**ENTOTO TVET COLLEGE**

under

flagge_12

**Ethiopian TVET-System**

INFORMATION TECHNOLOGY

SUPPORT SERVICE

Level I

**LEARNING GUIDE # 6**

Unit of Competence : Install Software Application

Module Title : Installing Software Application

LG Code : ICT ITS1 L02 06

TTLM Code : ICT ITS1 TTLM 0511

LO 2: Obtain software or software upgrade

|  |  |
| --- | --- |
| **Information Sheet 1** | **Software that conform to the organization needs** |

# Software upgrade plan

The most exciting and at the same time, most frustrating things about information technology is that it is fast changing with which requires new software.

New software products emerge, with fantastic new "must have" features that make it easier for you and your employees to perform your jobs.

**There are two extreme approaches to upgrading:**

1. **The desire to be updated.** Some companies, especially those in the technical industry themselves, desire to be on the cutting edge and always have the latest and greatest. These early adopters toss out their old systems and replace them with the new as soon as it's available.
2. **The conservative outdated way.** Some companies, especially those in more "traditional" or conservative fields, go by the "if it's not broke, don’t fix it" philosophy.

These two strategies have something in common: they aren’t really strategies at all. They’re both reactive - the first driven by reaction to the marketing of new products, and the second driven by reaction to necessity.

The **smarter course is to develop a plan and timeline for upgrading**, taking many factors into consideration including scalability. A long-term plan must take into account future growth and changes anticipated by your company and the way those changes will affect your upgrade needs.

## The five software upgrade plans

## 1. Categorize and prioritize

The first step in putting together your upgrade plan is to categorize and prioritize **potential upgrades**. Some categories of upgrade items include:

* Operating System
* Client software
* Server software
* Word Processing
* Database software
* Graphics software
* Device drivers and etc

Of course, you don’t have to upgrade all of the items in a particular category at the same time.

Software is a very broad category and you may upgrade operating systems and/or specific applications while continuing to use older versions of other applications.

## 2. Set priorities

Once you’ve identified the categories, you can prioritize them based on the cost/benefits ratio of upgrading a particular category.

Of course, in real life your upgrade priorities may not always be based strictly on need. If the big boss wants the top of the line software, there’s a good chance he/she will get it.

## 3. Consider dependencies

Upgrading some categories may be dependent on first upgrading other categories in upgrading or installing new software, you need to also check the hardware requirements.

## 4. Phased rollout

Once you’ve decided to upgrade a particular category, you shouldn’t jump in feet first and roll out the upgrade to every system or device in that category. What if the upgrade causes major problems that make systems or the network unusable? The most prudent strategy is to rollout each upgrade in phases. Test it first in a non-production environment. This gives you a chance to work the bugs out without any impact on employee productivity.

## 5. Keeping it scalable

Your upgrade plan should be set out in writing, and you should get input from different departments and different levels to help you create a plan that will create the least disruption and proceed smoothly.

|  |  |
| --- | --- |
| **Self-Check 2** | **Written Test** |

Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Please ask your trainer for the questionnaire for this self-check.*

|  |  |
| --- | --- |
| **Information Sheet 2** | **Licensing requirements** |

**Software license**

A **software license** (or **software license** in commonwealth usage) is a legal instrument (usually by way of [contract law](http://en.wikipedia.org/wiki/Contract_law)) governing the usage or redistribution of software. All software is [copyright](http://en.wikipedia.org/wiki/Copyright) protected, except material in the public domain.

**Software licenses** can generally be fit into the following **categories**:

* [proprietary](http://en.wikipedia.org/wiki/Proprietary_software) licenses
* [Free and open source](http://en.wikipedia.org/wiki/Free_and_open_source_software) licenses, which include [free software licenses](http://en.wikipedia.org/wiki/Free_software_license) and other [open source licenses](http://en.wikipedia.org/wiki/Open_source_license).

## Proprietary software

The hallmark of proprietary software licenses is that the software publisher grants the use of one or more copies of software under the [***end-user license agreement***](http://en.wikipedia.org/wiki/End-user_license_agreement)***(EULA)***, but ownership of those copies remains with the software publisher (hence use of the term "[proprietary](http://en.wikipedia.org/wiki/Proprietary_software)").

## Free and open source software

A primary consequence of the free software form of licensing is that acceptance of the license is essentially optional — the end-user may use the software without accepting the license. However, if the end-user wishes to exercise any of the additional rights granted by a free software license (such as the right to redistribute the software), then the end-user must accept, and be bound by, the software license.

[Open source licenses](http://en.wikipedia.org/wiki/Open_source_license) generally fall under two categories:

* + - **Copy left licenses.** Those that aim to preserve the openness of the software itself, This license is aimed at giving the end-user permission to redistribute
    - **Permissive licenses.** Those that aim to give freedoms to the users of that software. which essentially grant the end-user permission to do anything they wish with the source code in question, including the right to take the code and use

**Product key**

A **product key**, also known as a **CD key**, is a specific software-based key for a computer program. It certifies that the copy of the program is original. [Activation](http://en.wikipedia.org/wiki/Product_activation) is sometimes done offline by entering the key, and online activation is required to prevent multiple people using the same key.

Product keys consist of a series of numbers and/or letters.

|  |  |
| --- | --- |
| **Self-Check 2** | **Written Test** |

Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Please ask the trainer for the questionnaire for this lap test*

|  |  |
| --- | --- |
| **Lap Test** | **Practical Demonstration** |

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Time started: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time finished: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please ask your trainer for the instruction for this Lap Test