

CHAPTER 3: PERFORM COMPUTER SOFTWARE INSTALLATION

Unit of Learning Code: IT/CU/ICT/CR/2/5

Related Unit of Competency in Occupational Standard: Perform Installation of Computer Software

1.1 Introduction to the Unit of Learning

This unit specifies the competencies required in installing computer software.

It involves identification of the software to be installed, install Computer software, manage configuration of the software, test software, and undertake user training.

1.2 Summary of Learning Outcomes

1. Identify software to be installed.
2. Install the software.
3. Software configuration management
4. Test software functionality
5. Perform user training.

1.2.1 Learning Outcome 1: Identify software to be installed.

1.2.1.1 Introduction to the Learning Outcome

The trainee should be able to define terms used in computer software installation, classify software according to their functionality, discuss criteria for software selection, establish acquisition methods of the software, discuss operating systems, their function, Types of operating system interfaces and selection of software to be installed based on usage and system requirements.

1.2.1.2 Performance standard

- 1.2.1.2.1 Software is classified according to the functionality, resource requirement and use.
- 1.2.1.2.2 Selection of software to be installed is identified based on usage and system requirements.
- 1.2.1.2.3 Acquisition methods of the selected software are established.

3.2.1.3 Information Sheet

Definition of Terms

Software - Is a set of programs, which is designed to perform a well-defined function hence tells a computer what to do or how to perform a task as well as accomplishing the specific user's task

A program - Is a sequence of instructions written to solve a particular problem.

Installation - Refers to setting up of a computer program hence making the program ready for execution.

Program execution - Execution is the process by which a computer interprets our program or performs the instructions of our program.

Software Acquisition – Includes processes that fund, manage, integrate, deploy and support software before, during and after their software engineering life cycle.

Operating Systems - This is a set of complex programs that work together to control execution of user programs called applications and acts as a go between (interfaces) between the applications of the computer hardware.

Human Computer Interface - Refers to the method of interaction between the computer and the user and determines how easily the user can operate the computer.

Security - Refers to providing a protection system to computer system resources such as CPU, memory, disk, software programs and most importantly data/information stored in the computer system.

Classification of Computer Software

Computer software are classified using three approaches:

- a. Classification according to purpose/functionality
- b. Classification according to method of acquisition
- c. Classification according to end-user license

A. Classification According to Purpose/Functionality

Computer software is classified according to the task they perform/functionality/purpose. i.e., software that tells the computer hardware what to do and also the software that enables the user accomplish their specific tasks. This software includes:

i. System software

System software performs a variety of fundamental operations that avails computer resources to the user. These functions include:

- i. Booting the computer and making sure that all the hardware elements are working properly.
- ii. Performing operations such as retrieving, loading, executing and storing application programs.
- iii. Storing and retrieving files.
- iv. Performing a variety of system utility functions.

System software is further divided into:

- a. Operating system.
- b. Firmware.
- c. Utility software.
- d. Networking software.

a. The Operating System

This is a set of complex programs that work together to control execution of user programs called applications and acts as a go between (interfaces) between the applications of the computer hardware.

It manages input/output and storage operations in a computer. Examples of common operating systems are Microsoft Windows 7/8/8.1/10, UNIX, Linux, Macintosh (Mac OS) and OS/2.

Therefore the operating system is the main program on the computer system.

b. Firmware

Firmware, also referred to as **stored logic** is a combination of both the software and hardware recorded permanently on electronic chips. Usually, a firmware is a read-only memory chip that is mounted or plugged into the motherboard. Firmware may hold an operating system, utility programs, language processors etc.

c. Utility software

Utility software is a special program that performs commonly used services that make certain aspects of computing to go on more smoothly. Such services include **sorting, copying, file handling, disk management** etc. The two basic types of utility software are:

- i. **System-level utility software:** These help the user to work with the operating system and its functions. For example, a utility software tells the user when he/she enters a wrong command and gives suggestions how the error can be corrected.
- ii. **Application utility software:** These make the use of an application program smoother and efficient. These utility programs are commonly purchased separately or may be part of an operating system.

d. Networking software

This type of software is mostly used to establish communication between two or more computers by linking them using a communication channel like cables to create a computer network. Networking software enables the exchange of data in a network as well as providing data security. Network software may come as independent software or integrated in an operating system. An example of networking software is Cisco's iOS, Microsoft Windows Server 2003/2008/2012/2016, UNIX, Linux, Mac OS X and Novell NetWare.

ii. Application Software

Application software, also called **application packages**, are programs that are designed to help the user accomplish specific tasks.

Examples and uses of common application packages are listed in Table 3 below:

Software	Uses	Examples
1. Word Processors	Typing documents like letters.	Ms Word, Lotus, WordPro, WordStar.
2. Desktop publishing	Designing publications like newspapers, books.	Adobe PageMaker, Ms publisher
3. Computer aided design	Technical drawing.	AutoCAD, ArchiCAD, MAYA 3D etc
4. Databases	Keeping records and files.	Ms Access, dBase.
5. Graphics software	Creating and manipulating pictures.	Corel Draw, Adobe

		Photoshop/Illustrator/ Sketchup etc
--	--	--

Table 1 common application package

iii. **Programming Languages/Computer Programming Tools**

Computer programming tools, such as compilers and linker, are used to translate and combine computer program source code and libraries into executable RAMs (programs that will belong to one of the three said).

Criteria for Selection of Computer Software

Although one may have a good computer with the best hardware, the actual real determinant of a computer's value to the user is the software in that it can run to solve the data and information processing needs.

The following factors/criteria should be considered when selecting software:

a) Authenticity

The term authenticity refers to genuineness, validity and or legitimacy of an item. When you acquire software from the vendor, make sure it is an original copy that is accompanied by the developer's license and certificate of authenticity. This is because some people illegally produce pirated copies which is an offence.

b) Documentation

It refers to the manuals prepared by the developer having details on how to install, use and maintain the software. These include installation guides, maintenance guides and user guides. These documentations enable the user to work with the software with minimum guidance.

c) User needs

The needs of the user determine the type of operating system and application programs that should be considered for acquisition. For example, if the user needs to type documents most often, he/she would go for a word-processor.

People with special disability will require software that recognizes other forms of input like voice and natural sound. A good example is software used in mobile phones to store voice and allow the user to make a call by just calling a name instead of keying in the number.

d) Reliability and security

People are more comfortable with software that offers good security to confidential and private information. Reliability refers to the guarantee that a software will perform with the same consistency over the period which the owner will be using it with minimal or completely no errors.

e) User friendliness

This is a measure of how easily the users can be able to operate the computer. Some programs are more user- friendlier than others.

f) Cost

The cost of software is perhaps one of the most controversial issues that must be considered carefully. One cannot just go for software because it is cheap. Many other factors may force a person to buy far much more expensive software even with cheaper alternatives available. However, it is illogical to buy expensive software if there is a reliable cheaper alternative that will meet one's needs.

g) Compatibility and System Configuration

Software compatibility refers to the ability of the computer to run the software depending on the system setup. For example, some software may only run on a computer that has 4GB of RAM and above so that they will be held temporarily as they are being executed.

h) Portability

Portability in this aspect refers to whether a program can be copied or installed in more than one computer.

Software Acquisition Methods

They include:

- a. Off-the-Shelf/Standard Software
- b. In-House Developed Programs
- c. Open Source/ Freeware

a. Off-the-Shelf/Standard Software

These programs are developed by software engineers, packaged and then made available for purchase through a vendor, a distributor or directly from the developer. Examples of suites are Lotus Suite, Microsoft-Office, Adobe Suite, Corel WordPerfect etc. while those of single purpose are QuickBooks and Sage Line 50.

The word package is sometimes used to refer to these types of software that are packaged and sold by vendors.

The Advantages of Off the Shelf Software

- i. The initial cost of acquisition is cheaper compared to in-house developed software.
- ii. They are readily available in the market.
- iii. They are more reliable since they have tested and reviewed intensively.
- iv. They offer a wide variety of functionality.
- v. They receive automatic and regular updates.
- vi. They offer after-sales service support.
- vii. They are time and cost-saving in comparison to in-house developed software.

The Disadvantages of Off the Shelf Software

- i. Can be more expensive in the long-run due to licensing and subscription fees.
- ii. They cannot easily be customised to the owner's needs.
- iii. They have limited functionality and cannot easily be modified.
- iv. Some have an end-of-life support.

- v. May attract extra costs for the upgrades.
- vi. May offer unnecessary features and functionalities.
- vii. May have integration issues.

b. In-House Developed Programs

These are programs that are uniquely designed and tailored to meet a particular user's needs. For example, a bank may decide to manage its banking operations using a unique program developed by hired programmers. These programs are not available in the shops and once developed for one company or user may not address the needs of other users.

Advantages of In-House Developed Programs

- i. They are easily customisable to the customer's needs.
- ii. They offer services and functionalities that precisely fit the owner's needs.

Disadvantages of In-House Developed Programs

- i. They are costly to develop and implement.
- ii. They are prone to bugs and errors.
- iii. They require special training for their proper use.
- iv. They may not be compatible with other off-the-shelf software.

iii. Open Source/Freeware

A software for which the original source code is made freely available and may be redistributed and modified according to the requirement of the user. It is free to download.

Read more on open-source software by following this link:
https://en.wikipedia.org/wiki/Software_categories

Advantages of Open Source/Freeware

- i. Cheaper than commercially marketed products.
- ii. Created by skillful and talented people.
- iii. Highly reliable.
- iv. Help you become more flexible.

Disadvantage of Open Source/Freeware

- i. Vulnerable to malicious users.
- ii. Might not be as user-friendly as commercial versions.
- iii. Don't come with extensive support.

Operating Systems

An OS is the system software that provides a platform for other programs to run on in a computer. It's the program that controls/interacts with the hardware of the computer and controls other application software.

Factors to Consider When Choosing An Operating System

When choosing an operating system for a computer, the following factors should be considered:

1. Hardware configuration or provision of the computer e.g., memory size, hard disk capacity, type of processor, processor's speed, display requirements etc.
2. Basic design of the computer e.g., is it an IBM or IBM-Compatible, or an Apple computer.
3. Applications intended for the computer – consider the architecture of the OS i.e., 32-bit or 64-bit.
4. User friendliness or human computer interface i.e., is it command line based, menu driven, or graphical user interface based?
5. Availability in the market e.g., Microsoft Windows based operating systems are very common.
6. Cost - how affordable is the operating system?
7. Reliability i.e., can it run without crashing or hanging i.e., stop responding to commands.
8. User support- what are the terms of after-acquisition support for the OS?

Types of Operating Systems

They include:

- i. Single and multi-user
- ii. Single and multitasking
- iii. Traditional OS or Batch mode OS
- iv. Multiprocessor OS or Time-Shared Operating System
- v. Real time
- vi. Distributed

i. Single and Multi-User

a. Single User Operating System

A single user operating system is designed for use by only one person. It cannot support access to more than one person at a time. e.g., Windows 10 or Ubuntu 20.04 installed on a personal computer.

b. Multiuser Operating System

Multiuser or multi access operating system allows more than one user to interactively use the computer. It can be installed on a computer that is accessed by many people at the same time. Examples of such operating systems are UNIX Solaris, Windows Server 2016, Linux Ubuntu Server 20.04.

NB: When categorising OSs as single user or multi-user, the key parameter used is the number of users who can concurrently access the services of the OS and not how processes or programs it can run concurrently.

ii. Single and Multitasking Oss

a. Single Program/Single Tasking OS

Single program operating system allows processing of only one user program in the main memory at a time. This means that the user can only run one interactive program at a time. Then the user

must exit from the program before loading and running another program. An example of a single user operating system is MS DOS from Microsoft Corporation.

b. Multi-Tasking OS

This type of operating system allows a single CPU to execute what appears to be more than one program at the same time. However, internally only one program is being executed at a time. The CPU switches its attention between programs as it receives requests for processing, executing statements from one program, and then from another using the concept of giving a time slice to each application. This switching of attention is so fast that it appears as if the programs are being executed simultaneously.

iii. Traditional OS or Batch Mode OS

Historically they were the first OS. Refers to a single processor i.e., OS that control a single processor (Uniprocessor) system which are centralized. Usually allow one job to run at a time e.g., the OS for the 2nd generation.

In such systems the program, data & commands to manipulate the program & data are all submitted together to the computer in form of a job. A job must be completed for the next job to be loaded & reprocessed. The improvement of such OS is the implementation of batch systems to reduce on time wasted when the CPU is idle.

