

## CHAPTER 4: SOFTWARE INSTALLATION

### 4.1 Introduction

This unit describes the competencies required in installing computer software. It involves identification of software to be installed, installation of the software, configuration of the software, software testing, user training and software maintenance.

### 4.2 Performance Standard

- Classified the software according to the functionality, resource requirement and use
- Established software acquisition methods as per the functionality
- Configured software as per the installation manual provided
- Performed software testing
- Prepared user-training manuals according to software functionality

### 4.3 Learning Outcome

#### 4.3.1 List of the Learning Outcomes

These are the key learning outcomes, which make up workplace function:

- Identify software to be installed
- Install the software
- Configure the software
- Test software functionality
- Perform user training
- Perform Software Maintenance

#### 4.3.2 Learning Outcome 1: Identify software to be installed

##### 4.3.2.1 Learning Activities

The following are the performance criteria:

- Software are classified according to the functionality, resource requirement and use.
- Criteria for selection of software is identified based on user requirements and functionality
- Appropriate software acquisition methods are established as per the functionality.

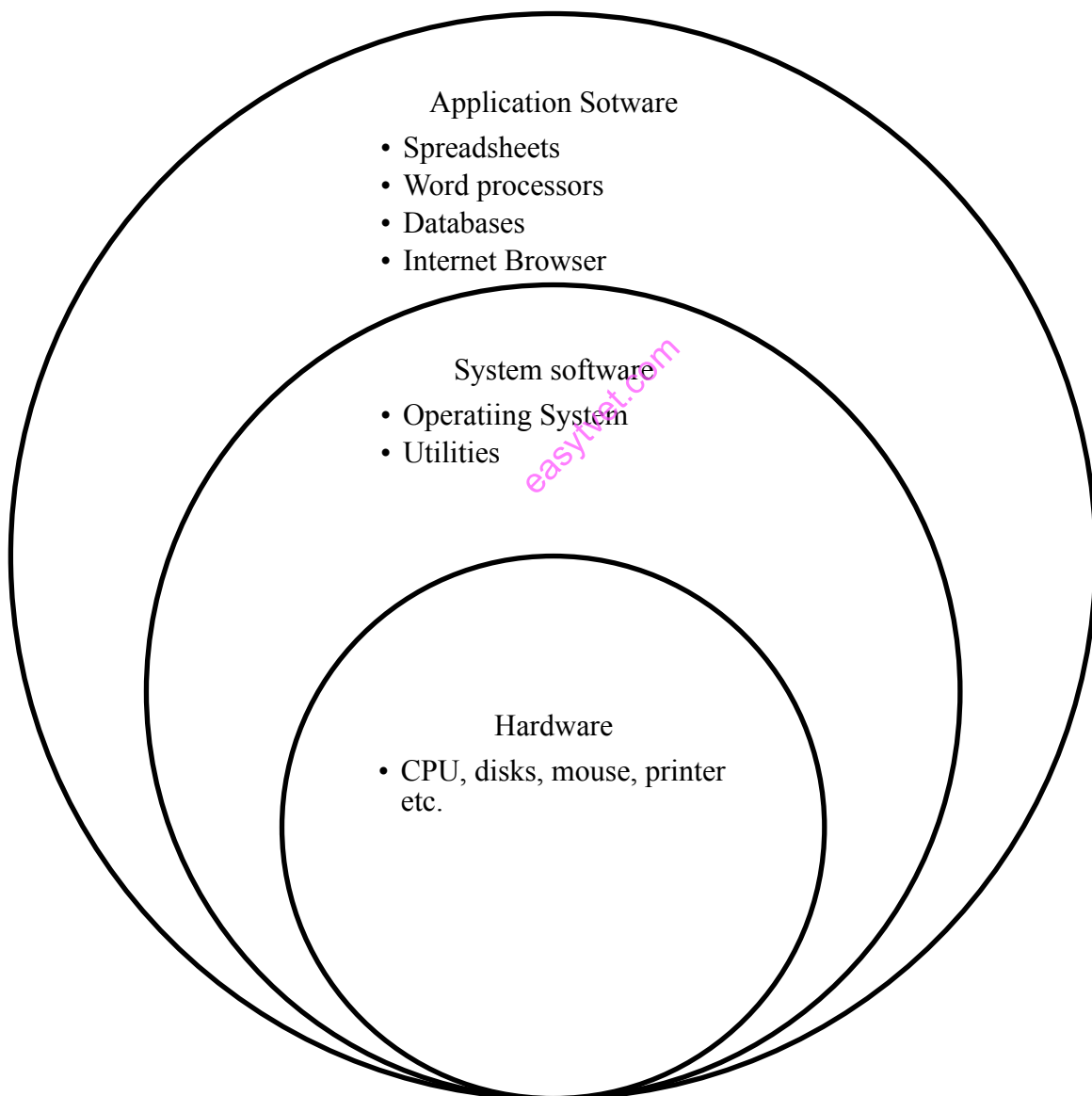
Trainees to demonstrate knowledge in relation to:

- Definition of software
- Classification of software: System and Application
- Criteria for selection
- Operating systems
- Types of operating systems: Single and multi-user, Single and multitasking, Real time, Distributed, Batch

- Functions of operating systems: Device management, Memory management, Storage management, Process control, Security Management
- Types of operating system interfaces: Menu driven and Graphical user Interface

