# Plan and Implement an Installation and Upgrade

Unit 29, Assignment 2

**Thomas Robinson**
rob21005043

25th April 2023
Solihull College

# Contents

# 1 Preparation

## 1.1 Plan

1. Back up the existing data that may be on the system.
   This is to avoid any potential data loss during the upgrade and installation of software.
2. Prepare the Operating System upgrade.
   This might need to be written to a bootable USB drive for installation.
3. Download and install the Antivirus software.
   Before installing the other pieces of software, we should first download and install a reputable Antivirus solution
4. Download and install OpenOffice.
5. Download and install Blender.
6. Ensure everything has been updated and is running the latest version
7. Ensure all software functions as-intended. Check for missing drivers, etc.

It is expected that each computer will not take a significant period of time to set up and configure. To finish the installation with in a week, a rate of two new computers a day would need to be sustained.

### Other Considerations

Having communicated with the site team and various tutors, it has been established that the library is primarily used at lunch times. For this reason, it is proposed that the installation take place between 0900–1100 and between 1400–1700 on weekdays to minimize disruption. Consideration should be taken to leaving equipment and components left around during the lunch period and technicians should be mindful of stray cables and similar hazards.

Permission to go-ahead with the work has been obtained on 1/Jan/1970 from the Governors and was signed off by Joe Bloggs (health & safety rep) on 2/Jan/1970. These statements can be found alongside the documentation for this project.

## 1.2 Hardware & Software Required

### 1.2.1 Hardware

**Computers** Ten desktop PCs are required. These should be capable of running the software required. The most intensive program required is Blender, which has minimum requirements of a 64-Bit Quad-Core CPU with SSE2 support, 8 gigabytes of RAM and a GPU with 2 gigabytes of VRAM that supports OpenGL 4.3.

**Monitors** Ten monitors are required. Blender requires a monitor with Full HD (1920x1080) resolution, though the other software will run on lower resolutions.

**Keyboards & Mice** Ten keyboards and mice are required. These should be wired to avoid the need for batteries.

**Network Switch** A network switch is required to connect the computers to the network, and the internet. Appropriate Ethernet cables will also be required. This should have enough ports to connect all ten computers, as well as uplink to the existing network.

**Power Strips** Several power strips will be required. These should be capable of handling the power draw of the computers and monitors, and should have surge protection to protect the equipment from power surges.

### 1.2.2 Software

**Operating System** The computers will run Windows 10, therefore ten Windows 10 licences are required.

**Backup & Disk Cloning Utility** This will be used to create a disk image of the existing data before the upgrade, and to restore the system if anything goes wrong. It can also be used in the future to create backups of the systems.

**Antivirus Software** Will be used to protect the computers from malware and viruses.

**OpenOffice** This is a free, open-source office suite.

**Blender** This is a free, open-source 3D modelling and animation program.

# 2 Handover Process

The handover process is the transfer of responsibilities, information, and control of an IT project or system from one team to another. It will involve sharing documentation and relevant information with the new team, such as plans and credentials, as well as any training that may be required.

Involving a client early in the handover process can ensure that they are familiar with the project or system and allows them to give input into any issues they may experience. This early engagement can build a rapport, and allows clear expectations to be established about the process. The earlier in the process a client is involved, the more time they will be able to spend validating documentation and validate any misunderstandings.

In this context, it is important that any staff that may use the computers with students are well-versed about their operation and things that may go wrong.

# 3 Data Integrity

Data integrity refers to the accuracy, consistency, and reliability of data. In this context, it can mean ensuring that documents stay unchanged and uncorrupted over-time. The integrity of data can be compromised due to human error, software bugs, or hardware failures. Data integrity can be maintained by implementing measures such as validation rules, backups, and conducting regular quality checks.

## 3.1 Implementation

There are many ways to ensure that data integrity is not compromised. The following are some of the most common:

**Hashing** File hashing is the process of taking a file and converting it into a fixed-length string of characters, based on the file contents. This string is called a hash. Hashes are used to verify the integrity of files, and to detect any changes that may have been made to them. Hashes are also used to verify the integrity of data transmission, such as when downloading files from the internet.

