape routes should be clearly marked.
- Don't use the Elevator during a fire.
- All people at the workplace must be given brief instructions about the positions of the escape routes.
- Enough exits and routes must be there for all people to escape.
- Emergency lighting (Infrared lights for night and blurred vision) must be present.
- Emergency doors, that open easily, must be present.
- Brief instructions must also be given regarding the availability and use of fire extinguishers.
- The workplace must have a safe meeting point or assembly area for the staff.

Correctly demonstrate rescue techniques applied during fire hazard:

1. Responding to Fire

- The Fire Alarm System must be initiated and an alert must be raised.
- The appropriate class of Fire Extinguisher must be chosen.
- A safe evacuation path must be identified before dealing with the fire.
- Immediate evacuation must be initiated if the extinguisher is exhausted and the fire still exists.
- Call the workplace security or the local emergency services.
- Summon the fire-fighting services at the earliest.
- Look out for the nearest emergency exit routes and call out for people, who you can take along with you.
- Always use a staircase and not the elevator.
- While opening a door, first touch the door with the back side of your palm.
- The P.A.S.S technique must be adopted for extinguishing the fire.
- Always move downstairs and avoid returning to the burning premises, till the fire-fighters arrive.
- As you move out of the building, gather people, whoever you come across.
- Stay as far as possible from smoke, because smoke may comprise toxic gases.
- Cover your mouth and nose with a damp cloth to protect yourself. If possible, help your colleagues (those who are with you) to repeat the same.
- Keep doors open, after you open them.
- Start moving out of the building and ask your colleagues to do so.
- Do not rush.

2. Initiate Evacuation

- Stop your work and move out safely and without spreading panic.
- Carry only the most important items like cell phone.
- Await instructions from the Safety Committee.
- Leave the workplace from the nearest door bearing an "Exit" sign.
- Report to the designated Assembly Area.
- Incorporate first aid treatment to anyone in need.

For Natural Calamities / Disasters:

2. Flood and Storms

The emergency and evacuation procedures are:

- Move to the high grounds and help others move before the flood strikes
- Stay alert, avoid panicking and monitor the surroundings with eyes and ears open
- Accumulate disaster supplies like:
  - o Canned, dry, ready-to-eat and packaged food, which do not require refrigeration or cooking
  - o Liquid cash
  - o Drinking water in clean containers
  - o First Aid Kit
  - o Adequate batteries
  - o Flashlights
  - o Essential clothing
- Instruct people around you not to drive.
- Shut off the Mains Supply (electricity) at the circuit breakers.
- Do not walk or swim through the flooded water.
- Stay alert for evacuation calls and help people identify alternate routes of getting there.

3. Earthquake

The emergency and evacuation procedures are given below:

- Inform others in the area by raising an alarm if they have not heard it while you are evacuating yourself.
- Quickly shutdown any hazardous operations or processes .
- Exit the room.
- Take jackets or other relevant clothing material needed for protection from the weather.
- If possible, close windows and doors as you leave, but avoid locking the doors and emergency exit routes.
- Exit the building and walk to the nearest safe exit route.
- Do not run.
- Do not use elevators.

For Accidents:

The emergency and evacuation procedures are:

- Summon emergency medical help by ringing the Safety Committee officials or the toll-free number.
- One must inform the immediate supervisor about an injury or illness.
- Check and examine the site, to gather as much information as possible, so that the same can be provided to the emergency team, once it arrives.
- One must extend help and assistance to others.
- If possible, workers may treat themselves to first aid or ask colleagues to do so.

The general steps involved in carrying out an evacuation are:

- Stop your work and move out without spreading panic.
- Gather and carry only the most important items like cell phone.
- Report to the designated Assembly Area.
- Leave the workplace through the nearest door bearing an "Exit" sign.
- Await instructions from the Safety Committee.
- Incorporate first aid treatment to anyone in need.

Evacuation and emergency procedures for the especially abled:

- With Impaired Hearing
    - o Turn lights on/off to gain the person's attention, or specify directions with gestures, or write a note with evacuation directions.
- The Visually Impaired
    - o Announce the type of emergency.
    - o Offer your arm for help.
- People with Prosthetic Limbs, Crutches, Canes, Walkers, etc.
    - o Evacuate these individuals along a route specially designated as injured persons.
    - o Accompany and assist to evacuation site if possible.
    - o Notify emergency crew of their location.
    - o Use a sturdy chair, or a wheeled one, to move the person to an enclosed stairwell.

Notes

Scan the QR Code to watch the related videos

https://www.youtube.com/watch?v=2_5B7e5YWuM

Evacuation Procedures for Workers and visitors

UNIT 11.3: Medical Assistance and the Emergency Service

Unit Objectives

By the end of this unit, participants will be able to:

1. Identify how to summon medical services and emergency services.

11.3.1 Respond to Emergency Situation

How to Respond to an Emergency Situation

An Emergency can be defined as "a serious, unexpected, and often dangerous situation requiring immediate action." Responding to an Emergency situation while working at the site involves the given steps:

Undertake first aid activities in case of any accident, if required and asked to do so.

- First Aid is an emergency care or treatment given to an ill or injured person before regular medical aid can be acquired.
- Before administering First Aid to a victim, one must check the category and degree of emergency and then apply the techniques accordingly.
- Stop and take a look at the scene, and the person before reacting.
- Ask yourself the following questions:
  - What is the casualty?
  - What happened exactly?
  - What is the category and nature of the emergency?
  - Is the accident dangerous for the victim?
  - Is anyone else available at the place to assist?
- Do the following if the victim is conscious and injury is not dangerous:
  - Ask for the victim's consent before providing first aid.
  - Use appropriate PPE, if possible.
  - Interview the victim to ask basic medical questions, so that accurate information may be provided to the Emergency Medical Team, once it arrives.
  - Conduct a thorough check for unnoticed injuries.
  - Ensure appropriate care and technique.
- If the victim is unconscious, try reviving the person by addressing him / her, rubbing shoulders, hands or the sole of feet.
- Use the AED and use it, along with Artificial Respiration.

Report hazards that you are not competent enough to deal with to the relevant person in line with the organizational procedures and alarm others who may get affected.

- As an important part of the emergency management procedure, any workplace must designate a Safety Committee, which comprises liable and senior people from all departments and teams.
- This committee would act as the legislative body, the authority and the first point of contact for reporting any hazard, potential risks / threats and emergency situations in the workplace.
- This committee would also be liable to conduct training sessions, safety audits, and drills, to help all employees prepare themselves for emergency and unprecedented situations.
- The list of the committee members, their designations and job titles, as well as contact numbers, must be listed and circulated among the employees.
- The Safety Committee must comprise important members from the following departments:
    - Supervisor/Manager/Team Lead from each project
    - Security Services
    - Building Operations and Maintenance team
    - Counselling and Psychological Services team
    - Emergency Medical Services
    - Reception/Front Desk
- This list must be put up for easy display at popular parts of the workplace, in the form of an Emergency Escalation Matrix and must be updated daily.
- Furthermore, this list must be mandatorily included in every First Aid kit in the workplace premises, so that a person treating a victim with first aid techniques may call for additional help and report the accident.

Practice no Loss for Company Due to Safety Negligence

Safety negligence at the workplace or even at home can prove to be lethal to the individual. So to ensure that there no chances of safety carelessness, companies should follow these aspects:

- The companies should ensure that the wiring in the workplace is insulated.
- No malfunctioned machinery should be kept with the new or spare ones.
- No sharp objects or equipment are kept on the walkway.
- First aid kit should be kept either at the reception or in a separate medical supply area.
- There are no open or damaged sockets.

Practice regular safety drills for being prepared in the event of a fire or natural calamity

- The first step in this process is to raise the alarm as all companies and workshops do have push-glass fire alarm system. Breaking the glass and pushing the alarm button should be the first step to let the people know that the building is on fire.
- On hearing the emergency evacuation alarm, the foremost thing that a person must do is cease and wind up all activities and look for an exit path.
- The next should be to find out the place where the fire started.
- It should be followed by tackling the fire with an appropriate fire extinguisher.
- Meanwhile, a person from that workshop or building should call for emergency help services like ambulance and fire brigade officers.
- People should take the stairs to get out of the office building instead of using the lift.
- Every company should keep folding wheelchairs so that company employee or even visitors can transport individuals with severe mobility impairments or health.
- It is important that all individuals emptying the building should be calm and composed.

The method of using a fire extinguisher is to follow the method: P.A.S.S.

PASS is the acronym for:

- Pin (P)
- Aim (A)
- Squeeze (S)
- Sweep (S)

1. To use an extinguisher in a proper way, the first step is to pull the handle's pin.
2. The next step is to aim the extinguisher's nozzle. The direction should be toward the fire's base. This is because the sprayed foam at the top will diminish or extinguish only the fire at the top. This will not serve the purpose for which the extinguisher is used and the burned down flame may spring up to life if it gets enough oxygen or any combustible material.
3. Then in an extremely controlled manner, you need to release the agent. This is done by squeezing the trigger.
4. You already know that you should direct the nozzle at the fire's base. You must sweep the extinguisher's nozzle from left to right. Continue with this process until you put out the fire as you need to act fast as most extinguishers' discharge time is nearly 10-20 seconds.

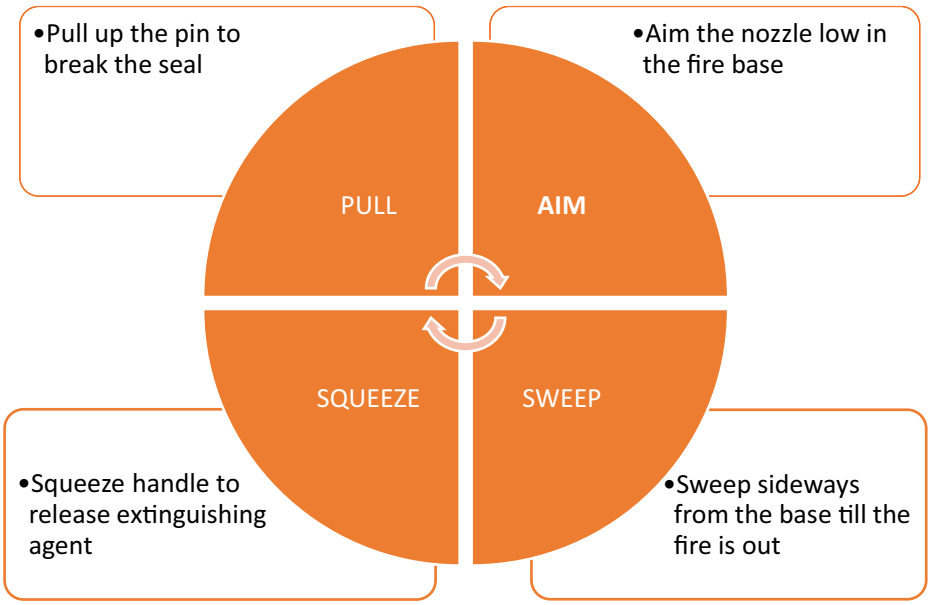


Fig 11.3.1 P.A.S.S Technique for Fire Fighting

Participate in Emergency Procedures

- **Raising Alarm:** Fire Alarms may either have a "Break Glass" or a "Pull / Push" mechanism. In case of the break glass system, the glass sheet must be forcefully hit with a clenched fist. One must continue repeating the process until the glass breaks. In case of the "Pull / Push" systems, one must break the glass first and then either pull down or push up the lever to raise the alarm.
- **Correct Assembly Point:** Proper instructions must be given to the workers about the site of and the directions to the correct assembly point in the workplace. Information about this must be given during mock evacuation drills and training sessions as well.
- **Safe and efficient evacuation:** Suitable evacuation procedures must be adopted for the common public and for especially abled persons. Specially-abled persons must be helped to evacuate the place by giving them access to Wheelchairs and other aids.
- **Roll call:** Once everybody has evacuated the building / workshop and arrived at the Assembly Point, Roll call or Head Count must be done to ensure that nobody is left behind in the affected area. This must be done mandatorily to ensure that everybody on the premises is safe.
- **Correct return to work:** Evacuation must be conducted in a very streamlined, organized, and noiseless manner. Likewise, everybody, who had evacuated the workplace, must return to his / her respective locations / positions / seats, following normal or emergency routes, depending on whether the situation has been re-established to normal or not. Once everybody is back in place, another Roll call is taken.