#### 4.3.2.2 Information Sheet

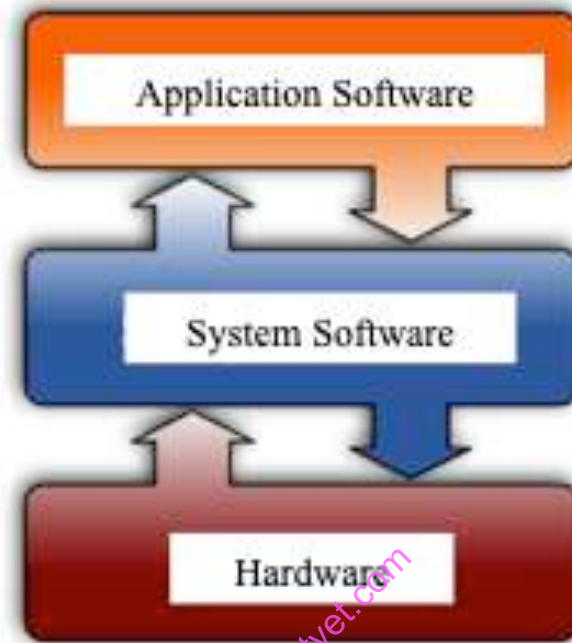
**Software**, in its most general sense, is a set of instructions or programs instructing a computer to do specific tasks. Software is a generic term used to describe computer programs. Scripts, applications, programs and a set of instructions are all terms often used to describe software.



Source: <https://courses.cs.vt.edu>

**Figure 53: Overview of software**

**System software** manages and controls the internal operations of a computer system. It is a group of programs, which is responsible for using computer resources efficiently and effectively. For example, an operating system is **system software**, which controls the hardware, manages memory and multitasking functions, and acts as an interface between application programs and the computer. System software is designed to provide platform for other software.



Source: <https://en.wikibooks.org>

**Figure 54: Interface of application software, system software and hardware**

**Application software** is the general designation of computer programs for performing tasks. Application software may be general purpose (word processing, web browsers, etc.)

**Watch:** Introduction to operating system: <https://youtu.be/0ZjkUMKy6x0>

**Table 10: Difference between System software and Application software**

System Software	Application Software
<ul style="list-style-type: none"> <li>System software is used for operating computer hardware.</li> </ul>	<ul style="list-style-type: none"> <li>Application software is used by user to perform specific task.</li> </ul>
<ul style="list-style-type: none"> <li>System softwares are installed on the computer when operating system is installed.</li> </ul>	<ul style="list-style-type: none"> <li>Application softwares are installed according to user's requirements.</li> </ul>
<ul style="list-style-type: none"> <li>In general, the user does not interact with system software because it works in the background.</li> </ul>	<ul style="list-style-type: none"> <li>In general, the user interacts with application software.</li> </ul>
<ul style="list-style-type: none"> <li>System software can run independently. It provides platform for running</li> </ul>	<ul style="list-style-type: none"> <li>Application software can't run independently. They can't run without</li> </ul>

application softwares.	the presence of system software.
<ul style="list-style-type: none"> <li>Some examples of system softwares are compiler, assembler, debugger, driver, etc.</li> </ul>	<ul style="list-style-type: none"> <li>Some examples of application softwares are word processor, web browser, media player, etc.</li> </ul>

**An Operating System (OS)** is a program that acts as an interface between the software and the computer hardware.

The main difference between **single user and multiuser operating system** is that in a single user operating system, only one user can access the computer system at a time while in a multiuser operating system, multiple users can access the computer system at a time.

An operating system that allows a single user to perform two or more functions at once is a **single-user or multitasking operating system**. Early versions of both Microsoft Windows and the Macintosh operating systems were examples of this category.