**Backups** Backups are copies of data that are stored in a separate location. This allows data to be restored in the event of a failure, corruption or other loss. Backups can be stored on a separate hard drive, or on a cloud storage service. Backups should be made regularly, and should be tested to ensure that they are working correctly.

**Data Validation** Data validation is the process of ensuring that data is correct and consistent. This can be done by implementing validation rules, such as ensuring that a field is not left blank, or that a value is within a certain range. Data validation can also be done by checking the data against a reference dataset, such as a list of valid postcodes. These checks can be run both when data is entered, and when it is requested. They may also be run periodically to ensure that the data is still valid.

## 3.2 Data Integrity Procedure

An example process of ensuring data integrity using the above methods would consist of the following steps:

**Data Entry** Data is entered into the system. This can be done by a user, or by an automated process. This data would be validated before it is stored, to ensure that it is correct and consistent. Validation would be done using validation rules and reference datasets.

**Data Storage** The data is stored in a database. This database is backed up regularly, and the backups are tested to ensure that they are working correctly. The database is also protected from unauthorized access, and is monitored to ensure that it is running correctly. These backups are checked periodically using hashes to ensure that they are not corrupted.

**Data Retrieval** Data is retrieved from the database. This can be done by a user, or by an automated process. This data would be validated before it is returned, to ensure that it is correct and consistent. The validation could be using hashes to ensure that the data has not been corrupted while in-transit or while being stored.

# 4 Restore Process

Before upgrading the system, we will create a disk image (‘clone’) of the contents of its storage medium. This will contain a copy of everything on the drive, from filesystem metadata to user documents. During the clone process, it is possible to use checksums to monitor any data transmission errors that may produce a corrupted image.

Once the image is made, it can be mounted on a different computer and inspected. Choosing a file on both the real system and the cloned drive, their hashes can be compared to ensure that they are identical. This can be repeated several times for more peace-of-mind.

If the upgrade procedure were to go awry, for example if the system is erroneously switched off during the process and the software is unable to recover, the cloned disk image can be used. Writing the disk image back to the drive will restore its original contents, before the upgrade was initiated.

# 5 Upgrading the Operating System

This section will cover the process of upgrading from Windows 7 (Figure 1) to Windows 10. The following chronicles the steps taken to upgrade the system:

1. The first step was to back up the existing data on the system. This was done by creating a disk image of the system's storage medium. This was done using Clonezilla, a free and open-source disk cloning utility. The disk image was saved to a second hard drive. (Not pictured)
2. After the backup was complete, the system was booted into Windows 7 for the last time. The Windows 10 installation media (a DVD, in this case) was inserted into the system.
3. `setup.exe` was run from the installation media. This launched the Windows 10 installer. (Figure 2)
4. The installer checked the compatibility of the system with Windows 10. It was found that the system was compatible, and the installation could proceed. (Figure 3)
5. After confirming, the system began the upgrade process. The system rebooted several times during this process.
6. After the upgrade was complete, the system booted into Windows 10. The first boot installed several updates (Figure 4) and asked for various user privacy settings, such as location and diagnostic data (Figure 5).
7. Now the system was at a Windows 10 desktop, the first task was to make absolutely sure that the system had the latest updates. This was done from the option in the Settings app. (Figure 6)

The system was now running Windows 10 (Figure 7). It was confirmed that basic functionality was working, such as the network connection and the display, alongside the ability to run programs. The system was now ready to be used.


Figure 1: `winver` showing that the system is running Windows 7 before the upgrade