Demonstrate How to Free a Person from Electrocution

Electrocution is injury or death caused due to electric shock. The following steps must be adopted while freeing a victim from electrocution:

- **Approach**
  - The first step is to approach the spot to find out if you run the risk of electrocution as well.
  - Call for help from a colleague, who is trained in treating electrocution victims.
  - Inspect the accident scene to ensure if the source of electrocution is still active.
  - Inspect if the victim is still in contact with the source of shock.

Fig 11.3.2 Approach the Victim and Inspect the Accident from a Safe Distance

  - Detach the main power supply of the area.
  - Dodge any electrical conductors in the surroundings.
  - Touch the victim only if all power sources have been deactivated.

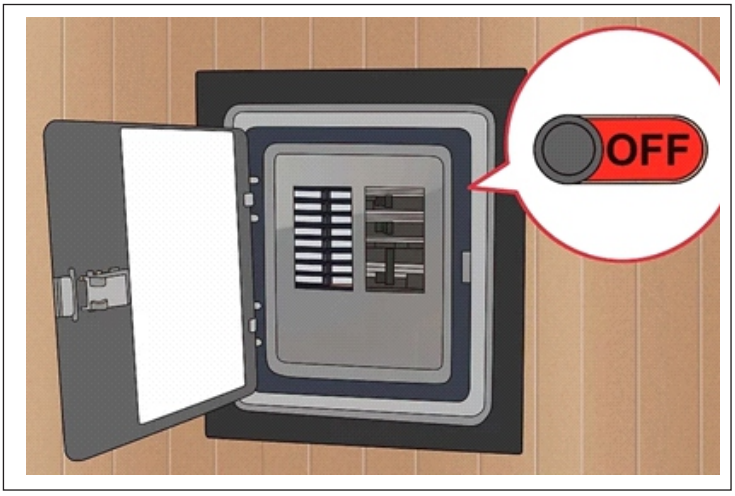


Fig 11.3.3 Disconnect the Source of Power

- In case it is impossible to deactivate the power supply, the victim must be removed from the location of the live power source.
- This should be done by wearing appropriate PPE.

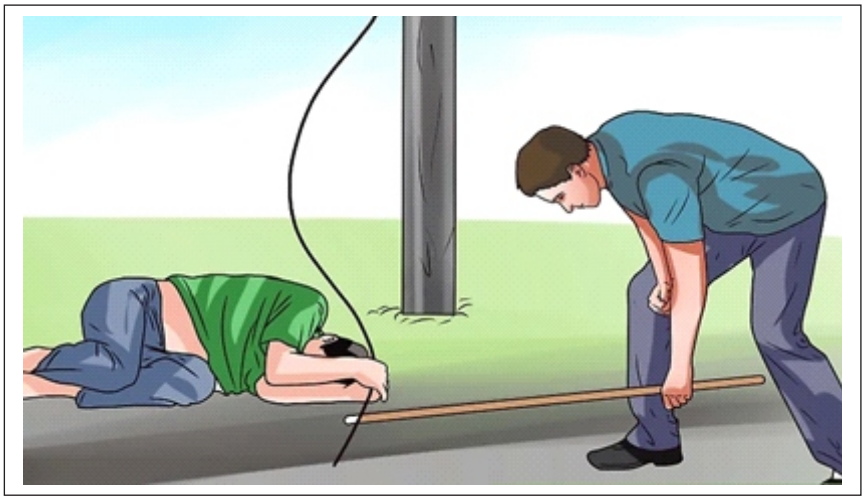


Fig 11.3.4 Use Insulators to Approach the Victim of Electrocution

- The victim must not be removed in case of neck or spine injury.
- The area must not be crowded so as to allow sufficient breathing air.
- The victim's pulses and breathing rate must be checked.
- CPR may be provided if required.

Fig 11.3.5 Perform CPR if Required

- Never touch the victim or the surroundings without detaching the main power supply.
- Wear appropriate insulating gloves and shoes in order to protect yourself from electric shocks.

Notes

UNIT 11.4: Health, Safety, and Accident Reporting Procedures

Unit Objectives

By the end of this unit, participants will be able to:

1. Discuss the method of health, safety and accident reporting, and the evacuation process.

11.4.1 Emergency Procedure and Reporting Accident

Follow the organization's emergency procedures quickly, efficiently and calmly

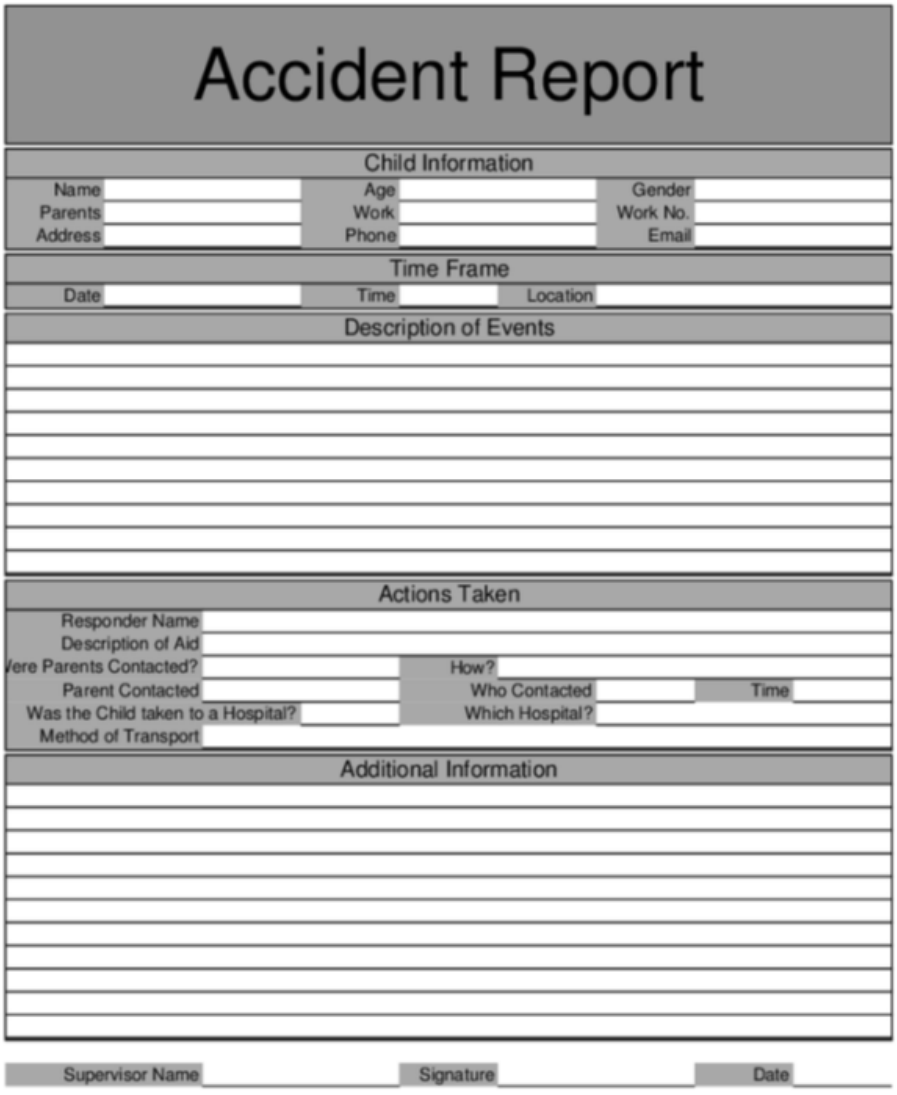
Evaluating the Emergency

- One must rationally and critically think and assess the severity of the emergency and determine, what requires to be done on an immediate basis.
- One must remain calm and composed during an emergency situation since stress during an emergency complicates things and may confuse a person.
- The emergency dispatcher aims at providing instant and appropriate help based on the nature and degree of emergency.
- One must look for additional help by calling up the emergency toll-free number, which would help the caller reach an official or 'dispatcher'.
- One must help the dispatcher by answering his / her questions and providing the dispatcher with the precise location and nature of the emergency.
- It is suggested that one should call from a GPS equipped phone so that the dispatcher is able to track the location, even if the caller is unable to speak.
- One must be aware of the nature of the emergency, i.e. whether it is a medical, mental health or behavioral emergency.
- One must evaluate the immediate threats, for example, in case a person is severely injured from a running machine, the machine must be turned off instantly to prevent others from getting hurt as well.

Handling the Emergency

- Extremely high casualties must be informed to the Occupational Health and Safety Committee (OHSC).
- One must move farther from the emergency spot and help others follow the same.
- Secondary Hazards must be removed or mitigated, at least. For example, a car accident comprises the risk of a violent explosion and fire outbreak resulting from spilled fuel.
- One must not feel guilty if nothing can be done to help the others.

- In case nothing can be done to lessen the severity of the situation, one must provide support to the others by uplifting them mentally, inquiring about their medical history, noting events as they occur, etc. This information may prove vital for the emergency response team.
- One must help the other victims and take suitable measures to assist the specially abled ones.
- One must refrain from moving a severely injured victim and provide only the basic first aids.
- Once the emergency team arrives, assist them with all required and relevant information.
- A First Aid kit must be used if required.
- One must try reviving a seemingly unconscious victim by rubbing the chest, pinching the earlobes, providing Cardiopulmonary Resuscitation (combination of chest compression and artificial respiration).