**Read:** Categories of operating systems:

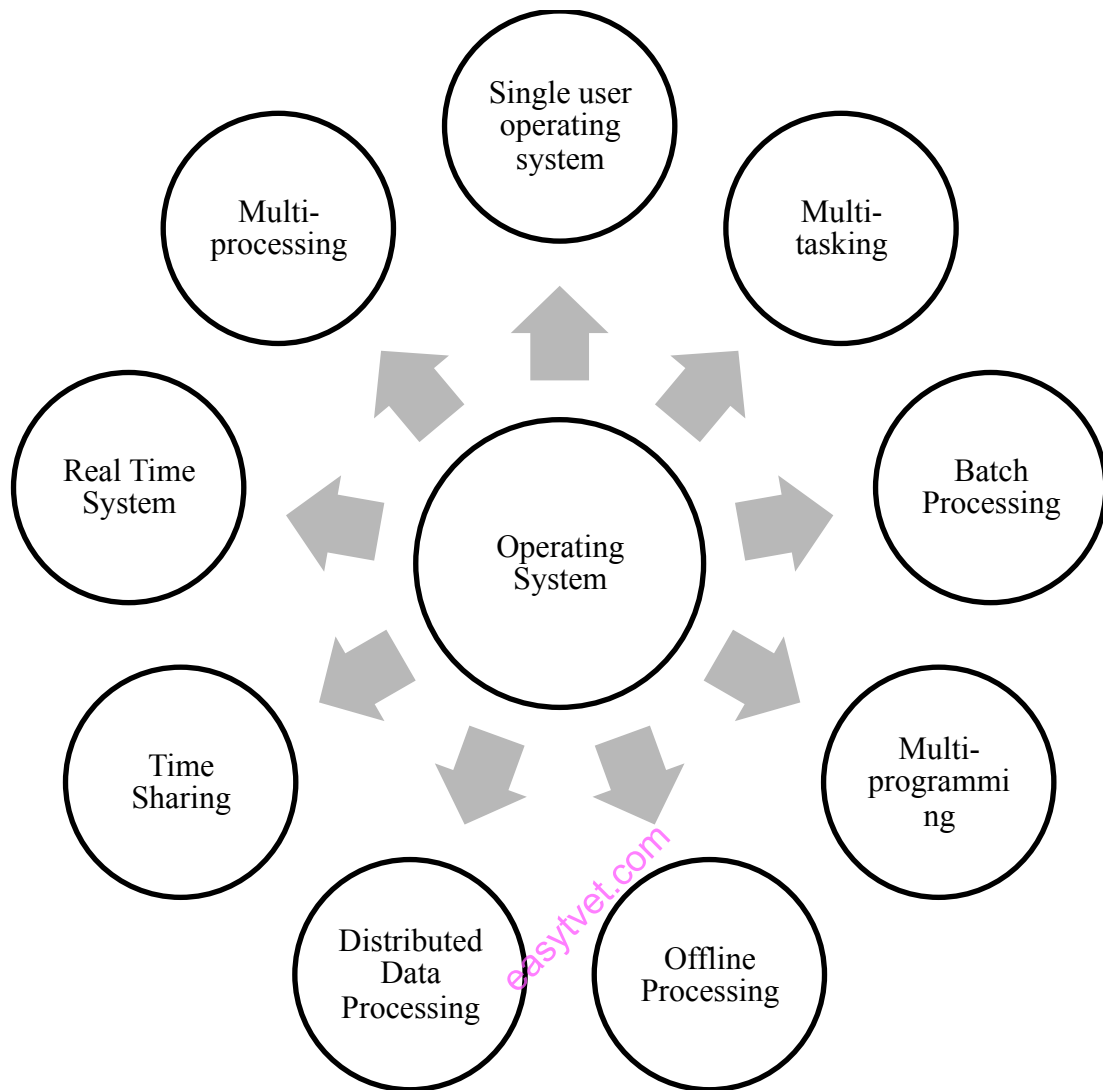
[https://www.tankonyvtar.hu/en/tartalom/tamop412A/2011\\_0009\\_Herdon\\_Miklos-Agroinformatics/ch03s09.html](https://www.tankonyvtar.hu/en/tartalom/tamop412A/2011_0009_Herdon_Miklos-Agroinformatics/ch03s09.html)

**Watch:** Introduction of single or multitasking: <https://youtu.be/glGIGtxwfQ4>

A **real-time operating system (RTOS)** is a very fast and relatively small OS. Often embedded, meaning it is built into the circuitry of a device and not normally loaded from a disk drive, a real-time operating system runs real-time applications. It may support multiple simultaneous tasks or it may only support single tasking.

A **distributed operating system** manages a group of distinct computers and makes them appear to be a single computer. The development of networked computers that could be linked and communicate with each other gave rise to distributed computing. Distributed computations are carried out on more than one machine. When computers in a group work in cooperation, they form a distributed system.

**Batch processing** is a technique in which an Operating System collects the programs and data together in a batch before processing starts. An operating system does the following activities related to batch processing. The OS defines a job, which has predefined sequence of commands, programs and data as a single unit.



**Figure 55: Activities of operating system**

An Operating System manages device communication via their respective drivers. It does the following activities for **device management**:

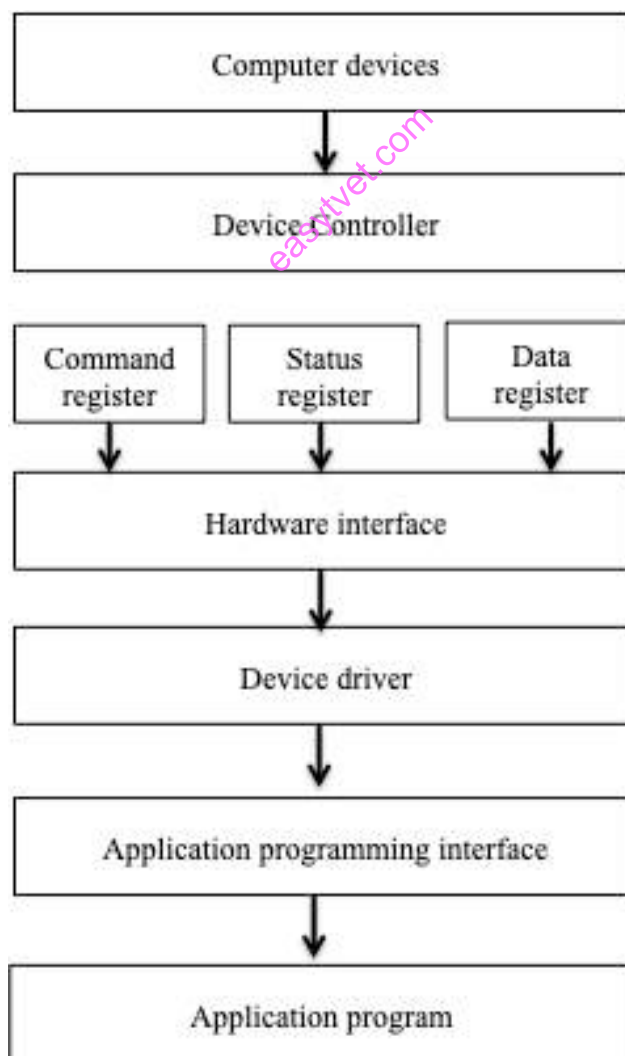
- Keeps tracks of all devices. Program responsible for this task is known as the I/O controller.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

**Memory management** refers to management of **Primary Memory** or **Main Memory**. Main memory is a large array of words or bytes where each word or byte has its own address. Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must be in the main memory. An Operating System does the following activities for memory management:

- Keeps tracks of primary memory, i.e., what part of it are in use by whom, what parts are not in use?
- In multiprogramming, the OS decides which process will get memory when and how much.
- Allocates the memory when a process requests it to do so.
- De-allocates the memory when a process no longer needs it or has been terminated.