In the batch OS, several jobs are collected & processed until processing is aerial or sequential (One job batch after the other.). Consequently, they support one user at a time. There is little or no interaction between the user and the executing program (OS is less user friendly to the user)

They do not support highly interactive computing with graphics, or point & draw devices e.g., a mouse.

iv. Multiprocessor O/S or Time-Shared OS

Refers to OS that controls several processes that have a common access to the main memory storage.

This type of operating system allows several processes to share the processor creating an illusion that the processes are running simultaneously by shifting execution rapidly among all the active processes.

This iteration of various processor with their inter-control is necessary & it is done in the multi-processor OS.

v. Real Time OS

This type of operating system is designed to support execution of tasks within specific wall clock time constraints. The operating system must guarantee that the task can be executed within a specified time constraint. This type of operating system is usually limited to dedicated applications such as industrial control systems, weapon systems and computer-controlled products.

Distributed OS

Refers to a common OS shared by a network of computers. It provides support for inter- process migration, multi-execution & prevention of deleting deadlock. It appears to user as a uniprocessor system, even though composed of microprocessors. The users should not be aware of where their programs are being run or where their files are located. This should all be handled automatically & efficiently by the OS e.g., when a user starts a program, it may run on a local machine, but if the computer is heavily loaded & OS knows that another computer is idle, then the job may be transferred to that machine.

Functions of Operating Systems

The OS performs various functions which include:

a. Storage Management/Memory management.

All data and instructions must be temporarily held in the main memory before and after processing. The operating system may organize the main memory into blocks of sizes called partitions. It constantly assigns main memory storage partitions to data and instructions. To access a piece of data or instruction, the operating system knows where to find each piece of data if the correct address of the partition is used.

The OS ensures that each program and the data it requires are allocated adequate space in memory.

b. Process Control/ Job scheduling

The processor can only handle one task at a time. Therefore, the operating system has to determine which task will be processed first and makes sure that the one that is currently being processed is closely monitored to avoid wasting time in the processor. The criteria for selecting which task will come before the other depends on many factors. For example, the operating system may decide to process smaller tasks before larger ones.

c. Security Management

It helps in implementing mechanisms that secure and protect the computer system internally as well as externally. Therefore an operating system is responsible for securing the system at two different levels which are internal security and external security.

i. Internal security

Internal security refers to the protection of activities of one process from the activities of another process. The term internal security may also be regarded as system protection. The internal security of the computer system also ensures the reliability of the computer system. There may be several process running in the computer systems use the concept of least privilege to implement internal security.

ii. External security

External security refers to the implementation of a mechanism for securing the data and programs stored in the computer system as well as the various resources of the computer system against unauthorized access. The term external security may also be regarded as system security. External security is particularly required when a computer system is either on a network or connected to the internet.

d. Device Management/ Input/output Handling

Every computer has many input and output devices (I/O). Like a skilled traffic officer, the operating system coordinates between these various I/O and other peripheral devices such as auxiliary storage devices, making sure that data flows properly between them and sorting out any possible confusion. For example, when printing, the CPU directs its attention to the printing function. The operating system searches for the printer, chooses the correct one, translates the name for the CPU and finally the CPU sends the document to the printer. This then makes the CPU available for other activities.

e. Resource Control and Allocation

The OS gives each resource a unique identification number called an interrupt number in order for the processor to be able to recognize and prioritise the use and requests for resources. Hence when two tasks request to use a resource at the same time, the one with higher priority interrupt is granted control. This can be compared to the priority given to the presidential motorcade on a busy road.

Secondly, tries as much as possible to avoid a situation where a particular task holds a needed resource and refuses to release it for use by other tasks. When several tasks do this, an undesirable situation called deadlock occurs.

A deadlock is as situation which leads to one process holding onto the use of CPU without release and thus denying any other process access to the same resource leading to the computer device 'hanging'.

Therefore, resource control and allocation are core operating system functions because they determine which tasks uses a particular resource and at what time.

f. Error Handling

The operating system has many ways of alerting the user, of errors he or she makes. Many operating systems usually express what the error is, and where possible make suggestions on how to correct the error. The operating system does this by monitoring the status of the computer system and performing error checks on both hardware and the software.

g. Job Sequencing/ job scheduling

The operating system keeps a list of jobs or tasks currently being run and clocks them in and out of the processor. It also arranges them in a particular order to make it easy for the processor to execute them and to know how and when to fetch instructions and data for each task.

h. Interrupt Handling

An interrupt is a break from the normal sequential processing of instructions in a program. An external request causes the processor to stop executing the current task and do something else before returning the control back to the program that was interrupted.

Each hardware device communicates to the processor using a special number called the interrupt request number (IRQ number).

Types of Operating System Interfaces

The underlying principle in operating system design is to make complex tasks very simple for the user to carry out. Follow the link below to learn more on types of operating system interfaces: <https://www.youtube.com/watch?v=2HfvAbLq9rE>

Currently the three main types of human computer interface are:

a) Command Line Interface (CLI)

The user interacts with a computer by typing a command at the prompt found on a command line. A computer reads instructions from the command line and executes them. For a command to be more user friendly, the words used should be descriptive verbs e.g., print, copy etc. Unique abbreviations can also be used e.g., Del, Ren, Chkdsk etc. For example, if you are using MS DOS operating system, you can copy a file called Kimathi.Dat from a hard disk C to partition D as follows: COPY C:\Kimathi.Dat D:\

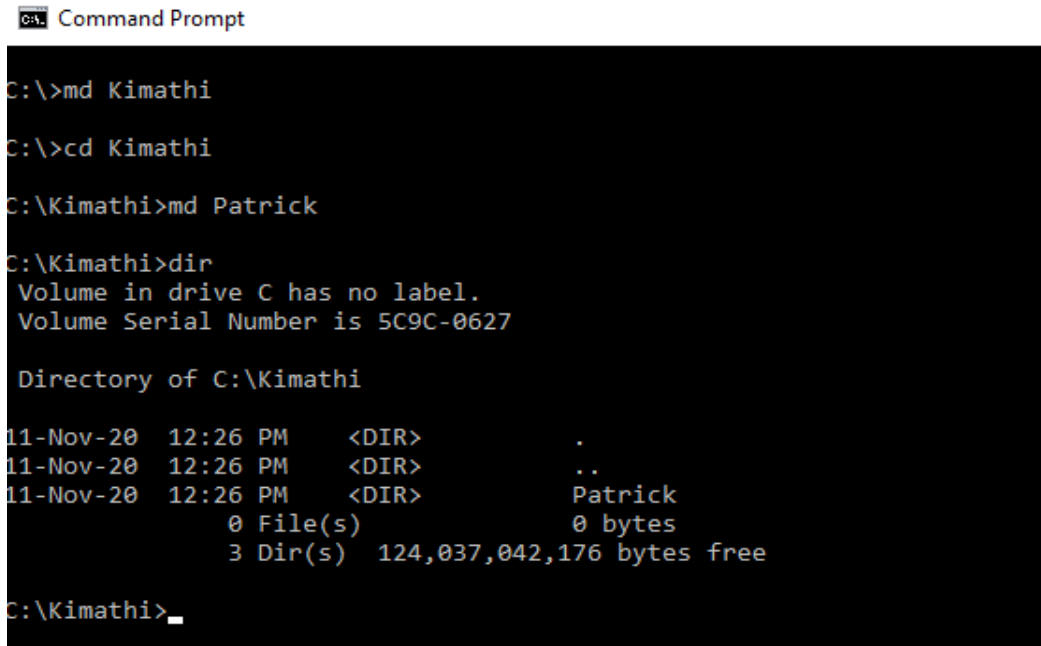
Examples of command line interface are the early versions of MS DOS, PC DOS, Linux Bash, OS/2, and UNIX.

NB: The user must press the enter key for the command to be executed.

Follow the link to learn more about DOS Commands:

<https://youtu.be/syrb1Fk3m2A>

The figure below shows the MS DOS command line interface.



```
C:\>md Kimathi
C:\>cd Kimathi
C:\Kimathi>md Patrick
C:\Kimathi>dir
Volume in drive C has no label.
Volume Serial Number is 5C9C-0627

Directory of C:\Kimathi

11-Nov-20 12:26 PM    <DIR>        .
11-Nov-20 12:26 PM    <DIR>        ..
11-Nov-20 12:26 PM    <DIR>        Patrick
                0 File(s)                0 bytes
                3 Dir(s)  124,037,042,176 bytes free

C:\Kimathi>
```

Figure 1: MS DOS COMMAND LINE

b) Menu Driven Interface

This type of interface provides the user with a list of options to choose from. The interface therefore is suitable for beginners who may have difficulties recalling commands. The figure below shows the Menu driven interface.

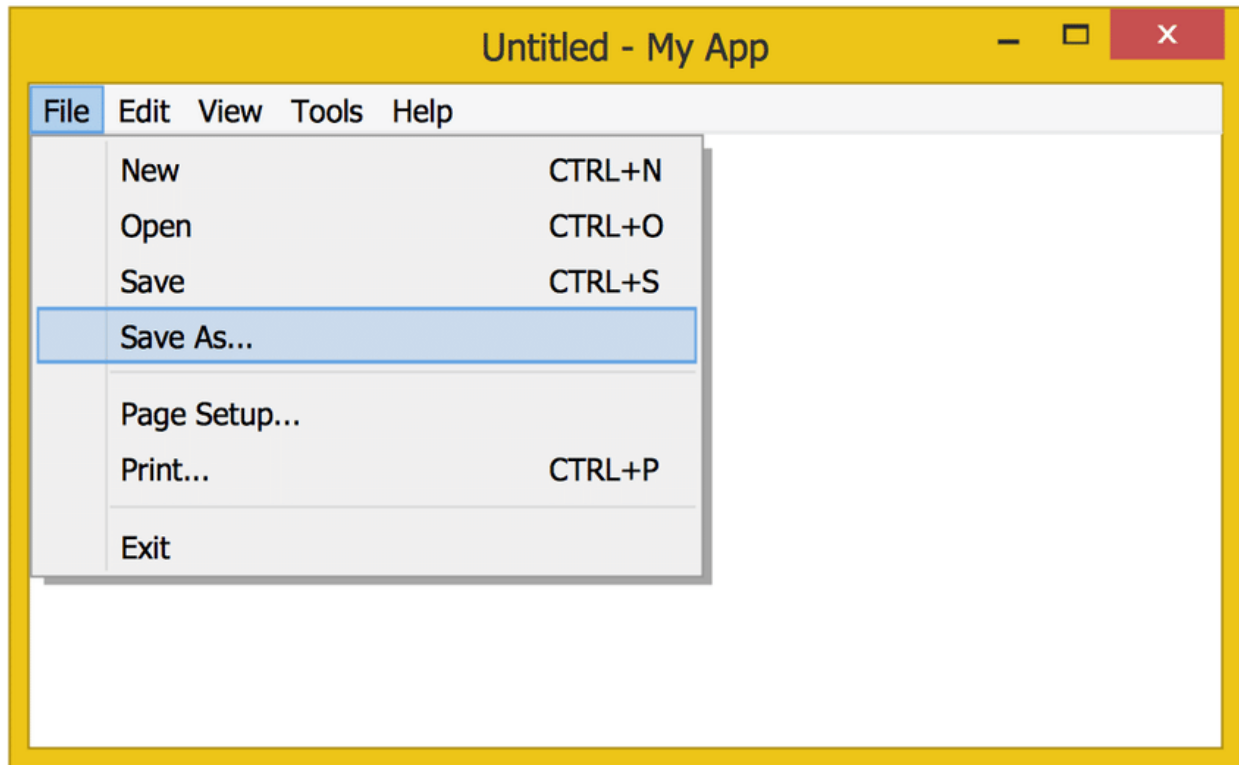


Figure 2:Menu Driven Interface

c) The Graphical User Interface (GUI)

This type of interface represents commands as small pictures on the screen called icons. Icons can be selected to issue a command using a pointing device like a mouse. It is user friendliness.

Examples of GUI based operating systems are OS/2 s Presentation Manager, Microsoft Windows, Linux and Apple Macintosh. A typical Windows QUI interface is shown in Figure 5 below:



Figure 3: Windows Graphical User Interface

Follow the link below to learn more on Graphical User Interface

https://www.youtube.com/watch?v=N52jNC8Mt_s&t=2s

3.2.1.4 Learning Activities

Project Work

1. Interact with each of the following Software:

- a) Ubuntu 20.04
- b) Windows 10
- c) MacOS 10.15 Catalina
- d) Microsoft Office 2016

From each software's user guide/user manual.