Accident Report

Child Information

Name		Age		Gender	
Parents		Work		Work No.	
Address		Phone		Email	

Time Frame

Date		Time		Location	

Description of Events

Actions Taken

Responder Name					
Description of Aid					
ere Parents Contacted?		How?			
Parent Contacted		Who Contacted		Time	
Was the Child taken to a Hospital?		Which Hospital?			
Method of Transport					

Additional Information

Supervisor Name		Signature		Date	

Fig 11.4.1 Sample form of Reporting Accidents

1. **Work safely at all times, complying with health and safety legislation, regulations and other relevant guidelines**

   - Ensure that all emergency route maps are on display in the accessible places on all floors of the building.
   - Ensure that appropriate Fire Extinguishers are present on all the floors of the workplace.
   - Learn and abide by company policy and procedures for dealing with security risks in the establishment.
   - Ask your supervisor how you may retrieve PPE and its maintenance and storage.
   - Stay aware that confined spaces must bear suitable signs, to restrict claustrophobic people from accessing them.

2. **Ensure that health and safety instructions applicable to the workplace are being followed**

   - Lighting should be satisfactory in all areas and additional bulbs should be kept handy.
   - While using cutting tools, the direction of cutting should always be away from your body.
   - Arrange for frequent Safety Drills and Trainings for employees to endorse safety awareness.
   - Ensure that all manual cutting tools must be honed in advance because blunt tools may slip and lead to deep cuts.
   - Have a clear idea of how much authority and accountability you have to deal with security risks, including your legal rights and duties.
   - Learn and abide by company policies and procedures for maintaining security while you work.

3. **Check the worksite for any possible health and safety hazards**

   - Employ a Safety Supervisor in the workshop.
   - Have your employer develop a daily checklist for all areas delegated to suitable employees.
   - This Safety Supervisor will stay accountable for checking the worksite for potential health and safety hazards.

4. **Follow manufacturers' instructions and job specifications relating to safe use of materials specifically chemicals and power equipment**

   - Ensure that all Chemical Solutions used on display shelves or for Housekeeping purposes must be used only after mentioning to the relevant MSDS (Material Safety Data Sheets) or Instruction Manuals.
   - Loosely fitted clothes must be totally avoided because the loose ends may get caught in powered machinery and tools and may be lethal.
   - Ensure that you read the Instruction Manual thoroughly before using powered tools and equipment.

5. Follow electrical safety measures while working with electrically powered tools & equipment

- Powered tools and equipment must be reviewed for any damage, before and after every use.
- Damaged switches must be reported to the supervisor and repaired with immediate effect.
- Plugs must be checked for missing or faulty prongs / pins.
- The power cord must be assessed carefully for any fraying, faults, cracks or loss of insulation.

6. Ensure safe handling and disposal of waste and debris

- All walkways should be free of clutter and debris, to avoid trips and falls.
- Any spill should be cleared off instantly and 'Wet Floor' or 'Work in Progress' signs should be used in suitable places.
- Store equipment, Tools and Chemicals should be stored correctly, abiding by all instructions provided in the Instruction Manual and 'Directions for Use'.

Ensure electrical safety compliances and EMI/EMC hygiene requirements are met as per the guidelines

The risks associated with the use of electrical equipment are extended to both the user and his / her surroundings in the workplace. Few of such risks are mentioned below:

- Lethal Electrocution accidents.
- Non-fatal electric shocks leading to serious burn injuries.
- Non-fatal yet severe shocks leading to damages caused to the internal tissues and vital organs like the brain and the heart.
- Non-fatal yet painful static electric shocks.
- Falls from cranes, ladders, and scaffolding and resulting mechanical injuries due to electric shocks.
- Explosions and fire outbreaks caused by the sudden ignition of flammable materials.
- Health issues like nausea, muscle spasms, unconsciousness, and palpitations.

Identify and modify any hazards that you can deal with competently, safely and within the limits of your authority

1. Safety

- Operational safety
  - o Employee safety
  - o Building and plant safety
  - o Process safety
  - o Accident prevention
  - o Emergency management
  - o Risk management

- Safety of chemicals and biological materials
  - o Handling
  - o Safety data, documentation
  - o Storage
  - o Transport

2. Security

- Personnel
- Products
- Knowledge
- Physical assets
- Information (not covering core IT security like data access control, firewalls, virus Protection etc.)

3. Health

- Health protection at the workplace
  - o Industrial hygiene
  - o Occupational medicine
  - o Accident prevention
  - o Noise
  - o Occupational toxicology
  - o Stress, mental health
  - o Biosafety
  - o Ergonomics
  - o Radiation protection

4. Health Promotion

Environmental Protection

- Emissions
  - o Into the air
  - o Into the soil
  - o Into the water

- Waste
  - o Avoidance
  - o Disposal
  - o Re-use, recycling
- Resources, including energy
  - o Change to best alternatives
  - o Eco-balance
  - o Efficient use

Remediation of Existing Contamination

The management of SHE issues should be organized and clearly communicated. For this reason, it is ideal to encourage the use of a framework that includes each of the basic steps in a management cycle.

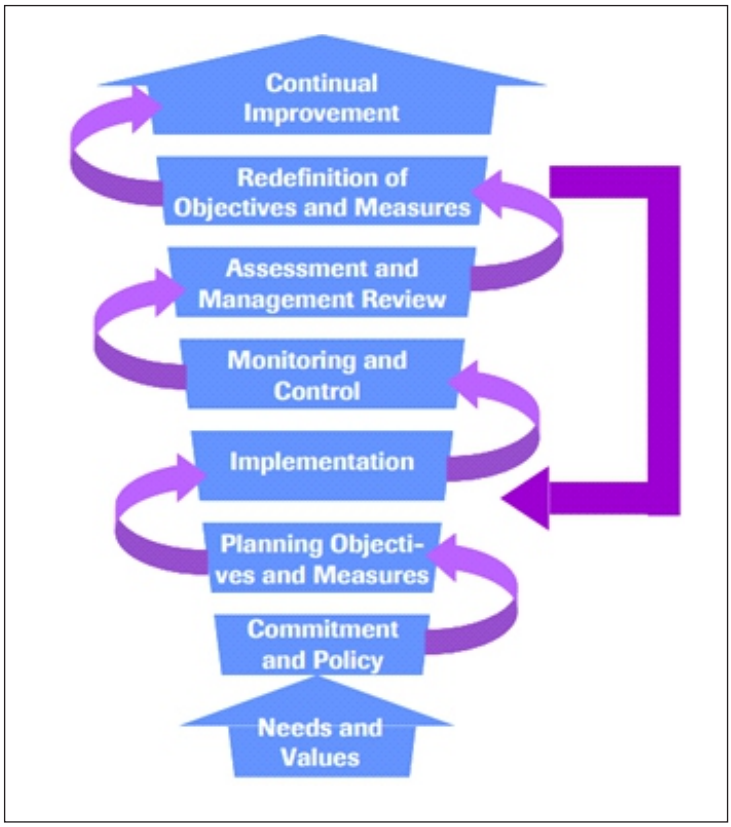


Fig 11.4.2 Step-by-Step Evaluation and Implication of SHE Aspects

Notes

Scan the QR Code to watch the related videos

https://www.youtube.com/watch?v=bsafBtbJmrk

Method of Health, Safety, and Accident Reporting

UNIT 11.5: Safety, Health, and Security Agencies' Norms and Services

Unit Objectives

By the end of this unit, participants will be able to:

1. Evaluate the safety and emergency readiness of the site.
2. Identify and rectify any hazards that can be dealt with competently, safely, and within the boundary of authority.
3. Identify and recommend opportunities for improving health, safety, security to the designated person
4. Explore the process of completing health and safety records legibly and accurately.

11.5.1 Government Agencies for Safety at Workplace

Identify and recommend opportunities for improving health, safety, security to the designated person.

Hazard and Operability (HAZOP) Study

- This technique involves a structured and methodical examination of an present method / procedure, thus, in turn, classifying and assessing the associated hazards.
- These hazards can be easily recognized in the form of Deviations in the process parameters (physical conditions and elements like flow, temperature, pressure, humidity, etc..
- A Deviation is a manner in which the process conditions stray away from the probable values.
- The severity of Deviation can be illustrated with the help of specific and prearranged Guide Words

The steps involved in conducting HAZOP are:

- Segregating the entire system or process into components or sections
- Select a study node or point
- Define the predictable outcome or consequence
- Choose a process parameter, based on the expected consequence
- Use a suitable Guide Word
- Find out the Cause behind the deviation
- Start with the cause that may lead to the worst possible consequence
- Evaluate the deviations thus detected
- Plan and prescribe action
- Record and document information
- Repeat the process from B

Common examples of process conditions / parameters are given below:

- Temperature
- Pressure
- Flow
- pH value
- Signal
- Mixing
- Viscosity
- Time
- Control
- Separation
- Addition
- Reduction
- Communication
- Sequence

Creating reports with comprehensive information is a must for every organization. The main idea behind this is to let the management body of the company as well as HSE to know the hazards at the workplace.

With the help of such reports, the company can examine, pinpoint the risks, and carry on the essential improvements within the organization. Because of such reports, companies can recognize long-term risks and short-term risks and achieve remedial actions for those risks.

In case of security-related issues or health-related issues, it is always a better choice to inform your supervisor or seniors.

A company can function in a systematic, smooth and successful way if it looks after the satisfaction of its employees. OH & S is one of the safety platforms where every corporation has to meet the safety guidelines.

Three specific articles (as per Indian constitution) ensure occupational safety and health for workers. Those Articles are:

- 42
- 39(e and f)
- 24

Some government agencies that look into the safety and security of individuals at the workplace are follows:

- Labour Departments (for both UT and State)
- Ministry of Labour
- Government of India
- NSCI (National Safety Council of India)
- National APELL (Awareness and Preparedness for Emergencies at Local Level)

Complete Any Health and Safety Records Legibly and Accurately

1. Health and Safety File

- These are electronic files that contain all the central safety and health records of the business. The other information kept in this file are given below:
- Copies of Risk assessments which covers the areas like:
  - Lifting operations
  - Lifting equipment
  - Manual Handling operations
  - Fire risk assessments
  - Lone working
  - COSHH (Control of Substances Hazardous to Health) assessments
  - General risk assessments
  - Risk of violence and aggression
  - Display Screen equipment workstation assessments
- For Organisation Health and Safety Risk Assessment, maintenance of risk assessment registers
- Copies of safety and health policies, guidance and procedures (local)
- Copy of the organization's Health and Safety Codes
- List of individuals (name) who are given the liability of examining the safety and health issues. They may be like:
  - Union Health and Safety Representatives
  - Risk assessors
  - Fire evacuation officers
  - DSE (Department of Sustainability and Environment) assessors
  - First Aiders

2. Log-book for Health and Safety

This involves:

- Risk assessment
- Training to overcome such situations
- Fire drills

Some other health and safety are:

- Information based on organizational meeting with Area Health and Safety Committee
- Annual safety and health audit checklists and applicable action plans
- Checklists for safety and health induction
- Copies of Safety Matters like an official newsletter
- Health and safety training records that have information regarding:
  - Name
  - Date
  - Health and safety training's course title
  - Response like Attended or Not Attended
  - Date fixed for Test Engineers training
- Fire Drill Records
- Examination and statutory inspection reports
- Material Safety Data Sheets
- Equipment maintenance and their service provision
- Record of dates for repeat of test, Portable Electrical Appliance tests and remedial action required
- Details related to emergency procedures

Summary

- Safety breaches in the designated premises are "Incidents" that need to be reported and duly responded to.
- The full form of EHS is Environmental Health and Safety.
- The first rescuers will make the victim sit reach under their armpits and grab their wrist.
- Information based on organizational meeting with Area Health and Safety Committee.
- One must rationally and critically think and assess the severity of the emergency and determine, what requires to be done on an immediate basis.
- First Aid is an emergency care or treatment given to an ill or injured person before regular medical aid can be acquired.