Since main memory is usually too small to accommodate all the data and programs permanently, the computer system must provide **secondary storage** to back up main memory. **Process Control** is a data structure in the operating system kernel containing the information needed to manage the scheduling of a particular process.

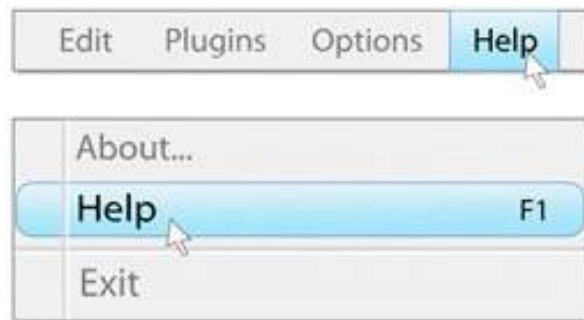
**Operating system security** refers to specified steps or measures used to protect the OS from threats, viruses, worms, malware or remote hacker intrusions. OS security encompasses all preventive-control techniques, which safeguard any computer assets capable of being stolen, edited or deleted if OS security is compromised.



Source: <https://www.includehelp.com>

**Figure 56: Operating System interfaces**

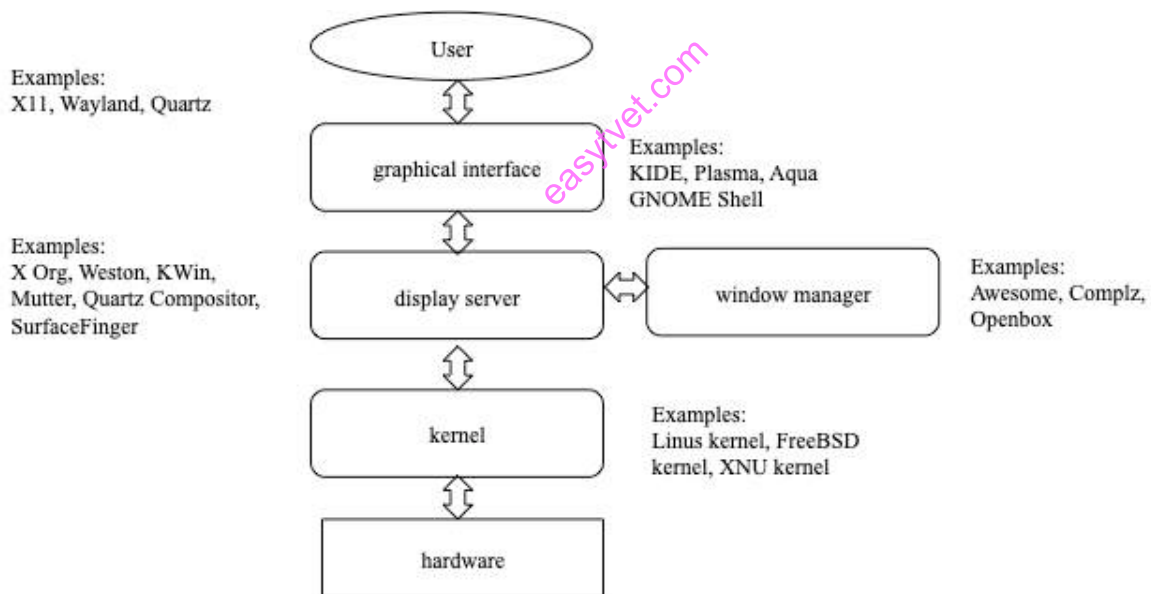
**Menu-driven** is used to describe a software program that is operated using file menus instead of using commands. Below is an example of how a user may quit a menu-driven program, as opposed to a non menu-driven program.



Source: <https://www.computerhope.com>

**Figure 57: Menu**

**Graphical User Interface (GUI)** is an interface that uses icons or other visual indicators to interact with electronic devices, rather than only text via a command line. For example, all versions of Microsoft Windows are a GUI, whereas MS-DOS is a command line.