- i. Note down the key system requirements.
 - ii. Note down key similarities and differences among the listed software.
2. Using internet resources, research on the major operating systems used in:
- i. Smart phones
 - ii. Computer desktop provided.

Practical Activities

1. Using the MS DOS command interface, perform the following tasks:
 - i. Create a directory called Kimathi.
 - ii. Rename the directory Kimathi to Patrick.
 - iii. Create a text file named Jane.txt and save it in the directory named Patrick.
 - iv. Add the following line of text in the file Jane.txt "This is my first text file!"
 - v. Save the changes to the file.

- vi. Display list of contents of current directory
- vii. Change directory/partition D.
- viii. Checks a disk C and displays a status report.
- ix. Display and set the computer's time and date to 12:00:00 31/12/2020.
- x. Open any application program from the command line.

3.2.1.5 Self-Assessment

1. Discuss five factors that one should consider when choosing an Operating System to install to a computer. [10 Marks]
2. Which five requirements would you consider before installing an Operating System on a computer? [5 Marks]
3. Discuss the three software acquisition methods and select the best method to use when choosing software for drawing a house plan. [10 Marks]
- 4 Using the ms dos command interface, perform the following task.
 - i. Create a directory called Kimathi.
 - ii. Rename a directory to Patrick.
 - iii. Display list of contents of current directory.
 - iv. Change directory/partition to D.
 - v. Checks a disk D and displays a status report.
 - vi. Displays or sets the computer's time.
 - vii. Displays or sets the computer's date.

3.2.1.6 Tools, Equipment, Supplies and Materials

- 1 Utility programs e.g antivirus
- 2 Computer(s)
- 3 Application Software
- 4 External Hard disk
- 5 Flash disk
- 6 Software CD/DVD
- 7 Digital instructional material including DVDs and CDs;
- 8 Operating systems
- 9 Power
- 10 Internet connection

3.2.1.7 References

Geoffery Chemwa, Stephen Mburu. (2006). Computer Studies Form 1. Nairobi: Longhorn.

tutorialspoint. (2016). Operating System. Retrieved 11 11, 2020, from www.tutorialspoint.com:
https://www.tutorialspoint.com/operating_system/operating_system_tutorial.pdf

Graphical User Interface. (2019, Jan 5). Retrieved Nov 11, 2020, from you tube : https://www.youtube.com/watch?v=N52jNC8Mt_s&t=2s

DOS Commands basic. (2012, Oct 24). Retrieved Nov 11, 2020, from youtube.com: <https://www.youtube.com/watch?v=1dGwn7cPNeU>

OS , CLI Command Line Interface and Menu Driven Interface. (2020, Jun 25). Retrieved Nov 11, 2020, from youtube.com: <https://www.youtube.com/watch?v=2HfvAbLq9rE>

Jain, P. (2018, June 28). Device and Security Management in Operating System. Retrieved Nov 22, 2020, from includehelp.com: <https://www.includehelp.com/operating-systems/device-and-security-management.aspx>

Chief, E. i. (2015, Jun 28). *7 Main Advantages and Disadvantages of Open Source Software*. Retrieved Dec 24, 2020, from connectusfund.org: <https://connectusfund.org/7-main-advantages-and-disadvantages-of-open-source-software>

3.2.1.8 Model Answers to self-assessment

1. Factors that one should consider when choosing an Operating System to install to a computer.
 - i. **Hardware configuration**
 - ii. **User friendliness**
 - iii. **. Applications**
 - iv. **Basic design**
 - v. **Availability in the market**
 - vi. **Reliability**
 - vii. **Cost**
2. The requirements for a computer to be installed with an Operating System
 - i. **Processor: - 1 gigahertz (GHz) or faster processor or System on a Chip (SoC)**
 - ii. **RAM: - 1 gigabyte (GB) for 32-bit or 2 GB for 64-bit**
 - iii. **Hard drive space:- 16 GB for 32-bit OS 32 GB for 64-bit OS**
 - iv. **Graphics card:- DirectX 9 or later with WDDM 1.0 driver**
 - v. **Display:- 800x600**
3. Compare all the software acquisition methods and identify the best method to use, justifying your answer.
 - i. **Open source**
 - ii. **Off the shelf/standard software**
 - iii. **In-house developed programs**

Nb: justification depending on the advantages over the disadvantages.

4 Using the ms dos command interface, perform the following task.

- i. Create a directory called Kimathi - **mkdir Kimathi.**
- ii. Rename a directory to Patrick - **rename Kimathi Patrick.**
- iii. Display list of contents of current directory - **dir Patrick.**
- iv. Change directory/partition to D - **D:**
- v. Checks a disk D and displays a status report - **chkdsk.**
- vi. Displays or sets the computer's time - **time.**
- vii. Displays or sets the computer's date - **date**

3.2.2 Learning outcome 2 install the software.

3.2.2.1 Introduction to the learning outcome

The trainee should be able to explain terms used in software installation, define software installation, prepare installation media, describe software installation legal requirements, describe existing data protection, explain types of software installation (attended, unattended, headless, schedule/automated, clean/updating and network), describe software installation, registration perform software configuration and explain importance of registration

3.2.2.2. Performance Standard

- 1.2.2.2.1 Software specifications and computer resource requirements are identified.
- 1.2.2.2.2 Source of software installation files is determined.
- 1.2.2.2.3 Existing data is backed up.
- 1.2.2.2.4 User vendor agreements are identified.
- 1.2.2.2.5 Software installation is done as per the installation manual provided.

3.2.2.3 Information Sheet

Definition of terms

Software installation - refers to the process of setting up a software to work with the computer, printer, and other hardware.

Installation media – refers to a device like USB flash drive or DVD used to install a new copy of OS, perform a clean installation, or reinstall OS.

Software License Agreement - is an agreement between a company and it's customers for use of the software they have the rights to.

Data protection - is the process of safeguarding important information from corruption, compromise or loss.

Data security - is both the practice and the technology of protecting valuable and sensitive company and customer data, such as personal or financial information.

Data backup - entails making a copy of your data and storing it on a separate system or medium such as a tape, disk, or in the cloud.

An End-User License Agreement (EULA) is a legal contract entered into between a software developer or vendor and the user of the software, often where the software has been purchased by the user from an intermediary such as a retailer. A EULA specifies in detail the rights and restrictions which apply to the use of the software.

Product activation - is a license validation procedure required by some proprietary computer software programs. Product activation prevents unlimited free use of copied or replicated software.

A software wizard or setup assistant - is a user interface type that presents a user with a sequence of dialog boxes that lead the user through a series of well-defined steps. Tasks that are complex, infrequently performed, or unfamiliar may be easier to perform using a wizard.

How to create installation media

To create the installation media, the following requirements are required:

- i. A PC with a reliable internet connection.
- ii. A USB flash drive or DVD. A blank USB flash drive with at least 8 GB of space, or a blank DVD (and DVD burner
- iii. A product key. Your 25-character product key (not required for digital licenses).

Procedure

To create installation media, go to the software download website, where you'll find step-by-step instructions. On that website, you can select a version of Windows and create your own installation media using either a USB flash drive or a DVD. To go directly to one of the versions, select one of these hyperlinks:

[Windows 7](#)

[Windows 8.1](#)

[Windows 10 \(Then select Download tool now.\)](#)

Example using windows 10

Follow these steps to create installation media (USB flash drive or DVD) you can use to install a new copy of Windows 10, perform a clean installation, or reinstall Windows 10.

Before you download the tool make sure you have:

- An internet connection (internet service provider fees may apply).
- Sufficient data storage available on a computer, USB or external drive for the download.
- A **blank** USB flash drive with at least 8GB of space or blank DVD (and DVD burner) if you want to create media. We recommend using a blank USB or blank DVD, because any content on it will be deleted.
- When burning a DVD from an ISO file, if you are told the disc image file is too large you will need to use Dual Layer (DL) DVD Media.

Check a few things on the PC where you want to install Windows 10:

- **64-bit or 32-bit processor (CPU).** You'll create either a 64-bit or 32-bit version of Windows 10. To check this on your PC, go to **PC info** in PC settings or **System** in Control Panel, and look for **System type**.
- **System requirements.** Make sure the PC meets the system requirements for Windows 10. It is recommended to go to the PC manufacturer's website for additional info about updated drivers and hardware compatibility.

- **Language in Windows.** You'll need to choose the same language when you install Windows 10. To see what language you're currently using, go to **Time and language** in PC settings or **Region** in Control Panel.
- **Edition of Windows.** You should also choose the same edition of Windows. To check what edition you're currently running, go to **PC info** in PC settings or **System** in Control Panel, and look for **Windows edition**.
- **Microsoft Office products.** If you just purchased a new device that includes Office 365, it is recommend redeeming (installing) Office before upgrading to Windows 10. To redeem your copy of Office, please see Download and install Office 365 Home, Personal, or University on your PC.

If you have Office 2010 or earlier and choose to perform a clean install of Windows 10, you will need to locate your Office product key. For tips on locating your product key, check Find your Office 2010 product key or Enter the product key for your Office 2007 program.

Using the tool to create installation media:

1. Select **Download tool now** and select **Run**. You need to be an administrator to run this tool.
2. If you agree to the license terms, select **Accept**.
3. On the **What do you want to do?** page, select Create **installation media for another PC**, and then select **Next**.

Select the language, edition, and architecture (64-bit or 32-bit) for Windows 10.

4. Select which media you want to use:
 - **USB flash drive.** Attach a blank USB flash drive with at least 8GB of space. Any content on the flash drive will be deleted.
 - **ISO file.** Save an ISO file to your PC, which you can use to create a DVD. After the file is downloaded, you can go to location where the file is saved, or select **Open DVD burner**, and follow the instructions to burn the file to a DVD. For more info about using an ISO file, see Additional methods for using the ISO file to install Windows 10 section below.
5. After the installation media is created, follow the steps below to use it.
6. After completing the steps to install Windows 10, please check that you have all the necessary device drivers installed. To check for updates now, select the Start button, and then go to settings update & security windows update, and select check for updates.

NB: Click the Media link below to follow how to How to Create Windows 10 Installation
<https://www.youtube.com/watch?v=DGcWddE6c4c>

Software installation legal requirements

A software license agreement is something you want to have in place to prevent or protect you from infringement of [copyright law](#)

Reasons for software installation legal requirements:

- i. It prevents abuses of your software – If you fail to have one of these agreements signed by one of your customers, there is nothing stopping them from trying to replicate it or copy it for their own benefit.
- ii. It allows you to license it and not sell it – When you allow users to purchase a license of your software, you still retain all rights to it. This allows you to license it to others and also allows you to place restrictions on its use.
- iii. It allows you to disclaim warranties – No matter what you do, a customer who installs your software will have expectations that cannot be met all the time. This can be something such as a guarantee of no software bugs, no downtime, or other such expectations. You can include terms in your software license agreement that will include a disclaimer of warranties which will require the user to accept it as is or as available.
- iv. It can limit your liability – you could potentially be exposing yourself to lawsuits. These lawsuits not only take up your valuable time, but they also can create some financial issues for you.
- v. It can allow you to terminate use at any time with no problems – There should be a portion of the agreement that states that you can revoke licenses at any time.

Sections of software installation legal requirements

There are four main sections of software license agreements and each one covers different information that is key to the execution of the agreement, as follows:

- i. **General information** – In this section, you will find information about when the agreement will go into effect, how long the terms of the agreement will be active for, and the type of agreement. While this is very general information, it is important because it sets the tone for the entire agreement.
- ii. **Parties involved** – It defines who the parties are that are entering into the agreement. It will not only detail your company as the one offering the license but will also include the details of the person or company that is purchasing the license. You will need to enter in their full name as well as their address and other contact information. You will also need to define whether it is an individual or a company. This information will need to be provided for both parties.
- iii. **Terms of the agreement** – it contains all the terms of the agreement. This includes the price to be paid for the license, which you can define as a flat fee or a flat fee with yearly maintenance fees. You will also include information about whether you will be including the code along with the license and if it is a site license. A site license is another option that will allow your customer to use the software on more than one computer but only at one location. This section will also include some information about maintenance, support, or refunds that will or will not be offered by your company through the agreement.