Activitiy

Activity 1: PowerPoint Preparation Activity

- The name of this activity is "PowerPoint Preparation" activity.
- In this activity, the Trainer will divide the class into few groups depending on the batch strength.
- Each group will be given 3 different topics on which they have to provide a broad explanation.
- Group A will write on the value of reporting accidents.
- Group B will write on Evacuation procedures.
- Group C will write about the correct method of firefighting.
- It is important that the Trainees present their answers not only rich in information but also supported by diagrams.

Activity 2: Evacuation Drill – Mock Practice

- This activity is in the form of "Evacuation Drill and Quiz Contest".
- The trainer will ask the trainees to practice evacuation drills. The trainer should guide them.
- After the evacuation drill is complete, there will be a quiz contest on the evacuation procedure.
- The trainer will conduct the quiz contest.
- The trainer will divide the entire class into two group.
- One group will be Team A and the other will be Team B.
- There should be a scorer to write points on the board.
- The trainer will ask questions from the book related to the topic discussed.
- For each correct answer, there will be 10 points, however, for wrong answers there will be a deduction of 10 marks.
- There will be 5 marks for each right answer given on a pass and 15 marks will be deducted in case the pass answer is wrong.
- There is no negative marking if a question is passed without any answer given.

Exercise

Choose the correct option from the list of responses to answer the following questions:

1. Find the odd one out in terms of the given statement:

The supervisor or the manager should see and identify the type of breach. It is only on the basis of the severity of the breach the appropriate actions can be taken. The actions can be like:

a) Dismissal

b) Felicitation

c) Warning

2. An emergency is –

a) Unexpected

b) Anticipated

c) Predictable

3. OHSC stands for –

a) Organizational Health and Safety Committee

b) Occupational Health and Safety Community

c) Occupational Health and Safety Committee

4. Most fire extinguishers' discharge time is near -

a) 15-20 seconds

b) 10-20 seconds

c) 30 seconds

5. Which one of the followings is not a P.A.S.S component?

a) Aim

b) Sweep

c) Shot

6. EHS stands for –

a) Environmental Health and Safety

b) Emergency Health Security

c) Emergency Health and Safety

7. Flood is a –

a) Natural phenomenon

b) Artificial phenomenon

c) Cosmic phenomenon

8. During an emergency evacuation, employees should adjourn at –

a) Nearest police station

b) Nearest fire station

c) Assembly area

9. Nobody should use the _____ during fire.

a) Stair

b) Exit door

c) Elevator

10. As an important part of the emergency management procedure, safety com a _____ Committee.

a) Security

b) Safety

c) Health

Notes

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

IT - ITeS SSC
NASSCOM

12. Workplace Data Management

Unit 12.1 - Basics of Data and Information Management

Unit 12.2 - Follow the Accurate Process Fow to Analyse Data

Unit 12.3 - Generate Report Based on the Observations

SSC/N9004

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Discuss the basics of data and information management.
2. Explain the accurate process flow to analyse data.

UNIT 12.1: Basics of Data and Information Management

Unit Objectives

By the end of this unit, participants will be able to:

1. Discuss and agree with appropriate people the data/information they need to provide.
2. Collect the data/information from reliable sources.
3. Compute the accuracy and completeness of the data/information.

12.1.1 Provide Data/Information Including Sources

Obtain the data/information from reliable sources and Check that the data/information is accurate, complete and up-to-date

Before starting any new process, it is important that a Technical Support Executive has a proper briefing of the process. Additionally, it is also significantly important that he or she is given all the data, required materials and their sources.

The required information involves:

- Collecting essential requirements and analysis of those.
- Any specific implementation procedure.
- The client/s requirements.
- Software related to the new process.

12.1.2 Templates and Formats Used for Data/Information

Templates are standard formats for documenting observations. The observation includes various aspects of the company. The type or the key parameters of the template change depending on the department. For instance, the template that is used by the transport department of a firm should be entirely different to that of the technical department which take care of operations.

The information provided in the template is used for performance examination.

Different techniques used to obtain data/information

- **Case Studies**

  This method basically maintains that the information that is collected is based on the experience of the clients.

- **Focus Groups**

  This data or information method is reliant on group discussions with in-depth topic assessment. This can be about marketing tactics, evolutionary aspects related to data, their sources, and searches, codes, programming languages or even any form of bugs.

Notes

UNIT 12.2: Follow the Accurate Process Fow to Analyse Data

Unit Objectives

By the end of this unit, participants will be able to:

1. Demonstrate how to carry out rule-based analysis of the data/information, if required.
2. Identify who to go to in the event of inaccurate data/information and how to report this.

12.2.1 Rule-Based Analysis on the Data/Information

The rule-based study practically involves decision-making process or conditional branching. It is a design of methodology production whose basis lies in software factors - analysis of techniques to make appropriate decisions for a new project.

In this rule, we will find the presence of three or more conditions like pseudo codes or if statement. The performance requirement is the accommodation of rule engine solution.

Process of Application

1. Select the input variables

As there are many variables present in a new project, creation of a matrix is essential between methodologies and factors. We can see the presence of factors in methodologies.

2. Bad Sub Rules

There are certain types of factors that cannot be connected with other types of factors. If they are combined, this step cannot lead to the creation of bad sub rules. In this case, two rule categories are recognised. They are:

- System rules with high requirement stability, low complexity and small size system.
- System rules with low requirement stability, medium complexity and size system.

3. Variable Reduction

These factors are identified but their elimination doesn't make any impact. These generally comprise application domain and project type.

4. Category Merge

For methodology identification, formation of many categories takes place and its foundation can be on project type.

5. Examples

We can see the beginning of hypothetical example sets if we look at extreme cases. The rule-based analysis is based on the acknowledgment of factors like less complexity, high requirement stability, and small size.

How to report inaccurate data/information

In an organisation, the power structure often consists of a single or small group at the top and increasing levels of power below them. The majority of enterprises, governments, and organised religions are hierarchical organisations with varying levels of management, power, or authority.

Notes

Scan the QR Code to watch the related videos

https://www.youtube.com/watch?v=zFmNhF1W3Bk

Rule Based Analysis

UNIT 12.3: Generate Report Based on the Observations

Unit Objectives

By the end of this unit, participants will be able to:

1. Review the accuracy of work, involving colleagues where required.
2. Identify any unresolved anomalies in the data/information to appropriate people.
3. Produce complete, accurate and up-to-date data/information to the appropriate people in the required formats on time.

12.3.1 Provide Complete, Accurate, Up-to-date Data/Information in Required Formats on Time

Check the accuracy of work, involving colleagues and the formats in which you need to provide it

Every project has a stipulated timeline. A project commences with setting a goal followed by other aspects like developing, testing, and quality analysis and final deployment.

Set-up Goals

Every stage has its own format where information has to be filled in precisely. A project goal template must be implemented and details must be filled in regularly. A sample format is given below:

For methodology identification, formation of many categories takes place and its foundation can be on project type.

PROJECT GOAL AND OBJECTIVES WORKSHEET

PROJECT NAME		PROJECT MGR.	
DATE CREATED	VERSION DATE	VERSION NO.	0.0.0

TEST GOALS & OBJECTIVES AGAINST SMART CRITERIA
SPECIFIC • MEASURABLE • ACHIEVABLE • RELEVANT • TIME-BOUND

GOAL STATEMENT	
OBJECTIVE NO.	An objective should look like this: "To increase the native plants between 1st and 3rd Streets by 50% by March 31st."
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Fig 12.3.1 Project goal template

Summary

- Before starting any new process, it is important that a Technical Support Executive has a proper briefing of the process.
- The rule-based study practically involves decision-making process or conditional branching.
- Every project has a stipulated time line.
- Every stage has its own format where information has to be filled in precisely.

Activitiy

Prepare Mock Reports

- The trainer will divide the class in few groups depending on the strength of the batch.
- The trainees will prepare mock reports on the shared format.
- The trainer will share a format on which trainees should the report.
- Each box should be filled with accurate information.
- The trainer will check the mock reports generated by each trainee.
- The best performers will be recognized by the class.

Chart Paper

- In this activity, the Trainer will divide the class into 3 groups.
- Each group will be given 3 different topics on which they have to provide a broad explanation.
- Group A will require writing on Reporting method of Inaccurate Data/Information and whom to report.
- Group B will require writing on Anomalies That May Occur In Data/Information.
- Group C will require writing on Different Techniques Used To Obtain Data/Information.
- It is important that the Trainees present their answers not only rich in information but also supported by hand-drawn diagrams.
- The group which can present their answers in the best way within 30 minutes will be awarded appreciation and accolades.

Group Discussion

- This activity is in the form of "Group Discussion"
- The trainer will divide trainees into 5 groups
- All the groups will sit together to discuss the goals of a project, estimate the timeframe required and plan the deployment
- Every trainee should actively participate in the discussion
- Each group will carry a notebook and pen to chalk out details
- They will jot down the points, important dates and processes
- After the discussion, each group will produce a Minutes of Meeting
- The trainer will supervise the entire process and ensure each and every one participates in the meeting

Exercise

A. Fill in the Blanks

1. It is important to provide the executive with specific _____ procedure.

   a). Deadline

   b). Implementation

   c). Comment

2. The full form of OSI is _____.

   a). Open Source Initiative

   b). Outsourcing Solutions Inc.

   c). Open Switching Interval

3. One of the templates used for information or data is _____.

   a). Software Architecture Design

   b). Relative Theory

   c). Hardware design

4. _____ method is dependent on group discussions with in-depth topic exploration.

   a). Discussion

   b). Focus group

   c). Coding Theory

Notes

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

13. Maintain an Inclusive, Environmentally Sustainable Workplace

IT - ITeS SSC
NASSCOM

Unit 13.1 - Sustainable Practices

Unit 13.2 - Respect Diversity and Strengthen Practices to Promote Equality

SSC/N9014

Key Learning Outcomes

By the end of this module, participants will be able to:

1. Discuss the sustainable practices in the workplace to optimize energy usage.
2. Demonstrate proper workplace etiquette to promote equality by respecting diversity.

UNIT 13.1: Sustainable Practices

Unit Objectives

By the end of this unit, participants will be able to:

1. Demonstrate how to optimize usage of electricity/energy, materials, and water in various tasks.
2. Explain the process of implementation of energy efficient systems in a phased manner.
3. Identify and segregate recyclable, non-recyclable and hazardous waste generated for disposal or efficient waste management.