Source: <https://en.wikipedia.org>

**Figure 58: Approach for Graphical User Interface (GUI)**

#### 4.3.2.3 Self-Assessment

- i. What are real-time systems?
- ii. Describe the objective of multitasking?
- iii. What is menu-driven? Give examples
- iv. What is multitasking?
- v. State differences: real-time operating system vs. distributed operating system
- vi. What is GUI?
  - A. A type of virus

- B. Good universal indicator
  - C. There is no such a thing
  - D. Graphical user interface
- v. **Case situation:** A user is running Windows 8 and wants to upgrade to Windows 10. Is this possible? If yes, how? If no, why?
- vi. Identify the difference between: Ubuntu and VLC player
- vii. \_\_\_\_\_ is an example of menu driven software.
- viii. \_\_\_\_\_ is an example of GUI software program.
- ix. Identify the application software from the image below:



Source: <https://brainly.in>

- x. Types of software programs are
- A. Application programs
  - B. Replicate programs
  - C. Logical programs
  - D. Both A and B
- xi. Examples of system programs includes
- A. Operating system of computer
  - B. Trace program
  - C. Compiler
  - D. All of above

#### 4.3.2.4 Tools, Equipment, Supplies and Materials

Computer, Operating system, Utility program, external hard disk, Deploy master

#### 4.3.2.5 References

- <https://www.techopedia.com/definition/4356/software>
- <http://ecomputernotes.com/software-engineering/characteristics-and-classification-of-software>



- [https://www.tankonyvtar.hu/en/tartalom/tamop412A/2011\\_0009\\_Herdon\\_Miklos-Agroinformatics/ch03s09.html](https://www.tankonyvtar.hu/en/tartalom/tamop412A/2011_0009_Herdon_Miklos-Agroinformatics/ch03s09.html)
- [https://www.tutorialspoint.com/operating\\_system/os\\_overview.htm](https://www.tutorialspoint.com/operating_system/os_overview.htm)
- <https://study.com/academy/lesson/what-is-a-graphical-user-interface-gui-definition-components-examples.html>
- Software Engineering authored by Sumit Prakash Tayal, Bharat Bhushan Agarwal, published by Firewall 2009
- Principles of Operating Systems: Design and Applications authored by Brian Stuart published by Cengage Learning 2008

### 4.3.3 Learning Outcome 2: Install the software

#### 4.3.3.1 Learning Activities

The following are the performance criteria:

- Software specifications and computer resource requirements are identified
- Source of software installation files is determined
- Existing data is backed up
- User vendor agreements are identified according to the installation manual
- Software installation is done as per the installation manual provided

Trainees to demonstrate knowledge in relation to:

- Define software installation
- Acquisition of software
- Installation media
- Software installation legal requirements
- Existing data protection
- Types of software installation: Attended, Unattended, Headless, Schedule/Automated, Clean/Updating, Network
- Software and installation and registration
- Software configuration
- Importance of registration

#### 4.3.3.2 Information Sheet

**Software installation** is the process of making hardware and/or software ready for use. Obviously, different systems require different types of installations. While certain installations are simple and straightforward and can be performed by non-professionals, others are more complex and time-consuming and may require the involvement of specialists.

The **software acquisition** is a computer-aided system that supports the improvement of an organization's **software acquisition** process capability and performance.

**Software installation legal requirements** : Software can be provided as a single user and

multi-user user license:

- **Single user(s):** Software can be activated on single PCs and notebooks.
- **Multi-user network license:** Software can be activated over a network for multiple users.

**Data Protection** controls how your personal information is used by organizations, businesses or the government.

**Read:** Data protection policy: <https://www.gov.uk/data-protection>

**Attended installation** goes through the following setup stages:

- **Setup Program (text mode):** prepares the hard drive for the subsequent stages of installation and copies the files required for running the Setup Wizard. Requires reboot. (Clean installations only.)
- **Setup Wizard (graphical mode):** prompts for additional information such as product key, names, passwords, regional settings, etc.
- **Install Windows Networking:** detects adapter cards, installs networking components (Client for MS Networks, File & Printer Sharing for MS Networks), and installs TCP/IP protocol by default (other protocols can be installed later). You can choose to join a workgroup or domain at this stage. You must be connected to network and provide appropriate credentials to join a domain. After all choices have been made, components are configured, additional files are copied, and the system is rebooted.
- **Post installation:** create user accounts and activate retail versions software. This stage is sometimes referred to as the "Out of Box Experience" (OOBE).

**Read:** Attended installation: [https://www.sqa.org.uk/e-learning/ClientOS01CD/page\\_01.htm](https://www.sqa.org.uk/e-learning/ClientOS01CD/page_01.htm)

**Unattended installation** is a procedure for installing software without user intervention.

**Read:** Unattended installation: <http://www.thenetworkencyclopedia.com/entry/unattended-installation/>

**Headless software** (e.g. "headless java" or "headless Linux") is software capable of working on a device without a graphical user interface. Such software receives inputs and provides output through other interfaces like network or serial port and is common on servers and embedded devices.

A **clean install** is a software installation in which any previous version is eradicated. The alternative to a clean install is an **upgrade**, in which elements of a previous version remain.

**Software license registration** tools and services provide the means for presenting a software

developer's software licenses (also known as an End-User License Agreement, or EULA) to an end user as well as facilitating the transfer of required registration information and consent to the license. Successfully registering a software application often activates the software for legal use by the end user or enterprise.

**Read:** Software licenses: [https://www.webopedia.com/TERM/L/license\\_registration.html](https://www.webopedia.com/TERM/L/license_registration.html)

In software engineering, **software configuration management** (SCM or S/W CM) is the task of tracking and controlling changes in the software, part of the larger cross-disciplinary field of configuration management. SCM practices include revision control and the establishment of baselines.