- iv. **Fine details** – This section is the one that you will use to detail any specifics that are not covered by the other sections of the software license agreement. These terms tend to be more specific to your situation instead of broadly expected terms. Also, in this section will be the location for signatures and dates.

Existing data protection

Data security core principles

Data security is a set of standards and technologies that protect data from intentional or accidental destruction, modification, or disclosure.

The three principles of data security also referred to as information security are;

- 1) **Confidentiality**
- 2) **Integrity**
- 3) **Availability**

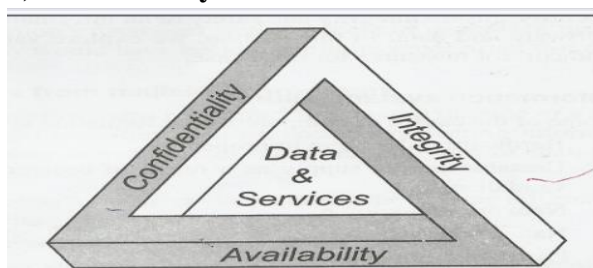


Figure 4:Data Security Core Principles

- 1) **Confidentiality** – means that sensitive data or information belonging to an organization or government should not be accessed by or disclosed to unauthorized people.
- 2) **Integrity** – means that data should not be modified without owner's authority. Data integrity is violated when a person accidentally or with malicious intent, erases or modifies important files
- 3) **Availability** – information must be available on demand. This means that any information system and communication link used to access it, must be efficient and functional.

TYPES OF DATA SECURITY CONTROLS

Types of data security controls include:

Authentication

Authentication technology verifies if a user's credentials match those stored in your database. Today's standard authentication processes include using a combination of ways to identify an authorized user, such as passwords, PINS, security tokens, a swipe card, or biometrics.

Access control

Authentication and authorization happen through the process called access control. Access control systems can include:

Discretionary access control (the least restrictive), which allows access to resources based on the identity of users or groups,

Role-based access control, which assigns access based on organizational role and allows users access only to specific information,

Mandatory access control, which allows a system administrator to strictly control access to all information.

Backups & recovery

Prioritizing data security also requires a plan for how to access your company's and client's data in the event of system failure, disaster, data corruption, or breach. Doing regular data backups is an important activity to help with that access. You can then recover lost data by using your backup.

Encryption

Data encryption software effectively enhances data security by using an algorithm (called a cipher) and an encryption key to turn normal text into encrypted cipher text. To an unauthorized person, the cipher data will be unreadable.

That data can then be decrypted only by a user with an authorized key. Encryption is used to protect the data that you store (called data at rest) and data exchanged between databases, mobile devices, and the cloud (called data in transit). Your encryption keys must be securely managed, including protecting your critical management systems, managing a secure, off-site encryption backup, and restricting access.

Data masking

Data masking software hides data by obscuring letters and numbers with proxy characters. The data is still there, behind the masking. The software changes the data back to its original form only when an authorized user receives that data.

Tokenization

Tokenization substitutes sensitive data with random characters that are not algorithmically reversible. The relationship between the data and its token values is stored in a protected database lookup table, rather than being generated by and decrypted by a mathematical algorithm (as in the case of encryption). The token representing the real data is used across different systems as a replacement, while the actual data is stored on a separate, secure platform.

Deletions & erasure

When electronic data is no longer needed and must be permanently cleared from the system, erasure can overwrite that data so that it is irretrievable. Erasure is different from deletion, which is a process that simply hides data in such a way that makes it easy to retrieve.

Types of software installation

- a. Attended
- b. Unattended
- c. Headless
- d. Schedule/Automated
- e. Clean/Updating.
- f. Network

Attended installation.

An installation process usually needs a user who attends it to make choices, such as accepting or declining an end-user license agreement (EULA), specifying preferences such as the installation location, supplying passwords or assisting in product activation. In graphical environments, installers that offer a wizard-based interface are common. Attended installers may ask users to help mitigate the errors. For instance, if the disk in which the computer program is being installed was full, the installer may ask the user to specify another target path or clear enough space in the disk.

Unattended installation

Installation that is performed without user interaction during its progress or with no user present at all. One of the reasons to use this approach is to automate the installation of a large number of systems. An unattended installation either does not require the user to supply anything or has received all necessary input prior to the start of installation. In unattended installation, it is assumed that there is no user to help mitigate errors. For instance, if the installation medium was faulty, the installer should fail the installation, as there is no user to fix the fault or replace the medium. Unattended installers may record errors in a computer log for later review.

Silent installation

Installation that does not display messages or windows during its progress. "Silent installation" is not the same as "unattended installation" All silent installations are unattended but not all unattended installations are silent. The reason behind a silent installation may be convenience. Malware is almost always installed silently.



**Figure 5: Silent installation
Headless installation**

Installation performed without using a computer monitor connected. In attended forms of headless installation, another machine connects to the target machine (for instance, via a local area network) and takes over the display output. Since a headless installation does not need a user at the location of the target computer, unattended headless installers may be used to install a program on multiple machines at the same time.

Scheduled or automated installation

An installation process that runs on a preset time or when a predefined condition transpires, as opposed to an installation process that starts explicitly on a user's command. For instance, a system administrator willing to install a later version of a computer program that is being used can schedule that installation to occur when that program is not running. An operating system may automatically install a device driver for a device that the user connects. E.g Malware can be installed automatically and automatic software updates

Clean installation

A clean installation is one that is done in the absence of any interfering elements such as old versions of the computer program being installed or leftovers from a previous installation. In particular, the clean installation of an operating system is an installation in which the target disk partition is erased before installation and formatted.

Network installation

Network installation, shortened netinstall, is an installation of a program from a shared network resource that may be done by installing a minimal system before proceeding to download further packages over the network. This may simply be a copy of the original media but software publishers which offer site licenses for institutional customers may provide a version intended for installation over a network.

Software installation and registration

Software installation

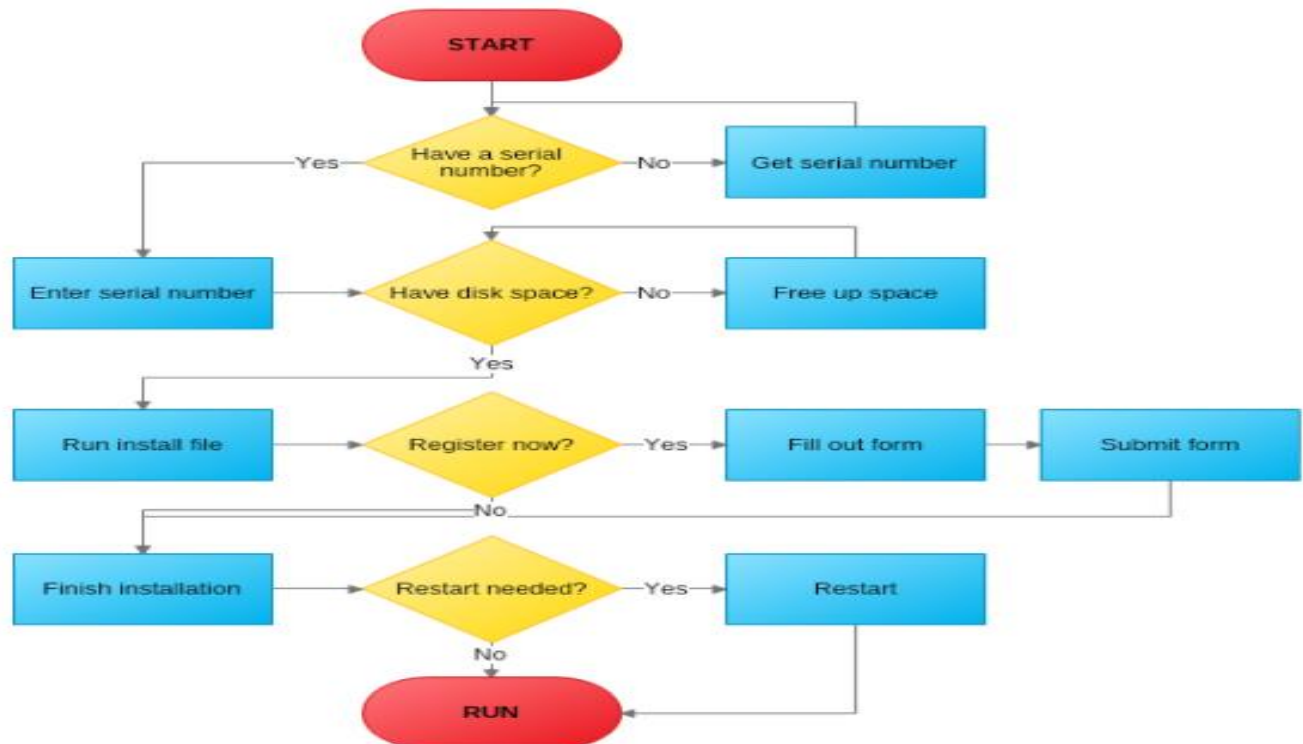


Figure 6:software installation flowchart
Operating System Installation Tasks

The following is an overview of the procedures that are needed to install a new operating system (OS).

1. Set up the display environment.

If you are not using the optional DVD drive for preparing the system for OS installation and installing the OS,

Follow the link for [Accessing the Server Output During Installation](#)

2. Erase the primary boot disk.

If you have an operating system preinstalled on the server, you will need to remove it before installing a new operating system.

To erase the primary boot hard disk:

1. Back up all data that you want to save on the hard drive.
2. Insert the Tools and Drivers CD into the server's optional DVD drive.
3. Boot the server from the CD.
4. When the main menu appears, select the following option from the Tools and Drivers CD main menu:

This option erases all partitions currently located on the primary hard drive, except for the diagnostic partition. If the diagnostic partition is present, it remains untouched.

3. Set up the BIOS.

You need to make sure that the BIOS is set up for the operating system that you plan to install.

Follow the link for [Setting up the BIOS for Operating System Installation](#).

4. Install the operating system.

See the chapter in this document that corresponds to the operating system that you plan to install.

5. Configure your server for RAID.

If you plan to configure your server for RAID operation, you will need to perform some setup tasks before installing the operating system.

Follow the link to [Configure Your System for RAID](#).

7. Install the operating system, update the drivers, and run operating system updates, as necessary.

Click the link below to download the Software Installation Manual

https://www.lanl.gov/orgs/n/n1/FMTTD/symm_analyses/pdfs/sm_installation.pdf

Software registration

Follow the link below for Software Registration Procedure

<https://www.youtube.com/watch?v=RVTBgq73acU>

Importance of registration

When you install software on your computer, you may be asked to register your copy of the software with the manufacturer.

By registering your software, manufacturers can alert you of **security** and **performance updates/patches**.

3.2.2.4 Learning Activities

1. Install Windows 10 Operating System and Microsoft windows 2010.
 - i. Identify Software specifications and computer resource requirements in the computer system provided to establish if it can be installed both Windows 10 and Microsoft 2010
 - ii. Determine the source of software installation files.
 - iii. Check and back up existing data before the installation.
 - iv. Using the tools, equipment, materials, and the manual provided, Install Windows 10 Operating System and Microsoft windows 2010.

2. Creating a Software License Agreement

If you need to create a software license agreement then simply use customizable **software license agreement template** to make your agreement.