13.1.1 Optimize Usage of Electricity/Energy, Materials, and Water

Greenery within and around the office premises and other corporate environments helps not only to enhance the décor of the workplace, but also has a positive impact on the productivity of the employees. Greenery helps people to concentrate on work, creates positive vibes among the workers and the visitors.

Apart from the introduction of greenery, conservation of energy and optimization of usage are equally important. There are certain essential tools and equipment that are used in every workplace, which require electricity. For example, air conditioner, light, fan, computer, coffee vending machine are such electrical gadgets or appliances which are extensively used in the offices. Similarly, steady water supply in the washroom is another important requirement. Optimized usage of all these essential energy or commodities is absolutely significant to conserve energy and create an eco-friendly work environment.

What does greenery do?

- Plants in workplaces purify the air; they reduce the concentration of CO_2 (Carbon dioxide gas) and other volatile organic compounds, keeping the air fresh and healthy.
- External vegetation moderates heat in and around office block in the summertime, pulling down heat stress and decreasing the necessity for air-conditioning.
- Green roofs and facades proliferate insulation or the absorption capacity of heat, plummeting heating and cooling expenses.
- Plants in and around office buildings release water vapour which moistens the air, dipping headaches
- 'Green views' also boost focus, and aid quicker recovery from stress.
- Green environments encourage people to undertake activities such as a lunchtime walk, keeping staff alert and healthy. Long periods of sitting adversely affect health.

Plan the implementation of energy efficient systems

Here are some simple energy management ideas one can implement in the work station.

- Do not use artificial lighting in offices when natural light is sufficient
- Open draperies and raise shades whenever adequate light from windows is available
- Use energy-saving fluorescent lights and lamps
- Switch off lights and appliances in unoccupied office spaces or unused rooms such as conference room.
- Switch on the lights and ACs/ fans during the conference
- Turn off the bathroom's fan and lights whenever they are not occupied
- Install the light sensors to remind and educate office users about wasted light
- Use rechargeable batteries for calculators and other office devices
- Turn off computers that are not used, and utilize computers' energy/power management tools (i.e. sleep mode, hibernate mode, screen saver)
- Reduce the use of lighting during night cleaning
- Keep office doors and windows closed if heating and air conditioning is on
- Switch off HVAC systems in offices when they are not in use
- Ensure thermostats are correctly adjusted
- Purchase and use high-efficiency office equipment and devices
- Set up a self-adult system for the office energy consumption

Initiatives towards efficient use of natural resources and energy, reduction and prevention of pollution

These are some measurements that help optimize the usage of energy in the workplace. However, another important aspect of optimizing the usage of energy and other materials is proper maintenance. Organizations should prepare a checklist to measure and maintain energy and material conservation. Following is a sample checklist for the energy and material conservation module at workplaces.

Category	Checklist Items
Energy management	Establishment of energy management organization, and employee education
	Energy conservation targets and investment budget setting
	Grasp status of implementation of energy conservation
	Measurements and recording of monthly usage (electricity, gas, oil, and water)
	Preparation of statistics, including graphs showing differences from previous month or year
	Grasp of energy intensity (MJ/m2/year)
	Establishment of management standards
Heat source and heat-conveying equipment	Temperature control for chilled water, cooling water, and hot water
	Adjustment of the flow rate and pressure of pumps and fans
	Steam leakage and insulation management
	Management of air ratio and exhaust gas of combustion equipment
	Control of steam pressure and blow-down
	Cooling water quality control (electrical conductivity)
	Control of opening of valves and dampers (e.g. automatic valves)
Air-conditioning and ventilation equipment	Proper temperature setting
	Turning off air-conditioning for rooms not in use or unoccupied
	Adjustment of appropriate outside air intake volume
	Review of operating hours
	Effective operation of total heat exchanger (e.g. Rosunai)
	Local cooling and local exhaust
	Indoor air quality control (e.g. CO2)
	Installation of (manual or automatic) inverter device to ventilation fans
	Suspending either of the operation of a 4-pipe air conditioning system, if used
	Control of ventilation in car parking space (CO concentration control)

Water supply/drainage and sanitation equipment	Control of supplied water flow and pressure
	Water saving measures (e.g. water-saving top and automatic flashing)
	Change temperature and pressure setting on the heat source equipment depending on the season
	Operation with intervals in hot water supply circulation pump
	Utilization of rain water and well water
	Management of kitchen equipment (e.g. cooking and washing machines)
Management of electric power receiving and transforming facilities	Optimization of demand
	Usage control
	Voltage adjustments
	Power factor management
Operation management of lighting equipment	Optimum illumination control
	Switching off lights when they are not necessary (use of daylight)
	Cleaning of lighting fixtures and change to more energy-saving fixtures
	Replace incandescent lamps to fluorescent lamps
	Adoption of energy-saving FFE (furniture, fixture, and equipment)
Operation & management of elevating machines	Operation
	Adoption of inverter control
	Adoption of human motion sensors to escalator
Buildings	Blocking of solar radiation on the windows (e.g. shading curtains and light-shielding films)
	Blocking of solar radiation on the roof (heat reflection coating)
Others	Maintain the place around the condensing units for air-conditioning and chillers
	Utilization of heat from hot spring
	Installation of boilers using waste materials as fuel
	Utilization of solar heat
	Wind, solar, and small hydro power generation
	Use late-night electricity
	Co-generation

Table 13.1.1 Energy and Material Conservation Checklist

Various energy options including renewable and non-renewable

Renewable Energy is an endless energy source that does not deplete upon use and produces no or minimum waste. Such energy sources are renewed spontaneously on a human timescale. The International Energy Agency (IEA), an independent authority on Environment and Sustainable Development based in Paris, explains: " "Renewable Energy is produced from perpetually replenishing natural processes. It derives in its different forms straight from the Sun or from heat generated deep inside the earth. Electricity and heat generated from solar, wind, ocean, hydropower, biomass, geothermal resources, biofuels, and hydrogen obtained from renewable resources are included in the definition."

- **Wind Power:** Wind power is a source of solar power. Wind energy (or wind power) refers to the utilisation of wind to create electricity. Wind turbines transform wind's kinetic energy into mechanical energy. A generator converts mechanical energy to electrical energy.
- **Geothermal Energy:** Although the Sun warms the Earth's surface, it is not responsible for the planet's interior temperature.
- **Solar Energy:** Solar energy is the conversion of solar energy into thermal or electrical energy. Solar energy is the most abundant and cleanest renewable energy source currently accessible.

  Bioenergy is renewable energy derived from biological and natural sources. Even landfills and garbage zones are bioenergy resources due to technological advancements. It can be utilised as a renewable energy source, supplying heat, gas, and fuel.
- **Hydropower Energy:** Hydropower, often known as hydro-energy, is a type of renewable energy that utilises water held in dams and flowing in rivers to generate electricity in hydropower plants. The blades rotate a generator that turns the mechanical energy of the spinning turbine into electrical energy.

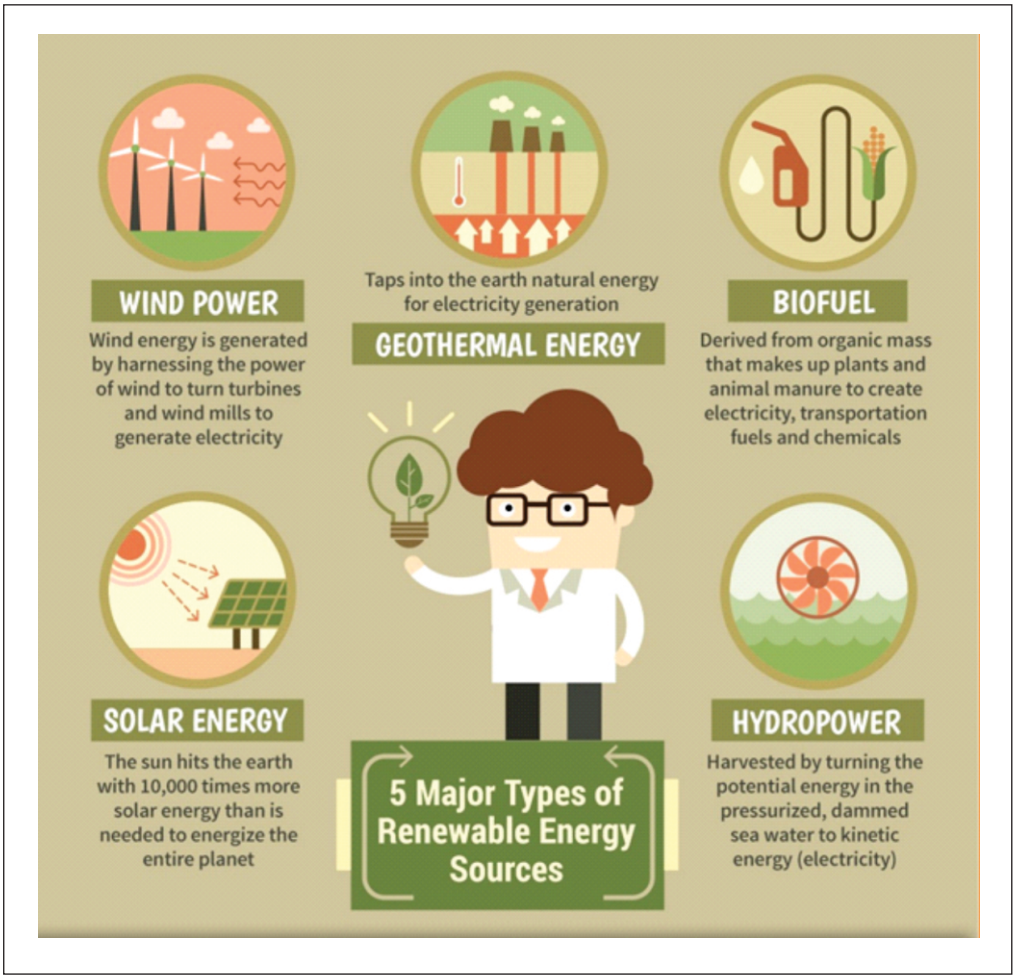


Fig 13.1.1 Renewable Energy Sources

Renewables generate no greenhouse emissions during energy production, making them the cleanest and most viable way to combat environmental damage. Unlike conventional energy sources such as coal, gas, oil, and nuclear, whose supplies are finite, clean energies are derived from and adapt to natural cycles. This makes them a crucial component of a sustainable energy system that allows for the development of the present without endangering future generations.