#### 4.3.3.3 Self-Assessment

- i. What is the importance of registering software?
- ii. Differentiate attended installation and unattended installation.
- iii. What is Software Acquisition?
- iv. Explain headless software.
- v. **Case situation:** Your computer application software prompts you to upgrade it and you do not have registration keys how will fix the issue?
- vi. Identify and install software for office application.
- vii. Identify and install software for doing creative design.
- viii. Hardness software is software capable of working on a device without a graphical user interface
  - A. True
  - B. False
- viii. \_\_\_\_\_ is the process of making hardware and/or software ready for use?
  - A. Hardness software
  - B. Software installation
  - C. Clean install
- ix. \_\_\_\_\_ software can be activated on single PCs and laptops?
  - A. Single user
  - B. Attended
  - C. Multi user
- x. Attended installation goes through the following setup stages:
  - A. Setup Program text mode
  - B. Setup Wizard graphical mode
  - C. Install Windows Networking
  - D. Post installation

#### 4.3.3.4 Tools, Equipment, Supplies and Materials

Computer, external hard disk, deploy master, install ware

#### 4.3.3.5 References

- <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=13031>
- <http://www.thenetworkencyclopedia.com/entry/unattended-installation/>
- [https://www.sqa.org.uk/e-learning/ClientOS01CD/page\\_04.htm](https://www.sqa.org.uk/e-learning/ClientOS01CD/page_04.htm)
- Software Engineering authored by Sumit Prakash Tayal, Bharat Bhushan Agarwal, published by Firewall, 2009
- Principles of Operating Systems: Design and Applications authored by Brian Stuart published by Cengage Learning, 2008

#### 4.3.4 Learning Outcome 3: Configure the software

##### 4.3.4.1 Learning Activities

The following are the performance criteria:

- Software configuration is done as per the installation manual provided
- Required software parameters are set as per the software manual
- Software configuration is done as per the set parameters

Trainees to demonstrate knowledge in relation to:

- Software configuration components: Software Configuration Identification, Software Configuration Control, Software Configuration Status Accounting and Auditing
- Reasons for software configuration: Tracking and Controlling
- Importance of software configuration management: identification and management
- Auditing and accounting

##### 4.3.4.2 Information Sheet

**Identification, control, audit, and status accounting** are the four basic requirements for a **software configuration** management system. These requirements must be satisfied regardless of the amount of automation within the SCM process. All four may be satisfied by an SCM tool, a tool set, or a combination of automated and manual procedures.

**Configuration identification** is the process of identifying the attributes that define every aspect of a configuration item. A configuration item is a product (hardware and/or software) that has an end-user purpose.

The recording and reporting of information needed for configuration management including the status of configuration items proposed changes and the implementation status of approved changes. Status accounting provides the means by which the current state of the development can be judged and the history of the development life cycle can be traced.

A typical Configuration Status Report might include:

- A list of the configuration items that comprise a baseline
- The date when each version of each configuration item was baselined
- A list of the specifications that describe each configuration item

- The history of baseline changes including rationales for change
- A list of open change requests by configuration item
- Deficiencies identified by configuration audits
- The status of works associated with approved change requests by configuration identifier

**Tracking** software observes and tracks the operations and activities of users, applications and network services on a computer or enterprise systems. This type of software provides a way to supervise the overall processes that are performed on a computing system, and provides reporting services to the system or network administrator.

**Software control** is the set of procedures used by organizations to ensure that a software product will meet its quality goals at the best value to the customer, and to continually improve the organization's ability to produce software products in the future.

The purpose of configuration identification is to maintain control of an evolving system by:

- Uniquely identifying the system, revisions of the system and the component parts of each revision
- Understanding the status of configuration items as they progress through the development process.

#### 4.3.4.3 Self-Assessment

- i. What are the requirements of software installation?
- ii. Why identification configuration is important?
- iii. What is software tracking?
- iv. What is software controlling?
- v. Prepare a configuration status report.
- vi. **Case situation:** Download different tracking software and prepare a detailed report on the software used.
- vii. **Case situation:** You are asked to visit a high school hostel to check their ICT lab set up. There key challenge has been high Internet consumption. What will be your approach of addressing their challenges?
- viii. \_\_\_\_\_ is the set of procedures used by organizations to ensure that a software product will meet its quality goals at the best value to the customer?
  - A. Microsoft
  - B. Operating system
  - C. Software control
- ix. \_\_\_\_\_ Software observes and tracks the operations and activities of users, applications and network services on a computer or enterprise systems.
  - A. Sniffing
  - B. Tracking
  - C. Hacking
  - D. Penetration

#### 4.3.4.4 Tools, Equipment, Supplies and Materials

Computer, external hard disk, Deploy master, utility program

#### 4.3.4.5 References

- <https://www.techopedia.com/definition/4313/monitoring-software>
- [http://www.chambers.com.au/glossary/configuration\\_status\\_accounting.php](http://www.chambers.com.au/glossary/configuration_status_accounting.php)
- Software Engineering authored by Sumit Prakash Tayal, Bharat Bhushan Agarwal, published by Firewall 2009

### 4.3.5 Learning Outcome 4: Test software functionality

#### 4.3.5.1 Learning Activities

The following are the performance criteria:

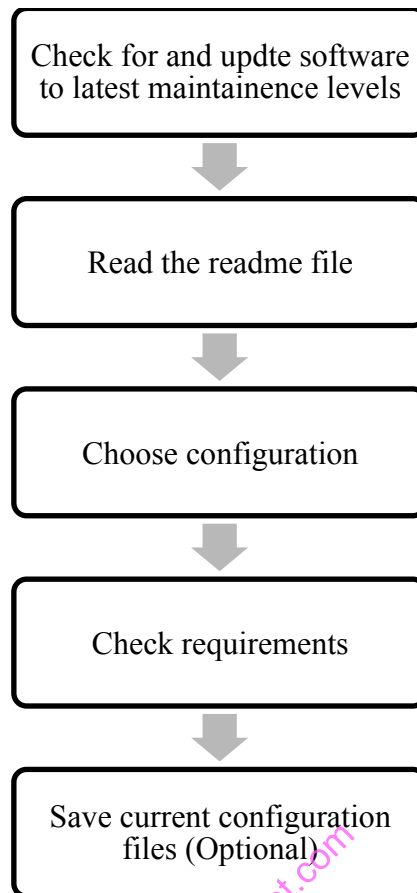
- Software test is performed
- Software functionality is determined according to the test performed
- Test report is generated
- Corrective measures are taken based on the test report

Trainees to demonstrate knowledge in relation to:

- Define software installation testing
- Installation checklist
- Functional Testing: Mainline functions, Basic Usability, Accessibility, Error Conditions

#### 4.3.5.2 Information Sheet

**Installation testing** is performed to check if the software has been correctly installed with all the inherent features and that the product is working as per expectations. Also known as implementation testing, it is done in the last phase of testing before the end user has his/her first interaction with the product.



Source: <https://www.ibm.com>

**Figure 59: Basic program installation steps**

**Installation checklist:** Following is the checklist for installation

- Determine the root path
- Attach the hardware keys
- Install the Software on the server
- Install the Software at every workstation
- Configure the root path at each workstation.
- Test your installation

Functional Testing is defined as a type of testing which verifies that each function of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing and it is not concerned about the source code of the application.

- **Mainline functions:** Testing the main functions of an application
- **Basic Usability:** It involves basic usability testing of the system. It checks whether a user can freely navigate through the screens without any difficulties.
- **Accessibility:** Checks the accessibility of the system for the user
- **Error Conditions:** Usage of testing techniques to check for error conditions. It checks whether suitable error messages are displayed.

**Watch:** How to resolve problems while installing software in Windows 10:

<https://youtu.be/rKKatwf6dSM>

**Watch:** How to install software and resolve issues: <https://youtu.be/3D4CkPNDUf8>

#### 4.3.5.3 Self-Assessment

- i. What are mainline functions?
- ii. Define installation testing.
- iii. Differentiate between accessibility and error conditions
- iv. In which activity of the Fundamental Test Process is the test environment set up?
  - A. Test implementation and execution
  - B. Test planning and control
  - C. Test analysis and design
  - D. Evaluating exist criteria and reporting
- v. Identify a productivity software and install it in a Windows computer.
- vi. Identify an entertainment software and install it in a Windows laptop.
- vii. \_\_\_\_\_ checks the accessibility of the system for the user
  - A. Accessibility
  - B. Usability
  - C. Durability
  - D. None of above
- viii. \_\_\_\_\_ test the main functions of an application
  - A. Basic usability
  - B. Mainline functions
  - C. Accessibility

#### 4.3.5.4 Tools, Equipment, Supplies and Materials

Computer, diagnostic tools

#### 4.3.5.5 References

- <http://www.professionalqa.com/installation-testing>
- <https://www.guru99.com/functional-testing.html>
- Software Engineering authored by Sumit Prakash Tayal, Bharat Bhushan Agarwal, published by Firewall 2009



### 4.3.6 Learning Outcome 5: Perform user training

#### 4.3.6.1 Learning Activities

The following are the performance criteria:

- Determine user skill set as per the instructions manual
- User training manuals are prepared according to software functionality
- User training is conducted according to system functionality

Trainees to demonstrate knowledge in relation to:

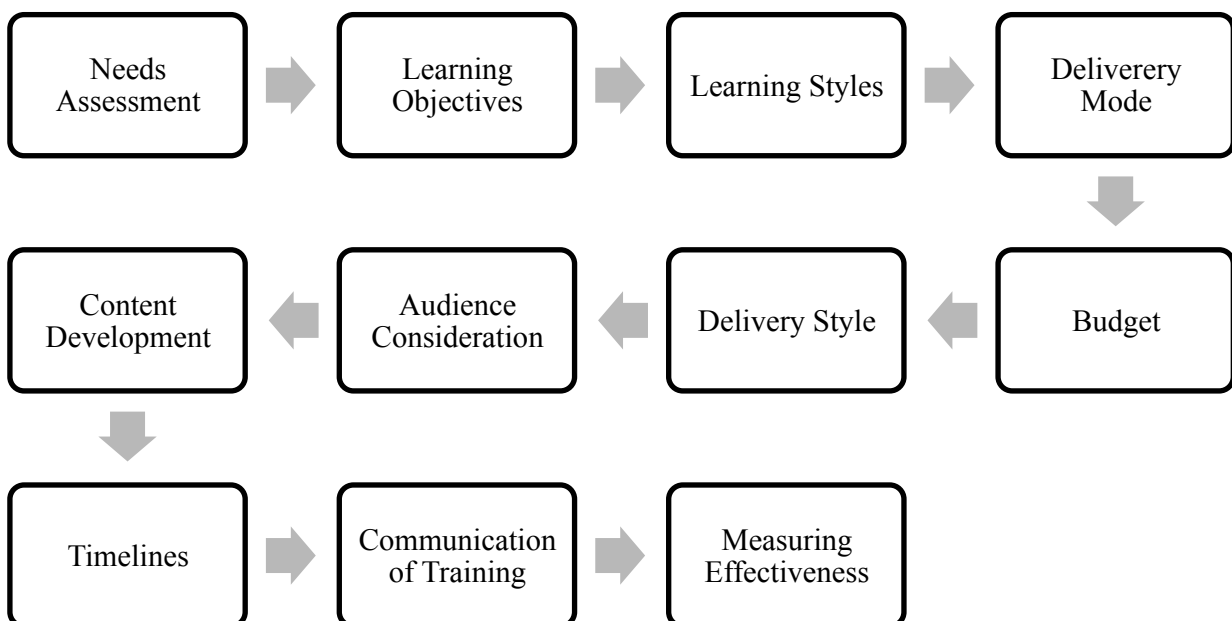
- Keys to Developing an End User Training Plan: Determine user skill set, Creating a training program, Setting training goals, Training delivery methods, Assessing end-user needs
- Training feedback