3.2.2.5 Self-Assessment

1. Install Windows 10 Operating System. Write down the procedure as you are installing Windows 10 Operating System.
2. Create Windows 10 Installation Media and write down the requirements and its procedure.
3. Describe things to be considered while installing Windows 10 on the PC.
4. Using the tool, create installation media:
5. Erase the primary boot hard disk:

3.2.2.6 Tools, Equipment, Supplies and Materials

- 1 Computer
- 2 External Hard disk
- 3 Flash disk
- 4 CD/DVD
- 5 Digital instructional material including DVDs and CDs.
- 6 Operating system windows 10 Software
- 7 Machines
- 8 Power
- 9 Application software

3.2.2.7 References

ComputerBoards, I. (1999, June 01). SOFTWARE INSTALLATION. Retrieved Nov 13, 2020, from [www.lanl.gov: https://www.lanl.gov/orgs/n/n1/FMTTD/symm_analyses/pdfs/sm_installation.pdf](https://www.lanl.gov/orgs/n/n1/FMTTD/symm_analyses/pdfs/sm_installation.pdf)

Guru99. (2020). Software Configuration Management in Software Engineering. Retrieved Nov 14, 2020 , from guru99.com: <https://www.guru99.com/software-configuration-management-tutorial.html#3>

LegalNature. (2019). The Basics of Software License Agreements. Retrieved Nov 12, 2020, from legalnature.com: <https://www.legalnature.com/guides/the-basics-of-software-license-agreements#what-is-a-software-license-agreement>

Looker Data Sciences, I. (2020). Data security. Retrieved Nov 13, 2020, from <https://looker.com/>: <https://looker.com/definitions/data-security>

Microsoft. (2020). Create installation media for Windows 10 Windows 8.1 Windows 7. Retrieved Nov 12, 2020, from support.microsoft.com: <https://support.microsoft.com/en-us/windows/create-installation-media-for-windows-99a58364-8c02-206f-aa6f-40c3b507420d>

Microsoft. (2020). Create Windows 10 installation media. Retrieved Nov 12, 2020, from microsoft.com: <https://www.microsoft.com/en-us/software-download/windows10>

Microsystems, S. (2006). Operating System Installation Overview. Retrieved Nov 14, 2020, from docs.oracle.com: <https://docs.oracle.com/cd/E19121-01/sf.x2100m2/819-6592-13/Chap1.html#13255>

Wikipedia. (2020, Nov 02). wikipedia. Retrieved Nov 13, 2020, from en.wikipedia.org: [https://en.wikipedia.org/wiki/Wizard_\(software\)](https://en.wikipedia.org/wiki/Wizard_(software))

youtube. (2013, Sep 06). Software Registration Procedure. Retrieved Nov 14, 2020, from youtube.com: <https://www.youtube.com/watch?v=RVTBgq73acU>

youtube. (2020, Mar 22). How to Create Windows 10 Installation Media. Retrieved Nov 12, 2020, from youtube.com: <https://www.youtube.com/watch?v=DGcWddE6c4c>

3.2.2.8 Model answers to self-assessment

1. Create Windows 10 Installation Media and write down the requirements and its procedure.

To create the installation media, the following requirements are required:

- i. A PC with a reliable internet connection.
- ii. A USB flash drive or DVD. A blank USB flash drive with at least 8 GB of space, or a blank DVD (and DVD burner). We recommend using a blank USB or blank DVD because any content on it will be deleted. When burning a DVD from an ISO file, if you're told the disc image file is too large, you'll need to use dual layer (DL) DVD media.
- iii. A product key. Your 25-character product key (not required for digital licenses).

Procedure

To create installation media, go to the software download website, where you'll find step-by-step instructions. On that website, you can select a version of Windows and create your own installation media using either a USB flash drive or a DVD. To go directly to one of the versions, select one of these links:

2. things to be considered while installing Windows 10 on the PC.

- **64-bit or 32-bit processor (CPU).** You will create either a 64-bit or 32-bit version of Windows 10. To check this on your PC, go to **PC info** in PC settings or **System** in Control Panel, and look for **System type**.
- **System requirements.** Make sure the PC meets the system requirements for Windows 10. It is recommended to go to the PC manufacturer's website for additional info about updated drivers and hardware compatibility.
- **Language in Windows.** You will need to choose the same language when you install Windows 10. To see what language you are currently using, go to **Time and language** in PC settings or **Region** in Control Panel.
- **Edition of Windows.** You should also choose the same edition of Windows. To check what edition you are currently running, go to **PC info** in PC settings or **System** in Control Panel, and look for **Windows edition**.
- **Microsoft Office products.** If you just purchased a new device that includes Office 365, it is recommend redeeming (installing) Office before upgrading to Windows 10. To redeem your copy of Office, please see Download and install Office 365 Home, Personal, or University on your PC.

If you have Office 2010 or earlier and choose to perform a clean install of Windows 10, you will need to locate your Office product key. For tips on locating your product key, check Find your Office 2010 product key or Enter the product key for your Office 2007 program

3. Creating installation media Using the tool,

1. Select **Download tool now** and select **Run**. You need to be an administrator to run this tool.
2. If you agree to the license terms, select **Accept**.

3. On the **What do you want to do?** page, select Create **installation media for another PC**, and then select **Next**.

4. Types of software installation

Attended installation - An installation process usually needs a user who attends it to make choices, such as accepting or declining an end-user license agreement (EULA), specifying preferences such as the installation location, supplying passwords or assisting in product activation.

Unattended installation - Installation that is performed without user interaction during its progress or with no user present at all. An unattended installation either does not require the user to supply anything or has received all necessary input prior to the start of installation.

Silent installation - Installation that does not display messages or windows during its progress.

Headless installation - Installation performed without using a computer monitor connected.

Scheduled or automated installation -An installation process that runs on a preset time or when a predefined condition transpires, as opposed to an installation process that starts explicitly on a user's command.

Clean installation -A clean installation is one that is done in the absence of any interfering elements such as old versions of the computer program being installed or leftovers from a previous installation.

Network installation -Network installation, shortened netinstall, is an installation of a program from a shared network resource that may be done by installing a minimal system before proceeding to download further packages over the network.

5. How to erase the primary boot hard disk:
 1. Back up all data that you want to save on the hard drive.
 2. Insert the Tools and Drivers CD into the server's optional DVD drive.
 3. Boot the server from the CD.
 4. When the main menu appears, select the following option from the Tools and Drivers CD main menu:

Learning Outcome 3: Software Configuration Management

3.2.3.1 Introduction to the Learning Outcome

The trainee should be able to explain terms used in software configuration management, describes software configuration components, explain reasons for software configuration (Tracking and Controlling), explain the importance of software configuration management, describe auditing and accounting.

3.2.3.2 Performance Standard

3.2.3.2.1 Software configuration management components are identified.

3.2.3.2.2 Importance and reasons for software configuration management are identified.

3.2.3.3 Information Sheet

Definition of Terms

- Configuration - is the way a system is set up, or the assortment of components that make up the system.
- Software configuration – means selecting programmable options that make the program function to the user's liking.
- Hardware configuration - means assembling desired components for a custom system as well as selecting options in the user-programmable parts of the system.
- Software Configuration Management (SCM)- is a process to systematically manage, organize, and control the changes in the documents, codes, and other entities during the Software Development Life Cycle.
- Change control - is a procedural method which ensures quality and consistency when changes are made in the configuration object.

Software configuration components

There are the four basic requirements for a software configuration management system. These include:

- i. Identification
 - ii. Control
 - iii. Audit
 - iv. status accounting
-
- i. **Identification**—each software part is labeled so that it can be identified.
 - ii. **Control**—, "control" means that proposed changes to a CI are reviewed and, if approved, incorporated into the software configuration. The goal is to make informed decisions and to acknowledge the repercussions associated with a change to the system.

- iii. **Auditing**—the audits allow managers to determine whether software evolution is preceding both logically and in conformance with requirements for the software.
- iv. **Status accounting**—the accounting provides the historic information to determine both what happened and when on the software project. This is critical to the software project manager in making estimates for new systems based on historic information.

Importance of Software Configuration Management

- i. Reduced risk of outages and security breaches through visibility and tracking of the changes to your systems.
- ii. Cost reduction by having detailed knowledge of all the elements of your configuration, avoiding wasteful duplication of your technology assets.
- iii. Improved experience for your customers and internal staff by rapidly detecting and correcting improper configurations that could negatively impact performance.
- iv. Strict control of your processes by defining and enforcing formal policies and procedures that govern asset identification, status monitoring, and auditing.
- v. Greater agility and faster problem resolution, enabling you to provide a higher quality of service and reduce software engineering costs.
- vi. Efficient change management by knowing your baseline configuration and having the visibility to design changes that avoid problems.
- vii. Quicker restoration of service. In an outage, you will be able to recover quickly as your configuration is documented and automated.
- viii. Better release management and clear status accounting.

Participants of SCM process:

Following are the key participants in SCM.

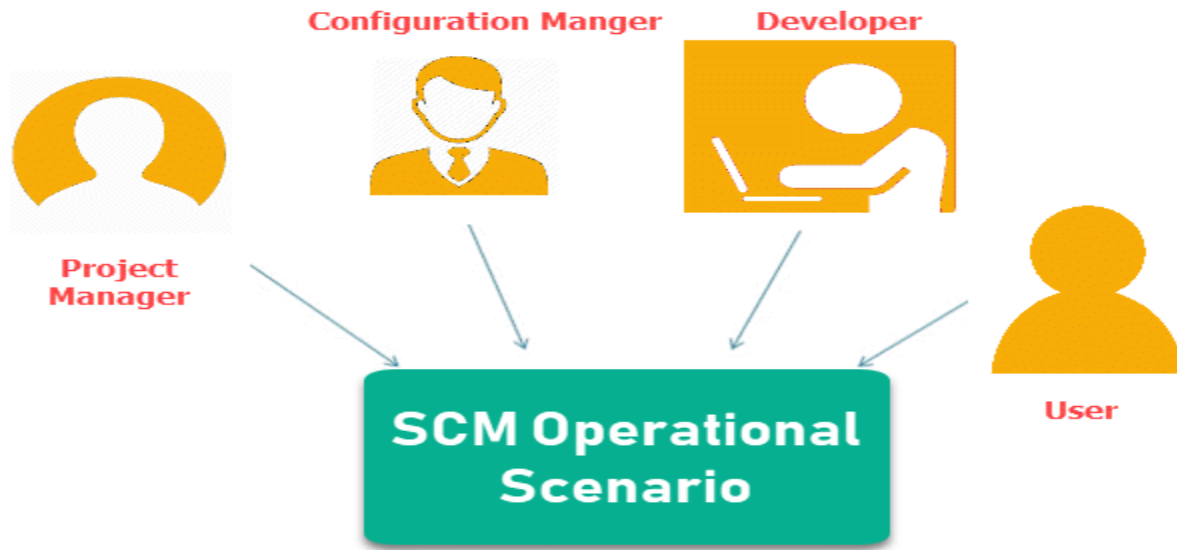


Figure 7:participants in SCM
Configuration Manager

Configuration Manager is the head who is Responsible for identifying configuration items.

CM ensures team follows the SCM process.

He/she needs to approve or reject change requests.

Developer

The developer needs to change the code as per standard development activities or change requests.
He is responsible for maintaining configuration of code.

The developer should check the changes and resolves conflicts.

Auditor

The auditor is responsible for SCM audits and reviews.

Need to ensure the consistency and completeness of release.

Project Manager:

Ensure that the product is developed within a certain time frame.

Monitors the progress of development and recognizes issues in the SCM process.

Generate reports about the status of the software system.

Make sure that processes and policies are followed for creating, changing, and testing.

User

The end user should understand the key SCM terms to ensure he has the latest version of the software.

3.2.3.4 Learning Activities

Visit an ICT department/ industry request for system/ software audit report and identify the key elements captured in the audit report.

3.2.3.5 Self-Assessment

What are the benefits of software configuration management?

Assume you are the Configuration Manager; identify configuration items, describe SCM process.

What are the steps/ processes for a software change control?

3.2.3.6 Tools, Equipment, Supplies and Materials

Computer

Software

External Hard disk

Flash disk

CD/DVD

Digital instructional material including DVDs and CDs;

Operating system

Power

Application software

Puppet

Chef

3.2.3.7 References

TechnologyAdvice. (2020). Configuration Definition & Meaning. Retrieved Nov 16, 2020, from webopedia.com: <https://www.webopedia.com/TERM/C/configuration.html>

ZIFF DAVIS, LLC. PCMAG DIGITAL GROUP. (2020). configuration. Retrieved Nov 16, 2020, from pcmag.com: <https://www.pcmag.com/encyclopedia/term/configuration>

Guru99. (2020). Software Configuration Management in Software Engineering. Retrieved Nov 14, 2020 , from guru99.com: <https://www.guru99.com/software-configuration-management-tutorial.html#3>

3.2.3.8 Model Answers to self-assessment

What are the benefits of software configuration management?

Reduced risk of outages and security breaches through visibility and tracking of the changes to your systems.

Cost reduction by having detailed knowledge of all the elements of your configuration, avoiding wasteful duplication of your technology assets.

Improved experience for your customers and internal staff by rapidly detecting and correcting improper configurations that could negatively impact performance.

Strict control of your processes by defining and enforcing formal policies and procedures that govern asset identification, status monitoring, and auditing.

Greater agility and faster problem resolution, enabling you to provide a higher quality of service and reduce software engineering costs.