Electricity First Aid Emergency Procedures

The first aid kit should have the following essential items for giving first assistance:

- Cuts, scratches, punctures, grazes and splinters
- Muscular sprains and strains
- Minor burns
- Amputations and/or major bleeding wounds
- Broken bones
- Eye injuries
- Shock

To ensure that workers have a thorough understanding of first aid in the workplace, one must establish and implement first aid protocols. The procedure must include:

- The type of first aid kits and their locations.
- The placement of first aid amenities include first aid rooms
- Who is accountable for the first aid supplies and facilities, and how often should they be inspected and maintained?
- How to create and maintain adequate communication systems (including equipment and procedures) to ensure timely communication with first aiders in the event of an emergency.
- The essential communication equipment and methods when first aid is required (especially for remote and isolated workers). These procedures should include information on where the communication equipment is located, who is accountable for it, and how it should be maintained.
- The work locations and shifts assigned to each first aid responder. These procedures should include the names and contact information for every first responder.
- Arrangements to guarantee that first aiders receive adequate instruction.
- Arrangements to ensure that employees obtain proper first aid information, instruction, and training
- Requesting information about any first aid needs that may require specific treatment in a medical emergency, such as severe allergies, when an employee begins work. Information about a worker's health must be kept confidential and shared with first aid personnel only with the worker's permission.
- Instructions on how to report work-related injuries and illnesses.
- Methods to prevent exposure to blood and bodily fluids.
- What to do if a worker or other individual is too injured or ill to remain at work, such as if they need assistance with transportation to a medical facility, home, or a place where they may rest and recover.
- Access to debriefing or counselling services to assist first responders and employees following a significant workplace incident.

Here the steps to free a person from electrocution

Switch off the main power.	
Don't touch the person who is electrocuted.	
Try to remove the person from the electrical source with the help of non-conducting objects like stick, cardboard, bamboo, etc.	
Lay the person in this position.	

Table 13.1.2 Steps to save a person from electrocution

13.1.2 Segregate Recyclable, Non-Recyclable and Hazardous Waste

Hazard is defined as a factor, which may cause harm to people and properties alike, like electricity, inflammable products, explosive material, corrosive chemical, using heavy ladders at workplace etc. Simply put, a Hazard is simply a condition or a set of circumstances that present a potential for harm. Risk is defined as the likeliness or the chance that a hazard can actually cause harm to somebody. For example, smokers of cigarettes run the risk of developing Cancer. The potential or imminent danger that Risks and Hazards expose the concerned premises to, is known as Threat. For example, a person, who has the potential of blowing up a building, is a threat to that building and its inhabitants.

The steps involved in Risk Management are:

Fig 13.1.2 Risk Management Matrix

The most common waste materials procured in a workplace can be categorized in the following:

Liquid Waste

- Sludge, dirty water, organic liquids, waste water after washing.

Solid Waste

- Industrial slag, plastics waste, wood waste, paper waste, metals, and glass.

Organic Waste

- Biodegradable food waste, animal waste, vegetable waste, garden waste, rotten meat of animals can be deposited at Landfills or converted into Manure and Biogas.

Recyclable Waste

- Paper, metals, wood, organic waste etc. can be recycled.
- Must be placed in appropriate Recycling Bin and treated according to the nature of the waste.
- For example, organic waste can be converted into manure and Biogas.

Hazardous Waste

- Such waste may be flammable, corrosive, radioactive, toxic etc.
- These can potentially harm the environment and must be placed in clearly and legibly labelled bins for appropriate treatment and disposal.

Fig 13.1.3 Waste Segregation and Disposal Bins

Hazards and potential risks / threats can be identified and then reported to supervisors or other authorized persons in the following ways:

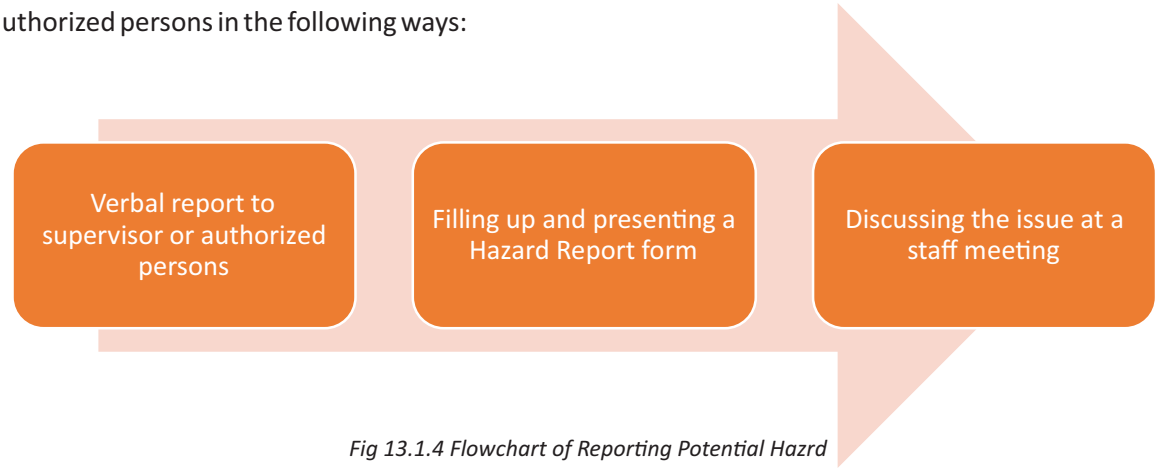


Fig 13.1.4 Flowchart of Reporting Potential Hazrd

Identification of hazard implies the job is half done. In order to take adequate precautionary measures against hazards, one needs to identify the hazards commonly found in the workplace. The common methods of hazard identification are:

Job Hazard Analysis (JHA)

- This is a popular technique to identify the perils associated with specific tasks in a job role, in order to lessen the risk of injuries to employees.
- The steps involved in successfully conducting JHA are:

A. Divide the entire job role into small tasks or steps

Let us understand the concept with the help of an example, where JHA is being conducted on corporate work such as Technical Support Engineer.

Steps	Hazards Associated	Recommendations
1. Handling tools and equipment of the trade		
2. Working with common electrical appliances of the workspace		
3. Stress factor of the job role		

Table 13.1.3 JHA Checklist for Hazard Identification

A. Spot out the hazards associated with each step by asking questions like:

- What can go wrong with this task?
- What would be the consequences if the task went wrong?
- How could the task go wrong?
- What are the other contributing factors?
- What are the chances that this hazard will take place?

B. Review and discuss the scope of the hazards with the employees, who would actually do the tasks on hand

C. Find out strategies and ways to mitigate or avoid the hazards

E. Review and revise the JHA periodically

Hazard and Operability (HAZOP) Study

- This technique involves a structured and systematic examination of an existing method / procedure, thus, in turn, identifying and assessing the associated hazards.
- These hazards can be easily identified in the form of Deviations in the process parameters (physical conditions and elements like flow, pressure, temperature, humidity, etc.
- The severity of Deviation can be illustrated with the help of specific and predetermined Guide Words.
- A Deviation is a manner in which the process conditions stray away from the expected values.

The steps involved in conducting HAZOP are:

- Segregating the entire system or process into sections or components
- Select a study node or point
- Define the expected outcome or consequence

- Choose a process parameter, based on the expected consequence
- Implement a suitable Guide Word
- Determine the Cause behind the deviation
- Start with the cause that may lead to the worst possible consequence
- Assess the deviations thus detected
- Devise and prescribe action
- Record and document information
- Repeat the process from B

Guide Word + Process Condition / Parameter = Deviation.

For example, No + Signal = No Signal

Common examples of Guide Words and their meanings are:

Guide Word	Meaning	Example
No (Not, None)	None of the desired consequence is achieved	No flow of gas through the gas cutting nozzle due to accumulated dirt
More (Higher than, More of)	Quantitative increase in a certain process parameter	More heat generated and higher temperature achieved than expected, during sawing operations
Less (Lesser than, Less of)	Quantitative reduction in a certain process parameter	Lower pressure than expected
As well as (In addition to)	All the design intentions are achieved and an additional activity takes place	All valves closed at the same time
Reverse	The logical opposite of the design intention takes place	The Power Drill continues drilling even after shutting down the power supply
Other Than	An unexpected activity takes place	Presence of liquid fuel in Gas Cylinder

Table 13.1.4 Guide Words and their Interpretation

The 3 Rs of Waste Optimization

- **Resource Optimization:** Raw materials must be used to the fullest, so that minimal waste is procured while converting the raw materials into finished products.
- **Recycling of Scrap Material:** Scraps, when created, must immediately be incorporated in the manufacturing process, so that they get reused completely as raw material.
- **Enhanced Quality Control:** This can be implemented by minimizing the number of rejects per batch. This is easily achievable with a higher frequency of careful inspection, accompanied with constant monitoring.
- **Exchange of Waste:** ome wastes cannot be completely eliminated from the manufacturing process. Such waste can be effectively managed via Waste Exchange techniques, where the waste procured in a certain process becomes the raw material of another, and vice versa.

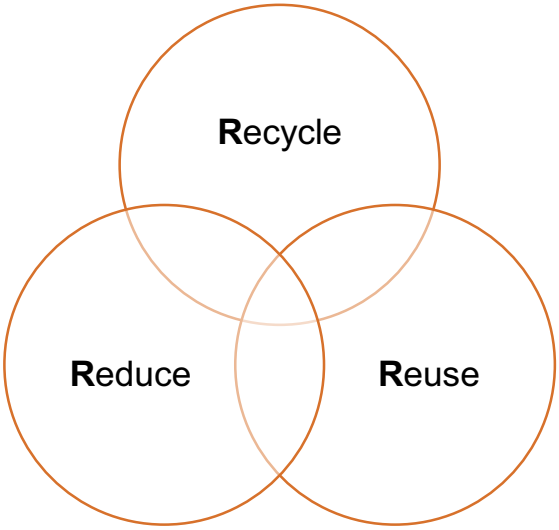


Fig 13.1.5 Rs of Waste Optimization

Landfill

- Waste, that cannot be recycled, is deposited and a layer of soil is added on top of it

Incineration

- Involves controlled combustion of waste
- 90% volume of waste gets reduced and converted into incombustible, light-weight materials like ash, gases and heat
- Gases are released into the environment while the heat is utilized in power generation

Biogas Generation

- Organic waste are biodegradable and can be converted into Biogas in Biogas Plants, with the help of certain fungi and bacteria
- The residue, after generation of Biogas, is used as Manure

Manure Generation and Composting

- Organic waste are often left buried under soil beds
- They decompose into rich manure, full of nutrients and minerals

Vermicomposting

- Involves the degradation of organic waste into manure, with the help of worms
- The worms feed on the organic waste and convert them into manure

Notes

Scan the QR Code to watch the related videos

https://www.youtube.com/watch?v=wQ7zJYBuY74

Demonstrate how to optimize usage of electricity, energy, materials, and water in various tasks

UNIT 13.2: Respect Diversity and Strengthen Practices to Promote Equality

Unit Objectives

By the end of this unit, participants will be able to:

1. Explain the diversity policy of the organization.
2. Comply to PWD inclusive policies for an adaptable and equitable work environment.