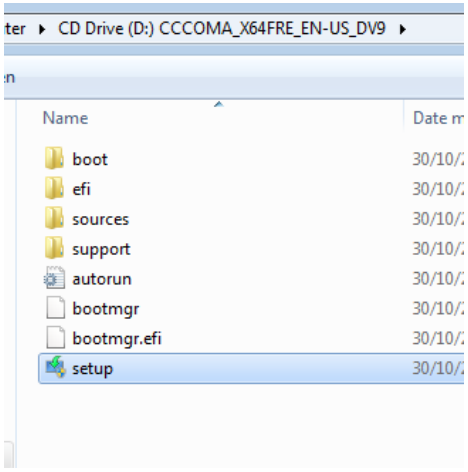


Figure 2: Opening the setup executable on the disc to begin the upgrade

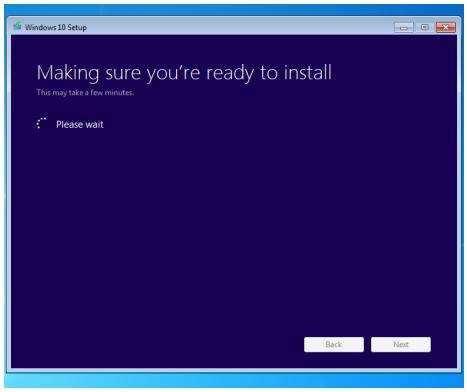


Figure 3: The Windows 10 installer checking the system's compatibility

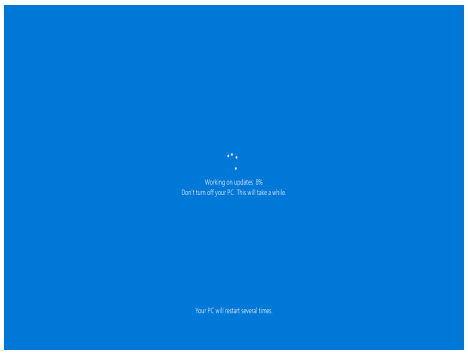


Figure 4: The first boot into Windows 10, installing upgrades

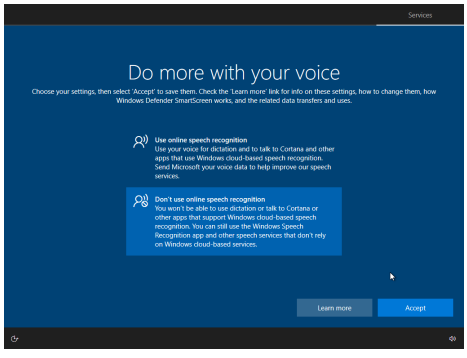


Figure 5: Configuring some settings in the Windows 10 OOBE ('Out of Box Experience')

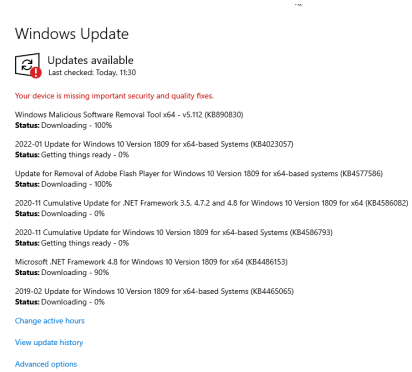


Figure 6: Making sure Windows 10 has all the latest updates


Figure 7: `winver` showing that we are now running Windows 10

# 6 Installing Software

## 6.1 Antivirus (Malwarebytes)

The Antivirus software chosen was Malwarebytes, a popular and well-reviewed piece of software with a capable free version. The following steps were taken to install the software:

1. The installer was downloaded from the Malwarebytes website (Figure 8).
2. The installer was run as an administrator and allowed to install the software (Figure 9).
3. It was confirmed that the software was running the latest version (Not shown).
4. A complete scan was run to verify that Malwarebytes was installed successfully and that the system was free of malware (Figure 10).

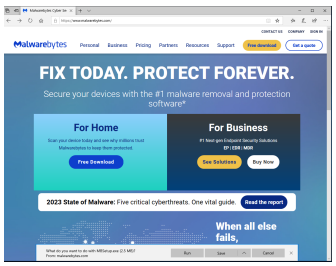


Figure 8: The Malwarebytes Website


Figure 9: Malwarebytes Installer

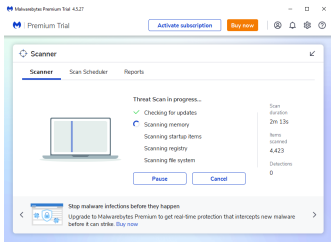


Figure 10: Running a complete scan

## 6.2 OpenOffice

OpenOffice is a free and open-source office suite, with some support for opening and editing Microsoft Office files. Installation of the software was broadly similar to that of Malwarebytes:

1. An appropriate installer was downloaded from the OpenOffice website (Figure 11).
2. The installer was run as an administrator and allowed to install the software. During the installation, the user was prompted to choose which components of the suite to install, alongside various personalization options (Figure 12).
3. Once the installation had finished, OpenOffice Writer was opened to verify that the software had been installed successfully (Figure 13).
4. A Word Document was opened to verify that OpenOffice could open and edit Microsoft Office files (Not shown).