Efficient change management by knowing your baseline configuration and having the visibility to design changes that avoid problems.

Quicker restoration of service. In an outage, you will be able to recover quickly as your configuration is documented and automated.

Better release management and clear status accounting

Assume you are the Configuration Manager, you need to configure a server in your organisation, identify configuration items.

Actual Request: Statement of the need.

Reason for the Request: Customer impacts if the request cannot be completed or if considerable time passes before the request can be completed.

Conditions of Success: what they expect from the change.

Expected Completion: The expected due date for the item.

Expected Value: why the request is needed.

What are the steps/ processes for a software change management?



Figure 8:Change management process

3.2.4 Learning outcome 4: Test software functionality and software maintenance

3.2.4.1 Introduction to the learning outcome

The trainee should be able to explain terms used in testing software functionality and software maintenance, describe Techniques of Software Testing (Boundary value analysis, Equivalence class partitioning and Error Guessing), generate Installation checklist, describe Functional Testing (Mainline functions, Basic Usability, Accessibility and Error Conditions) and Generate test report.

3.2.4.2 Performance Standard

- 3.2.1.2.1 Software Techniques are identified.
- 3.2.1.2.2 Software test is performed.
- 3.2.1.2.3 Software functionality is determined according to the test performed.
- 3.2.1.2.4 Test report is generated.

3.2.1.3 Information Sheet

Definition of terms

Testing: is to compare the actual result with the expected result. Testing is done to identify whether all the function is working as expectations

Software testing: is a technique to check whether the actual result matches the expected result and to ensure that the software has not any defect or bug

Installation Testing: It is the process performed to verify if the software has been setup with all the necessary components and it is working as expected.

Uninstallation Testing: Uninstallation testing is performed to verify if all the components of the application is removed during the process or NOT.

Software Testing Techniques - help one to design better test cases aimed at evaluating an attribute or capability of a program or product and determining that it meets its quality.

Software Maintenance - is a part of **Software** Development Life Cycle. Its main purpose is to modify and update **software** application after delivery to correct faults and to improve performance.

Software maintenance - It's the process to modify and update software application after delivery to correct faults and to improve performance.

Software quality factors

- a. **Reliability** - The degree to which the result of a measurement, calculation, or specification can be depended on to be accurate. i.e the software performing consistently well
- b. **Usability** - is the degree to which software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.
- c. **Integrity** - no unauthorized changes have been made to the software.
- d. **Security** - is any type of software that secures and protects a computer, network or any computing-enabled device
- e. **Capability** – software able to perform a specified task

- f. Efficiency - amount of resources required by a program to perform a specific function
- g. **Portability** - the usability of the same software in different environments.
- h. **Maintainability** - The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.
- i. **Compatibility** - is the capacity for two systems to work together without having to be altered to do so.

Techniques of Software Testing

a. Boundary value analysis

Boundary value analysis is based on testing at the boundaries between partitions. It includes maximum, minimum, inside or outside boundaries, typical values and error values.

This software testing technique base on the principle that, if a system works well for these particular values then it will work perfectly well for all values which comes between the two boundary values.

Guidelines for Boundary Value analysis

- If an input condition is restricted between values x and y, then the test cases should be designed with values x and y as well as values which are above and below x and y.
- If an input condition is a large number of values, the test case should be developed which need to exercise the minimum and maximum numbers. Here, values above and below the minimum and maximum values are also tested.
- Apply guidelines 1 and 2 to output conditions. It gives an output which reflects the minimum and the maximum values expected. It also tests the below or above values.

Example:

- Input condition is valid between 1 to 10
- Boundary values 0,1,2 and 9,10,11

b. Equivalence class partitioning

Equivalent Class Partitioning allows you to divide set of test condition into a partition which should be considered the same. This software testing method divides the input domain of a program into classes of data from which test cases should be designed.

The concept behind this technique is that test case of a representative value of each class is equal to a test of any other value of the same class. It allows you to identify valid as well as invalid equivalence classes.

Example:

Input conditions are valid between

- 1 to 10 and 20 to 30

Hence there are five equivalence classes

- --- to 0 (invalid)
- 1 to 10 (valid)
- 11 to 19 (invalid)
- 20 to 30 (valid)
- 31 to --- (invalid)

You select values from each class, i.e.,

-2, 3, 15, 25, 45

c. Error Guessing

Error Guessing is a software testing technique based on guessing the error which can prevail in the code. The technique is heavily based on the experience where the test analysts use their experience to guess the problematic part of the testing application. Hence, the test analysts must be skilled and experienced for better error guessing.

Guidelines for Error Guessing:

- The test should use the previous experience of testing similar applications
- Understanding of the system under test
- Knowledge of typical implementation errors
- Remember previously troubled areas
- Evaluate Historical data & Test results

d. Decision Table Based Testing.

A decision table is also known as to Cause-Effect table. This software testing technique is used for functions which respond to a combination of inputs or events. For example, a submit button should be enabled if the user has entered all required fields.

The first task is to identify functionalities where the output depends on a combination of inputs. If there are large input set of combinations, then divide it into smaller subsets which are helpful for managing a decision table.

For every function, you need to create a table and list down all types of combinations of inputs and its respective outputs. This helps to identify a condition that is overlooked by the tester.

Steps to create a decision table:

- i. Enlist the inputs in rows
- ii. Enter all the rules in the column
- iii. Fill the table with the different combination of inputs
- iv. In the last row, note down the output against the input combination.

Example: A submit button in a contact form is enabled only when all the inputs are entered by the end user.

Table 2:Decision Table Based Testing

	RULE 1	RULE 2	RULE 3	RULE 4	RULE 5	RULE 6	RULE 7	RULE 8
Input								
Name	F	T	F	T	F	T	F	T
Email	F	F	T	T	F	F	T	T
Message	F	F	F	F	T	T	T	T
Output								
Submit	F	F	F	F	F	F	F	T

State Transition

In State Transition technique changes in input conditions change the state of the Application Under Test (AUT). It allows the tester to test the behaviour of an AUT. The tester can perform this action by entering various input conditions in a sequence. The testing team provides positive as well as negative input test values for evaluating the system behaviour.

Guideline for State Transition:

- i. State transition should be used when a testing team is testing the application for a limited set of input values.
- ii. The technique should be used when the testing team wants to test sequence of events which happen in the application under test.

Example:

In the following example, if the user enters a valid password in any of the first three attempts the user will be able to log in successfully. If the user enters the invalid password in the first or second

try, the user will be prompted to re-enter the password. When the user enters password incorrectly 3rd time, the action has taken, and the account will be blocked.

State transition diagram

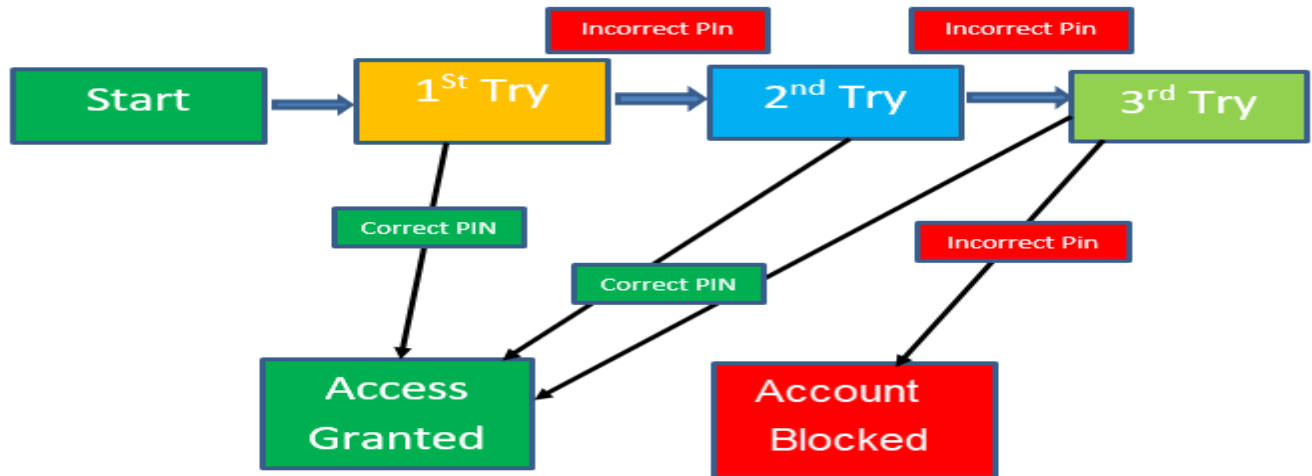


Figure 9:State transition diagram

In this diagram when the user gives the correct PIN number, he or she is moved to Access granted state. The table below is created based on the diagram above-

Table 3 state transition diagram

	Correct PIN	Incorrect PI
S1) Start	S5	S2
S2) 1st attempt	S5	S3
S3) 2nd attempt	S5	S4
S4) 3rd attempt	S5	S6
S5) Access Granted	-	-
S6) Account blocked	-	-

In the above-given table when the user enters the correct PIN, the state is transitioned to Access granted. And if the user enters an incorrect password, he or she is moved to next state. If he does the same 3rd time, he will reach the account blocked state.

Installation checklist

1. To check if, while installing product checks for the dependent software / patches say Service pack3.
2. The product should check for the version of the same product on the target machine, say the previous version should not be over installed on the newer version.
3. Installer should give a default installation path say “C:\programs\.”
4. Installer should allow user to install at location other than the default installation path.
5. Check if the product can be installed “Over the Network”.
6. Installation should start automatically when the CD is inserted.
7. Installer should give the remove / Repair options.
8. When uninstalling, check that all the registry keys, files, Dll, shortcuts, active X components are removed from the system.
9. Try to install the software without administrative privileges (login as guest).
10. Try installing on different operating system.

11. Try installing on system having non-compliant configuration such as less memory / RAM / HDD.
12. Support of the different platforms and configurations needed?
13. Does Installation in "clean state" working?
14. Does the installer is able to calculate needed disk space?
15. Does the installer capturing the baseline free space before launching the installer?
16. Does the amount of space the installer claims it needs for the various type of installation is taken up or is there any discrepancy?
17. How much disk space is used by the installer if the installation is quit midway (this would help gathering information on temp files that is deleted on quitting).
18. Does the installation recover in case an error is met during the installation?
19. Does the installer able to Repair any corrupt installation?
20. If the application installed properly from each type of installation (for typical, custom and complete)?
21. Does Installation over network working?
22. If any file association is made during installation, upon uninstallation, does the association is removed and the base file association is returned to the files?
23. Does running the installer, followed by launching the program to run some tests, and then running the uninstaller, also return your machine to the base state?
24. Does uninstallation leave any registry entry, data files in the system?
25. If there a version of the application to be installed already exists on the machine, does the installer identify that?
26. Does the installer identify if some needed components (such as, MSDE etc.) are already installed on the system?
27. Does running two instances of the installer should prompt a message to the user that an installation setup is already running?
28. If the user logged in, doesn't have write permission for the machine, how installation reacts to this?
29. What happens if the installer tries to install to a directory where there is no write access?
30. Is the installation path configurable/non configurable?
31. Check to ensure that when installing the product, it should provide a browse button which enables the user to install at any folder, and it should provide by default folder (For ex: C:\program files)
32. Is the registering and un-registering the components (dlls) on installation and uninstallation is occurring properly?
33. Are all the files installed in the respective folders and path?
34. Whether all the files/registry values/services are installed properly.
35. Check whether the shortcuts are installed properly and the PATH (any other ENV variables) is updated properly. (It can be CURRENT USERS profile, or All Users profile based on your requirements).