13.2.1 Concept of Gender, Gender Equality and Gender Discrimination

Policies and procedures about gender inclusivity, equality and sustainability while working with colleagues

The Constitution of India applies uniformly to equality of opportunity for all citizens (including every legal citizen of India, whether they are the disabled) in matters relating to employment or healthy or disabled. Under the Constitution the appointment to any office under the State. As a matter of fact, the employees of an organization constitute of major diversity. They come from different region, with different cultural and religious beliefs. However, the employer should provide equal opportunity to each and every employee, irrespective of gender, culture, religion. Particularly, the Indian Government has taken several measurements to ensure gender equality in the workplace. To establish women's right in the workplace, the government has passed bills. The Sexual Harassment of Women at Workplace (Prevention, Prohibition, and Redressal) Act, 2013 is an Indian law that aims to protect women against sexual harassment in the workplace. On September 3, 2012, it was approved by the Lok Sabha (the lower chamber of the Indian Parliament). The Rajya Sabha (the upper house of the Indian Parliament) approved it on 26 February 2013.

The major features of the policy include:

- The Act defines sexual harassment in the workplace and establishes a complaint resolution process. It also gives protections against fraudulent or misleading charges.
- The Act also includes 'quid pro quo harassment' and 'hostile work environment' as kinds of sexual harassment when they occur in conjunction with an act or behaviour of sexual harassment.
- The Act's expansive definition of "aggrieved woman" encompasses all women, regardless of their age or job level, whether in the organised or unorganised sectors, public or private, and also includes clients, consumers, and domestic workers.

- Section 2 defines an employer as any person responsible for the management, supervision, and control of the workplace, including those who establish and administer the organization's policies (g).
- While the "workplace" in the Vishaka Guidelines is limited to the traditional office setting where there is a clear employer-employee relationship, the Act includes organisations, departments, offices, branch units, etc. in the public and private sectors, organised and unorganised, hospitals, nursing homes, educational institutions, sports institutes, stadiums, sports complexes, and any place visited by the employee in the course of employment. This regulation will apply to non-traditional workplaces that entail telecommuting as well.
- The Committee is required to conclude its investigation within ninety days. The report will be given to the employer or the District Officer, depending on the circumstances, and they are required to take action within sixty days.
- Employers must establish an Internal Complaints Committee in each office or branch with 10 or more employees. The District Officer must form a Local Complaints Committee in each district, and at the block level if necessary.
- The Complaints Committees have the same authority to acquire evidence as civil courts.
- The Complaints Committees are supposed to provide conciliation prior to commencing an investigation if the complainant so requests.
- The investigation procedure under the Act must be kept confidential, and anyone who violates confidentiality is subject to a Rs 5,000 fine.
- Among other requirements, the Act compels employers to conduct education and sensitization initiatives and adopt policies against sexual harassment. The objective of Awareness Building can be attained via Banners and Posters displayed in the building, eLearning courses for employees, managers, and internal committee members, classroom training sessions, and email, eLearning, or classroom training for communicating the organization's sexual harassment policy. It is advised that eLearning or Classroom Training be provided in the employee's primary language of communication.
- Employers are now subject to penalties. Noncompliance with the Act's requirements is penalised by a fine of up to Rs. 50,000/-. Repeated infractions may result in harsher penalties and the revocation of a company licence or deregistration.
- The government can order an official to check the workplace and sexual harassment-related records of any organisation.
- To investigate any complaints made under the Act, which also applies to students in schools and colleges and hospital patients, employers and local governments would be compelled to set up grievance panels. Employers who refuse to comply can be fined up to 50,000 Indian Rupees.

13.2.2 Organization's Redressal Mechanisms

Inclusive tools and practices of communication to acknowledge/validate, share and promote the cause of gender parity at workplace

Women's safety and its related topics are addressed and debated globally. The number of sexual harassment reports continues to rise at an alarming rate each year. Therefore, in order to protect the safety of its female employees, a particular business must provide for their needs.

So, a company must inform women about the various facilities that they are going to provide them. Some of the basic facilities include the following.

1. **Transportation facilities:**

   Transportation plays a huge role in ensuring women safety. Ensuring that the women will be accompanied by trusted drivers will help enhance women's safety. Be transparent about the security that you may provide during night trips. Every woman must be aware of the various safeguards that the company may provide.

2. **Reporting Abuse:**

   The management must be prompt in its decision making whenever there is a mishap. The ways of reporting abuse must be made clear to the woman to ensure speedy remedy.

3. **Maternity-related grievance:**

   Employers are required to notify women entering the workforce in writing and electronically about the maternity benefits provided under the Maternity Benefit Act.

   The law permits women to work from home during their maternity leave if the nature of their work permits it.

4. **CCTV Cameras**

   Ensure that every station is equipped with CCTV cameras which are nowadays the most vital component for investigating sexual harassment cases.

5. **Security Guards**

   Ensure that adequate amount of security guards are stationed at strategic places so that any threat to women's safety can be nullified. Ensure that the women are informed about the various places where the security guards are present.

6. **Women's Helpline:**

   Share a leaflet containing the Women's helpline number/s and other important contacts.

7. **Chain locks/latches**

   Provide women with chain locks and latches so that their luggage can be properly and securely kept and to avoid any form of theft.

8. **Smoke Detector:**

   Inform the women about the location of smoke detectors inside the premises.

   Providing these basic amenities will ensure that the women enjoy comfortable accommodation without any fear.

All forms of gender discrimination, violence and inequality

The Sexual Harassment of Women at Workplace (Prevention, Prohibition, and Redressal) Act, 2013 establishes a system for investigating and redressing accusations of sexual harassment against women in the workplace. It also gives protections against fraudulent or misleading charges.

The principal clauses of the Act impose the following obligations on employers in order to ensure a safe workplace for women:

- Display penal consequences of sexual harassment
- Organize workshops and sensitization programs
- Formulate an internal policy, charter, resolution, declaration
- Form an 'Internal Complaints Committee' (ICC) where the number of employees is more than ten
- Provide necessary facilities to the committees
- Secure attendance of witnesses/respondent
- Monitor timely submission of committee reports
- Assist the woman in pursuing a criminal case if she so chooses
- Maintain confidentiality of the inquiry process. The Act lays down a penalty of Rs 5,000 (US$68) on the person who has breached confidentiality
- With sexual harassment being a crime, employers are obligated to report offenses

To tackle the problem of sexual harassment at workplace, the Ministry of Corporate Affairs, through a notification dated July 31, 2018, amended the Companies (Accounts) Rules 2014. The notification makes it mandatory for private companies to disclose their compliance with the Act in their directors' annual report.

Furthermore, the Act places responsibility on the appropriate state government to notify the district officer for setting up a Local Complaints Committee (LCC).

HR managers are on the front lines when it comes to changing cultural attitudes about sexual harassment.

Below are some best practices that HRs can develop to ensure safe work environment for women:

- Update the official employee handbook that outlines the procedure that will take place when sexual harassment is being experienced at work. Include an unequivocal statement that sexual harassment will not be tolerated.
- Give out a clear, simple, and easy-to-understand description of what constitutes harassing behaviour or conduct, including examples of the types of behaviours that are considered harassing at the workplace.

- Implement training for all to include more focus on gender identity and sexual orientation, and emphasize gender neutrality regarding who may experience sexual harassment.
- Sensitize male employees and reinforce confidence among women to come forward and file complaints.
- Stay updated on employment law changes where their employees live or work. HRs must also utilize professional associations, legal counsel and online resources to ensure that the company is compliant and aware of existing and upcoming legislative changes related to employee rights.

Use Internal & External Communication to Colleagues

It is often said that one's behaviour is the mirror to one's character. Indeed, your behaviour speaks a lot about the kind of person you are. Your educational degrees hold little importance if you are not a well-mannered person. You need to conduct well in almost every situation whether you appear for a job interview or pursue post-graduate degree, at your workplace or while dealing with your clients, in your school/college or while attending parties. Even at your home in front of your relatives, it is your good behaviour which counts the most. But behavioural etiquette is something which cannot be forced on anyone, it has to be cultivated and nurtured within oneself.

Showing compliant behavioural etiquette towards women is very important.

What are the various instances where one can show such etiquette? Let's take a look:

- **Before entering the room:** You must always knock and ask for permission before entering. This is perhaps the most basic etiquette. You must ensure that the privacy of the woman is unharmed. So, knock and take verbal permission before entering a room.
- **Avoiding touch contact:** You must always ensure that you do not intrude on the customer's personal space. This is not only unprofessional but also unhygienic. So try your best to avoid touch contact. If absolutely necessary, ask for permission and then assist the customer.
- **Using Abusive languages or gestures:** This is the last thing a women/customer expects from you. Ensure that you never use any foul language in front of the customer. Ensure that you don't abuse your colleagues in front of the guests.

Women are empowered by society and the law; some essential rights that are universally applicable to both sexes, but specifically for women, are as follows:

- Rights as a woman to dignity and respect, which entails that no man of any age is permitted to make sexual approaches, tease, or harass a woman.
- No one has the right to make women feel uncomfortable in the workplace, at home, on the streets, at school, college, or at a social gathering.

- Rights to physical and mental security: No one has the right to use physical force, to torture women physically or psychologically, or to coerce women in any way, regardless of their relationship to the perpetrator.
- Complaint privilege: Women have the right to lodge a complaint when even the slightest of their rights are abused. Take counsel and follow the correct course of action in such situations, regardless of the individual's status as a superior, relative, or neighbourhood bully.
- Security rights as a woman employee according to Visakha rules for preventing sexual harassment in the workplace.
- Physical or emotional violence against women is not a woman's inevitable fate, as is sometimes asserted. Dominant behaviour is neither a person's right nor a woman's destiny; therefore, it is perfectly acceptable to complain about it.

A security procedure is a predetermined sequence of operations that accomplishes a certain security task or function. Typically, procedures are structured as a sequence of steps to be followed as a consistent and recurring strategy or cycle to achieve a desired outcome.

Once adopted, security procedures give a specified set of steps for performing the organization's security affairs, hence facilitating training, process auditing, and process improvement. Procedures give a starting point for adopting the consistency required to reduce variation in security procedures, hence enhancing organization-wide security control.

An employer must ensure that the employees feel safe at all times without being over threatened by the security procedures and related environment.