Figure 11: The Apache OpenOffice website, from which an installer was downloaded

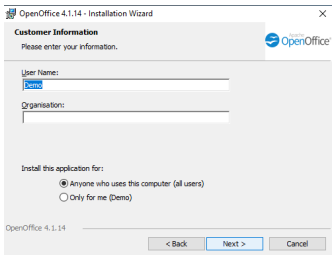


Figure 12: One of the various options presented during the OpenOffice install process

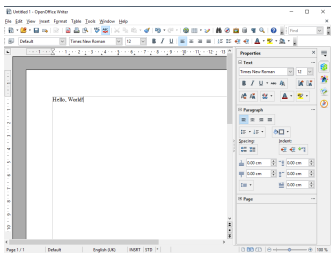


Figure 13: Writing a document in OpenOffice Writer

## 6.3 Blender

Blender is a free and open-source 3D modelling and animation suite. Installation of the software was broadly similar to that of Malwarebytes and OpenOffice, though some additional steps were required to get the software to run on this system.

1. An installer was downloaded from the Blender website (Figure 14).
2. Running as an administrator, the installer was stepped-through, making sure all the required components were selected (Figure 15).
3. Once the installation had finished, there was an attempt to open Blender. This presented an error (Figure 16), which is rectified in the following step.

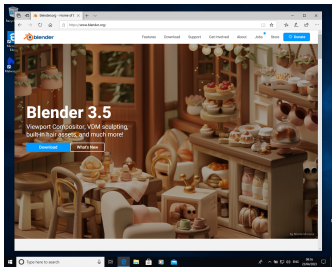


Figure 14: The Blender website

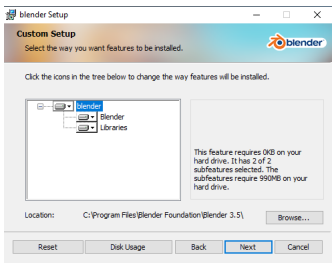


Figure 15: Installing Blender

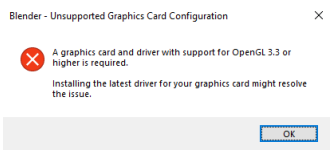


Figure 16: Opening Blender presented an error

### 6.3.1 Problems Encountered Running Blender

When trying to launch Blender for the first time, an error was presented (Figure 16) and the program did not open. This was due to the fact that the graphics card in this example virtual machine does not support OpenGL 3.3. This was rectified by downloading a software renderer, that runs OpenGL code on the CPU rather than the GPU. The installation of this was straightforward; a dynamic link library (DLL) was downloaded from the MESA3D website, and placed in the Blender installation directory (Figure 17). This rectified the issue, and Blender was able to be used as expected—though slower than it would have been if it was able to use the graphics card's hardware acceleration (Figure 18).

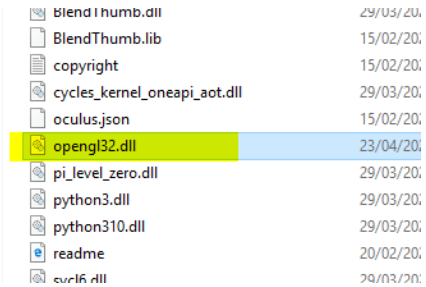


Figure 17: The MESA3D DLL file alongside the Blender program files

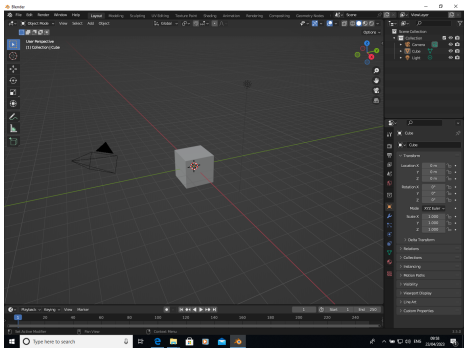

Figure 18: Blender running correctly