36. Does the installation support, 'UnInstall', 'Modify', 'ReInstall' options?? If yes, does it work?
37. If the installer is supporting upgrade feature, does it preserve all the necessary settings (mostly user preferences)?
38. Check for the user privileges before starting installation. (In most of the cases, installer requires ADMIN privileges).
39. Check the uninstaller entry in add-remove programs. (Check for display string, Display icon and Support information etc.).
40. Does running the installer, and then running the uninstaller, return the machine to the base state?
41. Reinstallation should, apart from identifying previous versions, also should give an option to Remove and Repair.
42. Check to ensure that license key is properly stored in Windows Registry library.
43. Check to ensure that if an evaluation version is installed, then a proper message should be displayed when the date of period is expired for evaluation version with proper error message.
44. Check to ensure that, if Windows Services are installed then it should install in the Services folder of windows directory.
45. Check to ensure that if any product is installed and it is dependent on some other product, then it should give proper message as "The Product is not installed, and it should exit". Check for dependencies.
46. If the product to be installed uses any third party dll and if it is already installed by some other product confirm that the current installation does not un-register/tamper it and uses the existing one.
47. If the dll is already there in the system how does the installation work?
48. And while uninstalling, check should be made as to whether that the shared dll is left without affecting other product.
49. Usability consideration of the installer.

types of testing:

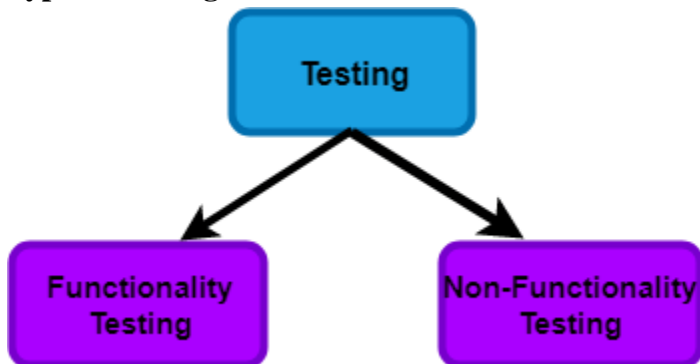


Figure 10:types of testing

Functional Testing:

It is a type of software testing which is used to verify the functionality of the software application, whether the function is working according to the requirement specification. In functional testing, each function tested by giving the value, determining the output, and verifying the actual output with the expected value. Functional testing performed as black box testing which is presented to confirm that the functionality of an application or system behaves as we are expecting. It is done to verify the functionality of the application.

Functional testing also called as black-box testing, because it focuses on application specification rather than actual code. Tester has to test only the program rather than the system.

Functional Testing types

- **Basic Usability:** Functional Testing involves the usability testing of the system. It checks whether a user can navigate freely without any difficulty through screens.
- **Accessibility:** Functional testing test the accessibility of the function.
- **Mainline function:** It focuses on testing the main feature.
- **Error Condition:** Functional testing is used to check the error condition. It checks whether the error message displayed.

Goal of functional testing

The purpose of the functional testing is to check the primary entry function, necessarily usable function, the flow of screen GUI. Functional testing displays the error message so that the user can easily navigate throughout the application.

Process of functional testing

Testers follow the following steps in the functional testing:

- i. Tester does verification of the requirement specification in the software application.
- ii. After analysis, the requirement specification tester will plan.
- iii. After planning the tests, the tester will design the test case.
- iv. After designing the test, case tester will make a document of the traceability matrix.
- v. The tester will execute the test case design.
- vi. Analysis of the coverage to examine the covered testing area of the application.
- vii. Defect management should do to manage defect resolving.

Advantages of functional testing

- i. It produces a defect-free product.
- ii. It ensures that the customer is satisfied.

- iii. It ensures that all requirements met.
- iv. It ensures the proper working of all the functionality of an application/software/product.
- v. It ensures that the software/ product work as expected.
- vi. It ensures security and safety.
- vii. It improves the quality of the product.

Disadvantages of functional testing

- i. Functional testing can miss a critical and logical error in the system.
- ii. This testing is not a guarantee of the software to go live.
- iii. The possibility of conducting redundant testing is high in functional testing.

Steps to perform functional testing:

- There is a need to understand the software requirement.
- Identify test input data.
- Compute the expected outcome with the selected input values.
- Execute test cases.
- Comparison between the actual and the computed result

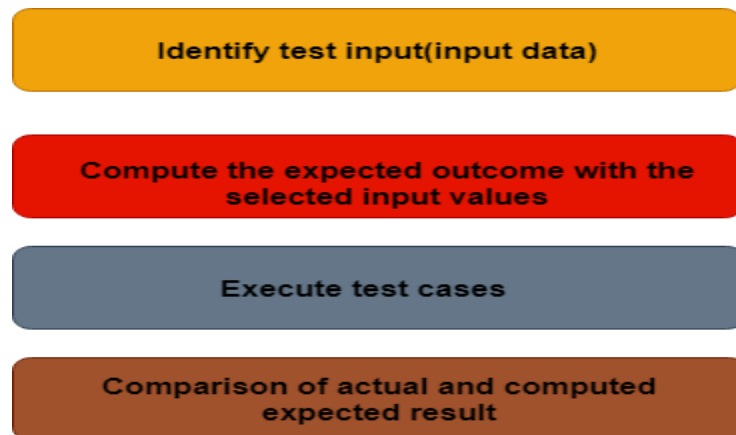


Figure 11:Steps to perform functional testing
Non-Functional Testing

Non-functional testing is a type of software testing to test non-functional parameters such as reliability, load test, performance and accountability of the software. The primary purpose of non-functional testing is to test the reading speed of the software system as per non-functional parameters. The parameters of non-functional testing are never tested before the functional testing.

For example, non-functional testing would be to test how many people can work simultaneously on any software.

Reasons for Non-Functional Testing

Functional and Non-functional testing both is mandatory for newly developed software. Functional testing checks the correctness of internal functions while Non-Functional testing checks the ability to work in an external environment.

It sets the way for software installation, setup, and execution. The measurement and metrics used for internal research and development are collected and produced under non-functional testing.

Non-functional testing gives detailed knowledge of product behaviour and used technologies. It helps in reducing the risk of production and associated costs of the software.

Parameters to be tested under Non-Functional Testing.



Figure 12:Parameters to be tested under Non-Functional Testing.

Performance Testing

Performance Testing eliminates the reason behind the slow and limited performance of the software. Reading speed of the software should be as fast as possible. A well-structured and clear specification about expected speed must be defined.

Load Testing

Load testing involves testing the system's loading capacity. Loading capacity means more and more people can work on the system simultaneously.

Security Testing

Security testing is used to detect the security flaws of the software application. The testing is done via investigating system architecture and the mindset of an attacker. Test cases are conducted by finding areas of code where an attack is most likely to happen.

Portability Testing

The portability testing of the software is used to verify whether the system can run on different operating systems without occurring any bug. This test also tests the working of software when there is a same operating system but different hardware.

Accountability Testing

Accountability test is done to check whether the system is operating correctly or not. A function should give the same result for which it has been created. If the system gives expected output, it gets passed in the test otherwise failed.

Reliability Testing

Reliability test assumes that whether the software system is running without fail under specified conditions or not. The system must be run for a specific time and number of processes. If the system is failed under these specified conditions, reliability test will be failed.

Efficiency Testing

Efficiency test examines the number of resources needed to develop a software system, and how many of these were used. It also includes the test of these three points.

- i. Customer's requirements must be satisfied by the software system.
- ii. A software system should achieve customer specifications.
- iii. Enough efforts should be made to develop a software system.

Advantages of Non-functional testing

- i. It provides a higher level of security. Security is a fundamental feature due to which system is protected from cyber-attacks.
- ii. It ensures the loading capability of the system so that any number of users can use it simultaneously.
- iii. It improves the performance of the system.

- iv. Test cases are never changed so do not need to write them more than once.
- v. Overall time consumption is less as compared to other testing processes.

Disadvantages of Non-Functional Testing

- i. Every time the software is updated, non-functional tests are performed again.
- ii. Due to software updates, people have to pay to re-examine the software; thus, software becomes very expensive.

Generate test report.

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

How to Write a Software Test Report?

Fields to be in the Test Report.

- a. Tested Products and Intermediates
- b. Test Environment
- c. Test Results Overview
- d. Error Definition
- e. Conclusion and Suggestions

Tested Products and Intermediates

The tested product and product components should be specified in this section. In addition, functions that have not been adequately tested should also be explained in this section.

Test Environment

Features related to the test environment are included. The hardware (CPU, ram etc.) features of the test server, the operating system installed on the test server, roles or test automation software installed on the server should be explained in this section.

Test Results Overview

It must be specified separately for each test level (Integration and System tests). The total number of test scenarios, successful and unsuccessful test scenarios written in this area should be clearly stated. Information about whether the criteria of success specified in the Software Test Plan are

met must also be added. Example explanations for the Integration Test Results Overview section can be as follows.

Table 4:Test Results Overview

Modules	Total test	Successful test	Failed test
User module	100	80	20

Error Definition

The error definition, error effects and error correction activities of the failed test scenarios included in the Test Results Overview section are clearly stated in the error definitions section. An example table of error definitions can be as follows.

Table 5 Error definition

Test status no	Error definition	Error effects	Error correction activities
Tc 1	The user whose authorization has expired can log into the system	It may cause security weakness	As a result of the error found in the test, an error record was opened to the relevant developer through the business management tool

Conclusion and Suggestions

Improvement suggestions, if any, can be shared with the project stakeholders by test experts in this section.

3.2.4.4 Learning Activities

1. Fill in the Installation checklist below after successfully installation of Applications and the Operating System,

DESKTOP INSTALLATION CHECKLIST

Name _____ Date _____

Department _____

Standard applications that are on most new installations of desktop computers:

- Microsoft Office (Word, PowerPoint, Excel)
- Antivirus/spyware protection (Endpoint)
- Firefox

- Encryption (Bitlocker)
- Internet Explorer

Action		Action Completed
Backup files	Have you backed up all of your files from the old computer?	
Desktop shortcuts	Do you use shortcuts on your desktop? If so, copy these shortcuts to your backup and just copy those back to your desktop on the new machine.	
Software application list	<p>Make a list of software applications that you need.</p> <p>Name of software application:</p> <ol style="list-style-type: none"> 1. 2. 3. 4. 5. 6. 7. 	

2. Click the link below to Download Sample Test Summary Report:

<https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/Sample-Test-Summary-Report-by-SoftwareTestingHelp.pdf>

3. Generate your own test report guided by the downloaded sample test summary report.

3.2.4.5 Self-Assessment

1. Describe Techniques of Software Testing and highlight their procedures.
2. Explain Installation checklist and describe items contained in it.
3. What is the process of functional testing?
4. Highlight steps to perform functional testing.
5. Describe types of maintenance.

3.2.4.6 Tools, Equipment, Supplies and Materials

1. Diagnostic tools
2. Utility programs
3. Processor and memory optimizers

4. Wise Installer
5. CruiseControl.Net
6. Install Aware
7. Computer
8. Software
9. External Hard disk
10. Flash disk
11. CD/DVD
12. Digital instructional material including DVDs and CDs.
13. Operating system
14. Machines
15. Power
16. Application software

3.2.4.7 References

Abhijit A. Sawant, Pranit H. Bari, P. M. Chawan. (2012). Software Testing Techniques and Strategies. International Journal of Engineering Research and Applications, pp.980-986.

Guru99. (2020). Software Testing Techniques with Test Case Design Examples. Retrieved Nov 14, 2020, from guru99.com: <https://www.guru99.com/software-testing-techniques.html>

JavaTpoint. (2018). functional-testing. Retrieved Nov 14, 2020, from .javatpoint.com: <https://www.javatpoint.com/functional-testing>

Sites, G. (n.d.). Installation Testing Checklist. Retrieved Nov 14, 2020, from sites.google.com: <https://sites.google.com/site/testershelp2/installationtestingchecklist>

tutorialspoint. (2020). Install/Uninstall Testing. Retrieved Nov 14, 2020, from www.tutorialspoint.com: https://www.tutorialspoint.com/software_testing_dictionary/install_uninstall_testing.htm

3.2.4.8 Model answers to self-assessment

1. Techniques Of Software Testing

- i. **Boundary value analysis** - is based on testing at the boundaries between partitions. It includes maximum, minimum, inside or outside boundaries, typical values and error values.

Guidelines for Boundary Value analysis

- a. If an input condition is restricted between values x and y, then the test cases should be designed with values x and y as well as values which are above and below x and y.
- b. If an input condition is a large number of values, the test case should be developed which need to exercise the minimum and maximum numbers. Here, values above and below the minimum and maximum values are also tested.
- c. Apply guidelines 1 and 2 to output conditions. It gives an output which reflects the minimum and the maximum values expected. It also tests the below or above values.

- ii. **Error Guessing** – it is a software testing technique based on guessing the error which can prevail in the code. The technique is heavily based on the experience where the test analysts use their experience to guess the problematic part of the testing application.