13.2.3 Comply to PWD Inclusive Policies

How to maintain and provide a conducive work environment that is free from any harassment; facilities and amenities to PWD

The Indian Government respects the equality and therefore no discrimination should be made on the ground of disability. The Constitution guarantees all people, including those with disabilities, the right to justice, freedom of thought, speech, belief, faith, and worship, equality of status and opportunity, and the development of brotherhood. No disabled person may be required to pay taxes for the promotion and maintenance of a specific religion or religious group. To enforce the same, the government has passed laws to protect disables and their right to equality. The laws pertaining to disables are as follows:

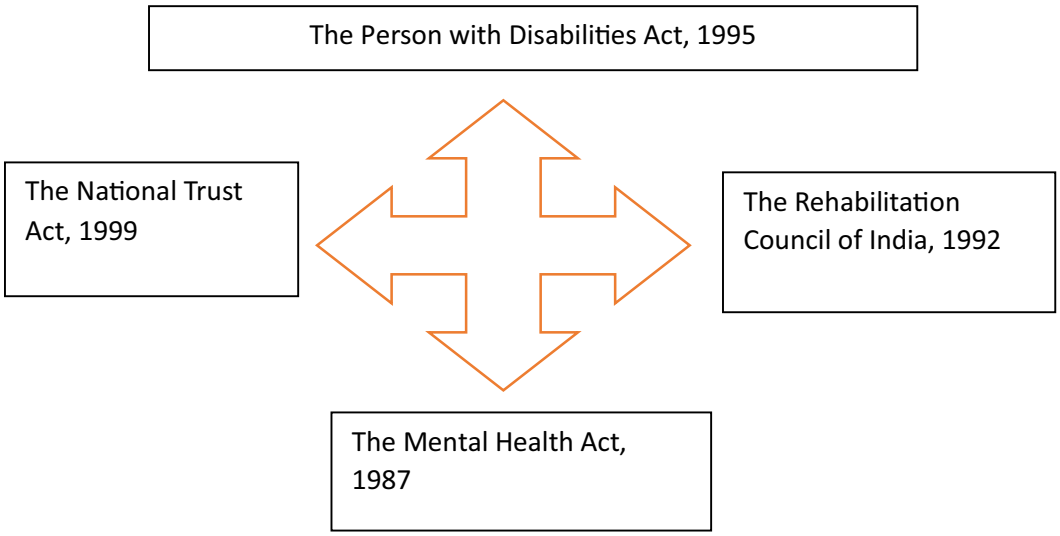


Fig 13.2.1 Acts Related to Disabilities

Improve through specifically designed recruitment practices, PWD friendly infrastructure, job roles, etc.

The 2016 Act expands the term of 'disabled person' to include persons with disability, persons with benchmark disability, and persons with disability and high support requirements. This inclusive concept classifies 21 categories of disabilities as "specific disabilities."

The Act is applicable to both government and private establishments. According to the law, a private establishment is a corporation, firm, cooperative or other society, associations, trust, agency, institution, organisation, union, or other government-designated establishment.

The Act mandates that all organisations develop and publish an Equal Opportunity Policy. All types of discrimination against those with disabilities are forbidden, unless it can be demonstrated that such discrimination is proportionate and essential for accomplishing legitimate ends.

The Act provides additional benefits for people with certain disabilities, including work openings in government agencies, educational opportunities, land distribution, and poverty alleviation programmes, among others.

To provide swift justice, special courts are established in each district to hear matters involving the infringement of the rights of disabled individuals. The maximum penalties for violating the rights of disabled people is $7,750 (Rs 500,000) and the maximum term of jail is five years.

Use and advocate for appropriate verbal/nonverbal communication, schemes and benefits of PWD

Although the majority of Act compliances apply only to government facilities, private establishments are also subject to the Act and must adhere to the following requirements:

- Frame and publish an Equal Opportunity Policy on the organization's website or in a prominent location inside the organization's premises. The Policy shall outline the accommodations and benefits made available to disabled employees. In addition, a copy of the Policy must be filed with the State Commissioner.
- Establishments with more than 20 employees must appoint a Liaison Officer to monitor the recruitment of handicapped individuals and the provision of specific facilities for them.
- Establishments must identify job openings that would be suitable for disabled candidates. In establishments receiving government incentives, a minimum of five percent of open positions must be reserved for disabled individuals.
- The employer must ban unlawful discrimination against disabled individuals in the workplace.
- The business must provide additional facilities or special advantages to disabled employees, such as special leave and training programmes, to boost their accessibility.
- All establishments must adhere to the government-issued accessibility standards for disabled individuals. The accessibility standards apply to infrastructure and communication technology in the workplace that must be accessible to impaired individuals.
- Every covered employer is required to maintain a record of its disabled personnel.

Summary

- Greenery within and around the office premises and other corporate environments helps not only to enhance the décor of the workplace, but also has a positive impact on the productivity of the employees.
- Plants in workplaces purify the air; they reduce the concentration of CO^2 (Carbon dioxide gas) and other volatile organic compounds, keeping the air fresh and healthy.
- External vegetation moderates heat in and around office block in the summertime, pulling down heat stress and decreasing the necessity for air-conditioning.
- Green roofs and facades proliferate insulation or the absorption capacity of heat, plummeting heating and cooling expenses.
- Plants in and around office buildings release water vapour which moistens the air, dipping headaches.
- Hazard is defined as a factor, which may cause harm to people and properties alike, like electricity, inflammable products, explosive material, corrosive chemical, using heavy ladders at workplace etc.

- In order to take adequate precautionary measures against hazards, one needs to identify the hazards commonly found in the workplace.
- The Constitution of India applies uniformly to equality of opportunity for all citizens (including every legal citizen of India, whether they are the disabled) in matters relating to employment or healthy or disabled.
- The Sexual Harassment of Women at Workplace (Prevention, Prohibition and Redressal) Act, 207 prescribes a system for investigating and redressing complaints against sexual harassment of women at the workplace.
- The definition of a 'disabled person' is broadened under the 2016 Act: it covers persons with disability, persons with benchmark disability, and persons with disability having high support needs.

Activitiy

Activity 1

Energy Conservation - Prepare a sample checklist and monitor

- This activity is in the form of "Prepare a sample checklist and monitor energy usage"
- This activity targets to make the trainees understand the optimization of energy in the workplace
- The trainer will divide the class into three groups
- The trainer will distinguish one particular room for the case study
- Each group will be assigned with the following tasks
    - o Count the number of lights, fans and ACs in the case study room
    - o Note down the duration of their usage
    - o Assess the proper usage and wastage
    - o Prepare a checklist to evaluate how to optimize the energy usage
    - o Submit a document furnishing observations
- The trainer will check the documents and declare the best group

Activity 2

Waste management

- This activity is in the form of "Waste management".
- The trainer will ask every trainee to prepare a sample hazard measurement checklist (as shown in Unit 7.1.2).
- The trainees should assess the waste management system of the building.
- They should prepare a document on the existing waste management system and propose systems to enhance it.
- They must be able to segregate between different types of waste and their treatment.
- On the merit of the document submitted by the trainees, the trainer will announce the best reports
- The trainees who furnished best reports will be appreciated by the class.

Exercise

A. Match the Followings:

Column A	Column B
The Sexual Harassment of Women at Workplace (Prevention, Prohibition and Redressal) Act	1995
The Person with Disabilities Act	1992
The Mental Health Act	1999
The Rehabilitation Council of India	2013
The National Trust Act	1987

B. Choose the Correct Answer from the Responses Given:

1. Which of the following options is incorrect?

   a). Greenery absorbs heat and keep the office building cool

   b). Greenery is mandatory as per the law enforced by government

   c). Greenery enhances productivity

2. IEA stands for -

   a). Indian Energy Agency

   b). Indian Energy Authority

   c). International Energy Agency

3. Employers who fail to comply will be punished with a fine of up to -

   a). INR 50,000

   b). INR 5,00,000

   c). Yet to determined

C. Answer the Following Questions

1. What are the basic steps of risk management?
2. Write down the key features of organization's redressal mechanism regarding women safety.
3. What are the common sources of renewable energy?
4. What could be the possible outcomes of violating PWD policies?
5. Write down the steps of saving a person from electrocution.

Notes

Annexure of QR Codes for Test Engineers

Chapter No.	Unit No.	Topic	Page No.	QR Code Links	QR Code (s)	Video Duration
Chapter 2: Concept and Principle of Quality Testing	Unit 2.1 - Concept and Principle of Quality Testing	QA Testing and Version Control	41	https://www.youtube.com/watch?v=LKUtzEHYK2I	QATesting and Version Control	0:02:15
Chapter 3: Design Tests for Software Products/Applications/Modules	Unit 3.1 - Design Tests for Software Products/ Applications/ Modules	Design Tests for Software Products/ Applications/ Modules	68	https://www.youtube.com/watch?v=RDBzZ-9XnDE	Design Tests for Software Products/Applications/ Modules	00:02:06
Chapter 5: Contribute to Quality Assurance of Projects	Unit 5.1 - Contribute to Quality Assurance of Projects	Discuss the scope of quality assurance for a Test Engineer	108	https://www.youtube.com/watch?v=TAjwPW2iULY	Discuss the scope of quality assurance for a Test Engineer	00:02:12
Chapter 6: Key Indicators for Software Applications	Unit 6.1 - Key Indicators for Software Applications	The purpose of impact indicator, efficiency indicator in testing	126	https://www.youtube.com/watch?v=I1uT7mhalio	The purpose of impact indicator, efficiency indicator in testing	00:02:17
Chapter 7: Technical Skills for Manual Tests	Unit 7.1 - Technical Skills for Handling Incidents	Latest Changes, procedures and practice in the field of designing test suite	153	https://www.youtube.com/watch?v=URmCWIL0IA0	How to store and retrive information	00:02:42
Chapter 9: Manage Your Work to Meet Requirements	Unit 9.2: Work Ethics to Follow in an Organization	Work Ethics to Follow in an Organization	192	https://www.youtube.com/watch?v=z_SeOWtpJVc	Work Ethics to Follow in an Organization	00:01:50
Chapter 10: Work Effectively with Colleagues	Unit 10.2 - Significance of Healthy Team Bonding in Ideal Work Culture	Importance and Understanding Problems from Your Colleague's Prospective and How to Provide Support	206	https://www.youtube.com/watch?v=R-YZonX_GSg	Importance and Understanding Problems from Your Colleague's Prospective and How to Provide Support	00:02:35

Chapter No.	Unit No.	Topic	Page No.	QR Code Links	QR Code (s)	Video Duration
Chapter 11: Maintain a Healthy, Safe and Secure Working Environment	Unit 11.2- Evacuation Procedures for Workers and Visitors	Evacuation Procedures for Workers and visitors	219	https://www.youtube.com/watch?v=2_5B7e5YWuM	Evacuation Procedures for Workers and visitors	00:02:02
	Unit 11.4- Health, Safety, and Accident Reporting Procedures and the Importance of These	Method of Health, Safety, and Accident Reporting	233	https://www.youtube.com/watch?v=bsafBtbJmrk	Method of Health, Safety, and Accident Reporting	00:02:02
Chapter 12: Workplace Data Management	Unit 12.2 - Follow the Accurate Process Flow to Analyse Data	Rule Based Analysis	250	https://www.youtube.com/watch?v=zFmNhF1W3Bk	Rule Based Analysis	00:02:15
Chapter 13: Maintain an Inclusive, Environmentally Sustainable Workplace	Unit 13.1 - Sustainable Practices	Demonstrate how to optimize usage of electricity, energy, materials, and water in various tasks	273	https://www.youtube.com/watch?v=wQ7zJYBuY74	Demonstrate how to optimize usage of electricity, energy, materials, and water in various tasks	0:02:27

Notes

Skill India
कौशल भारत-कुशल भारत

सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

Scan this QR Code to access eBook

https://eskillindia.org/Home/handbook/1430

IT - ITeS SSC
NASSCOM

IT – ITeS Sector Skill Council NASSCOM

Sector Skill Council Contact Details:

Address: Plot No. – 7, 8, 9 & 10 Sector – 126, Noida, Uttar Pradesh – 201303

New Delhi – 110049

Website: www.sscnasscom.com

Phone: 0120 4990111 – 0120 4990172

Price: ₹