#### 4.3.6.2 Information Sheet

**Determine set skill** 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).

**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.

**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.



Source: <https://saylordotorg.github.io>

**Figure 60: Training module development and delivery steps**

### Training delivery methods

- Individual hands-on instructor
- Hands-on classroom style instructor-led training
- Seminar style group demonstration
- Computer Based Training (CBT)
- Book-based self-paced training

**Assessing end-user needs** An important element in creating your training plan is to evaluate the technical skill level(s) of those who will actually use the software on a daily basis.

### Overall Considerations

Before considering specific training techniques, ask yourself these questions:

- What are your training goals for this session?
  - New skills
  - New techniques for old skills
  - Better workplace behavior
  - A safer workplace
  - A fair and equal workplace free of discrimination and harassment
- Who is being trained?
  - New employees
  - Seasoned employees
  - Upper management
- What is your training budget?
- How much time has been allocated for training within your organization?
- What training resources and materials do you have at your disposal?

A **training feedback** form is a tool used by training managers and human resource professionals to collect feedback from trainers and trainees. It is used to identify skill gaps and problems to help improve training programs and the overall experience these programs provide.

#### 4.3.6.3 Self-Assessment

- i. Determine user-training needs?
- ii. Explain training delivery methods for software training.
- iii. **Case situation:** Prepare a checklist for assessing training requirements of a women's Sacco group for using social media for enhancing their business opportunities.
- iv. **Case situation:** Prepare a basic introduction to computer course for a community group who has never used computers.
- v. **Case-situation:** You are asked to prepare and submit a proposal for conducting ICT training for using various office packages that promotes productivity in a legal firm. What will be the key training outcomes that you will focus on?

- vi. 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.
  - A. True
  - B. False
- vii. Blended approach is best for non-native computer users.
  - A. True
  - B. False
- viii. Blended training approach means:
  - A. Self-paced training
  - B. Instructor driven training
  - C. Both
  - D. None of the above

#### 4.3.6.4 Tools, Equipment, Supplies and Materials

Computer, Diagnostic tools, Application software

#### 4.3.6.5 References

- <https://www.sunviewsoftware.com/blog/learn/blog/5-keys-to-end-user-training>
- End-User Training (Technological Innovation And Human Resources) authored by Urs E. Gattiker published by D Gattiker

#### 4.3.7 Learning Outcome 6: Perform Software Maintenance

##### 4.3.7.1 Learning Activities

The following are the performance criteria:

- Software maintenance schedule is established
- Software upgrades and modules patches are applied.
- Software revisions are performed to correspond with functionality changes.

Trainees to demonstrate knowledge in relation to:

- Develop software maintenance schedule
- Evaluate the software
- Perform maintenance procedures
- Software maintenance report generated

##### 4.3.7.2 Information Sheet

**Software maintenance** in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes.

The **software evaluation** is done through reviewing non-systematic checklists at regular intervals with different users and evaluation of overall performance efficiency.

**Maintenance procedures** are important. This process requires detailed knowledge of maintenance requirements and the resources that are required in order to perform maintenance. The resources required include labour, parts, materials, and tool costs.

**Maintenance Reporting:** The software developer normally generates a maintenance request form (MRF), sometimes called software problem report that is completed by the user who desires a maintenance activity.

#### 4.3.7.3 Self-Assessment

- i. What is software maintenance?
- ii. What is maintenance evaluation?
- iii. Software Maintenance includes:
  - A. Error corrections
  - B. Enhancements of capabilities
  - C. Deletion of obsolete capabilities
  - D. All of the mentioned
- iv. Maintenance is classified into how many categories?
  - A. Two
  - B. three
  - C. four
  - D. five
- v. **Case situation:** You are software developer and you have developed software for one of your client but you didn't mention to him that the software needs maintenance to work perfectly. What are the steps you will consider taking to ensure that client pays additionally for software maintenance?
- vii. \_\_\_\_\_ in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes.
  - A. Software maintenance
  - B. Maintenance procedure
  - C. Maintenance reporting

#### Tools, Equipment, Supplies and Materials

Diagnostic tool, computer, processor and memory optimizers

#### 4.3.7.5 References

- <https://www.igi-global.com/dictionary/software-evaluation/27677>
- [http://swebokwiki.org/Chapter\\_5:\\_Software\\_Maintenance](http://swebokwiki.org/Chapter_5:_Software_Maintenance)
- [https://www.tutorialspoint.com/software\\_engineering/software\\_maintenance\\_overview.htm](https://www.tutorialspoint.com/software_engineering/software_maintenance_overview.htm)
- Software Engineering authored by Sumit Prakash Tayal, Bharat Bhushan Agarwal, published by Firewall, 2009