Guidelines for Error Guessing:

- a. The test should use the previous experience of testing similar applications
- b. Understanding of the system under test
- c. Knowledge of typical implementation errors
- d. Remember previously troubled areas
- e. Evaluate Historical data & Test results

- iii. **Decision Table Based Testing** - it is also known as to Cause-Effect table. This software testing technique is used for functions which respond to a combination of inputs or events. For example, a submit button should be enabled if the user has entered all required fields.

The first task is to identify functionalities where the output depends on a combination of inputs. If there are large input set of combinations, then divide it into smaller subsets which are helpful for managing a decision table.

For every function, you need to create a table and list down all types of combinations of inputs and its respective outputs. This helps to identify a condition that is overlooked by the tester.

Following are steps to create a decision table:

- Enlist the inputs in rows
- Enter all the rules in the column
- Fill the table with the different combination of inputs
- In the last row, note down the output against the input combination.

- iv. **State Transition** – it is technique changes in input conditions change the state of the Application Under Test (AUT). This testing technique allows the tester to test the behaviour of an AUT. The tester can perform this action by entering various input conditions in a sequence. In State transition technique, the testing team provides positive as well as negative input test values for evaluating the system behaviour.

Guideline for State Transition:

- a. State transition should be used when a testing team is testing the application for a limited set of input values.
- b. The technique should be used when the testing team wants to test sequence of events which happen in the application under test.

2. Installation checklist

1. To check if, while installing product checks for the dependent software / patches say Service pack3.
2. The product should check for the version of the same product on the target machine, say the previous version should not be over installed on the newer version.
3. Installer should give a default installation path say “C:\programs\.”
4. Installer should allow user to install at location other then the default installation path.
5. Check if the product can be installed “Over the Network”
6. Installation should start automatically when the CD is inserted.
7. Installer should give the remove / Repair options.
8. When uninstalling, check that all the registry keys, files, Dll, shortcuts, active X components are removed from the system.
9. Try to install the software without administrative privileges (login as guest).
10. Try installing on different operating system.
11. Try installing on system having non-compliant configuration such as less memory / RAM / HDD.
12. Support of the different platforms and configurations needed?
13. Does Installation in "clean state" working?
14. Does the installer is able to calculate needed disk space?
15. Does the installer capturing the baseline free space before launching the installer?
16. Does the amount of space the installer claims it needs for the various type of installation is actually taken up or is there any discrepancy?
17. How much disk space is used by the installer if the installation is quit midway (this would help gathering information on temp files that's deleted on quitting).
18. Does the installation recover in case an error is met during the installation?
19. Does the installer able to Repair any corrupt installation?
20. If the application installed properly fro each type of installation (for typical, custom and complete)?
21. Does Installation over network working?

22. If any file association is made during installation, upon uninstallation, does the association is removed and the base file association is returned to the files?
23. Does running the installer, followed by launching the program to run some tests, and then running the uninstaller, also return your machine to the base state?
24. Does uninstallation leave any registry entry, data files in the system?
25. If there a version of the application to be installed already exists on the machine, does the installer identify that?
26. Does the installer identify if some needed components (such as, MSDE etc.) are already installed on the system?
27. Does running two instances of the installer should prompt a message to the user that an installation setup is already running?
28. If the user logged in, doesn't have write permission for the machine, how installation reacts to this?
29. What happens if the installer tries to install to a directory where there is no write access?
30. Is the installation path configurable/non configurable?
31. Check to ensure that when installing the product, it should provide a browse button which enables the user to install at any folder, and it should provide by default folder (For ex: C:\program files)
32. Is the registering and un-registering the components (dlls) on installation and uninstallation is occurring properly?
33. Are all the files installed in the respective folders and path?
34. Whether all the files/registry values/services are installed properly.
35. Check whether the shortcuts are installed properly and also the PATH (any other ENV variables) is updated properly. (It can be CURRENT USERS profile or All Users profile based on your requirements).
36. Does the installation support, 'UnInstall', 'Modify', 'ReInstall' options?? If yes, does it work?
37. If the installer is supporting upgrade feature, does it preserve all the necessary settings (mostly user preferences)?
38. Check for the user privileges before starting installation. (In most of the cases, installer requires ADMIN privileges).
39. Check the uninstaller entry in add-remove programs. (Check for display string, Display icon and Support information etc.).
40. Does running the installer, and then running the uninstaller, return the machine to the base state?
41. Reinstallation should, apart from identifying previous versions, also should give an option to Remove and Repair.
42. Check to ensure that license key is properly stored in Windows Registry library.
43. Check to ensure that if an evaluation version is installed, then a proper message should be displayed when the date of period is expired for evaluation version with proper error message.

44. Check to ensure that, if Windows Services are installed then it should install in the Services folder of windows directory.
45. Check to ensure that if any product is installed and it is dependent on some other product, then it should give proper message as "The Product is not installed and it should exit". Check for dependencies.
46. If the product to be installed uses any third party dll and if it is already installed by some other product confirm that the current installation doesn't un-register/tamper it and uses the existing one.
47. If the dll is already there in the system how does the installation work?
48. And also while uninstalling, check should be made as to whether that the shared dll is left without affecting other product.
49. Usability consideration of the installer.

3. Process of functional testing?

Testers follow the following steps in the functional testing:

- i. Tester does verification of the requirement specification in the software application.
- ii. After analysis, the requirement specification tester will make a plan.
- iii. After planning the tests, the tester will design the test case.
- iv. After designing the test, case tester will make a document of the traceability matrix.
- v. The tester will execute the test case design.
- vi. Analysis of the coverage to examine the covered testing area of the application.
- vii. Defect management should do to manage defect resolving.

4. steps to perform functional testing:

- viii. There is a need to understand the software requirement.
- ix. Identify test input data
- x. Compute the expected outcome with the selected input values.
- xi. Execute test cases
- xii. Comparison between the actual and the computed result

5. Types of maintenance

- a. **Corrective Maintenance** - This includes modifications and updations done in order to correct or fix problems, which are either discovered by user or concluded by user error reports.
- b. **Adaptive Maintenance** - This includes modifications and updations applied to keep the software product up-to date and tuned to the ever changing world of technology and business environment.
- c. **Perfective Maintenance** - This includes modifications and updates done in order to keep the software usable over long period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance.

- d. **Preventive Maintenance** - This includes modifications and updates to prevent future problems of the software. It aims to attend problems, which are not significant at this moment but may cause serious issues in future.

3.2.5 Learning outcome 5: perform user training.

3.2.5.1 Introduction to the learning outcome

The trainee should be able to explain terms used in software user training, describe keys to develop an End User Training Plan (Determine user skill set, create a training program, set training goals, train delivery methods, and Assess end-user needs) and generate training feedback template.

3.2.5.2 Performance Standard

3.2.5.2.1 Determine user skill set.

3.2.5.2.2 User training is conducted according to system functionality.

3.2.5.3 Information Sheet

Definition of term

User training – it helps the user in operating the system in efficient way by understanding the problem and how to solve it.

Training program - an activity or activities that include undertaking one or a series of courses to boost performance, productivity, skills, and knowledge of software.

End User Training Plan – is one of the most important steps for a successful system implementation where the end users should be utilized during parallel testing, so training will need to be rolled out prior to that hence will make them excited about the system, as many of them may not have been involved with the project prior to training.

Delivery methods - is a standardized procedure for transferring the product or service to the destination of fulfillment chosen by the customer.

Training feedback - It helps learners to maximize their potential at different stages of training, raise their awareness of strengths and areas for improvement, and identify actions to be taken to improve performance.

Keys to successful implementation/ User training steps

1. Setting training goals your first objective in providing software training for end-users is minimizing any productivity losses associated with the software transition. This means you have to, as quickly as possible, get them up to the skill level required to do their jobs at least as quickly and accurately as they were doing with the old software.

2. Assessing end-user needs an important element in creating your training plan is to evaluate the technical skill level(s) of those who will use the software daily.

3. Training delivery methods

- a. Individual hands-on instructor - It provides real-world experience by allowing the trainee to get her hands directly on whatever she is learning, creating a sense of empowerment.

- b. Hands-on classroom style instructor-led training - This form of training can have one or more instructors; and they teach skills or material to another person or group through lectures, presentations, demonstrations, and discussions.
- c. Seminar style group demonstration - involves showing by reason or proof, explaining or making clear by use of examples or experiments.
- d. Computer Based Training (CBT) - involves the use of a personal or networked computer for the delivery and access of training programs.
- e. Book-based self-paced training - is defined as a specific learning method in which the learner can control the amount of material they consume as well as the duration of time they need to learn the new information properly.

4. Creating a training program End-user training is more effective and memorable if you tailor it to your own organization's use of the software, rather than generic lessons.

5. Making your training program scalable A scalable training program is flexible enough to accommodate both small numbers of users (for example, when new employees join the company and need to be trained on the software) and large numbers (as is necessary in an organization-wide rollout of a new product).

Useful tools in the planning stage

- ✓ Creating a training program
- ✓ Setting training goals
- ✓ Training delivery methods
- ✓ Assessing end-user needs

Training feedback

Table 6:A sample of Training Feedback

Training evaluation questionnaire					
(Name of training session)					
(Date)					
Was the training content what you expected?	Y		N		
Was the size of the group appropriate?	Y		N		
Did the methods of instruction keep you interested in the topic?	Y		N		
Could you see and hear the presentations clearly?	Y		N		
Please rate the quality of the instruction.	1	2	3	4	5

(1 = unacceptable; 5 = outstanding)					
How suitable was the mix of presentation versus activities?	1	2	3	4	5
Please rate the following the qualities of the instructor's delivery: (1 = unacceptable; 5 = outstanding)					
Speed of delivery	1	2	3	4	5
Level of content knowledge	1	2	3	4	5
Organization and preparation	1	2	3	4	5
Enthusiasm	1	2	3	4	5
Ability to keep the group focused	1	2	3	4	5
Please provide any additional feedback for the instructor:					

Did you learn anything new?

Y N

If yes, please provide details:

Do you have any suggestions to improve this course?

Name: _____(optional)

Thank you. We appreciate your feedback!

3.2.5.4 Learning Activities

1. Generate Training Feedback template

3.2.5.5 Self-Assessment

1. Highlight Training delivery methods!
2. Highlight the key question in the training feedback template.

3.2.5.6 Tools, Equipment, Supplies and Materials

- Diagnostic tools
- Utility programs
- Processor and memory optimizers
- Wise Installer
- CruiseControl.Net
- Install Aware
- Computer
- Software
- External Hard disk
- Flash disk
- CD/DVD
- Digital instructional material including DVDs and CDs.
- Operating system
- Machines
- Power
- Application software

3.2.5.7 References

Bhasin, H. (2020, June 30). What is Training Program? Definition, Meaning and Types. Retrieved Jan 01, 2021, from marketing91.com: <https://www.marketing91.com/training-program>

Hepp, M. (n.d.). DeliveryMethod. Retrieved Jan 01, 2021, from schema.org: <https://schema.org/DeliveryMethod>

Strickler, L. (2000, February 25). End User Training. Retrieved Jan 01, 2021, from princeton.edu: <https://www.princeton.edu/~pshrsys/Implementation%20Plan/Strategies/EndUserTraining.htm>

MARS. (n.d.). Training evaluation questionnaire.

Team, T. S. (2013, 06 17). 5 Keys to Developing an End User Training Plan. Retrieved Nov 15, 2020, from sunviewsoftware: <https://www.sunviewsoftware.com/blog/learn/blog/5-keys-to-end-user-training>

3.2.5.8 Model answers to self-assessment

1. Training delivery methods

- i. Individual hands-on instructor
- ii. Hands-on classroom style instructor-led training
- iii. Seminar style group demonstration
- iv. Computer Based Training (CBT)
- v. Book-based self-paced training

Table 7: Sample questions in the training Feedback form

Was the training content what you expected?	Y		N		
Was the size of the group appropriate?	Y		N		
Did the methods of instruction keep you interested in the topic?	Y		N		
Could you see and hear the presentations clearly?	Y		N		
Please rate the quality of the instruction. (1 = unacceptable; 5 = outstanding)	1	2	3	4	5
How suitable was the mix of presentation versus activities?	1	2	3	4	5
Please rate the following the qualities of the instructor's delivery: (1 = unacceptable; 5 = outstanding)					
Speed of delivery	1	2	3	4	5
Level of content knowledge	1	2	3	4	5
Organization and preparation	1	2	3	4	5
Enthusiasm	1	2	3	4	5
Ability to keep the group focused	1	2	3	4	5