Ryan Krage
29.2

Assignment 29.2 – Installing and Upgrading Software

P1 - Describe the potential prompts that initiate the installation of new or upgraded software

Installing and Upgrading software is an important part of computer maintenance, and a good way to expand the functionality of a PC. New software can achieve new tasks, fix problems, or make doing something on the computer easier. Upgrading software can fix problems with the old version, add extra functionality, or patch security vulnerabilities.

Usually, the circumstance that prompts the installation of new software is one of two things – a problem needs solving, or a new PC is being set up.
In the case of a new PC, most users will immediately install software they normally use that does not come with the OS, such as web browsers, office suites, and other tools. This ensures the user is comfortable using and completing tasks on the PC. This is so common that there is a program called Ninite that can automatically download and install a wide range of software, so that the user doesn't have to manually find and install them.

The other scenario that prompts the installation of new software is that the user wants to do something on the computer, but does not have the software required to do so. In this case, the user can either search for software that meets their requirements, or install software they already know about.

Users usually upgrade software for similar reasons. New PC's do not always come with the latest versions of pre-installed software, and often need updating 'out of the box'. This can include driver updates, updates to pre-installed software such as utilities.

The other reason, and this is probably the most common, is that a new version of software is released. Most programs check for updates and prompt the user to install them, or sometimes people check for updates themselves. Updates often bring extra functionality, bug fixes, security patches or performance improvements.

Another reason that people may install or upgrade software is that they have updated another component of the system, such as hardware or other software, and now the software has compatibility issues with the updated component. For example, if a user decides to get a new graphics card, they will need to update their graphics drivers to utilise it.

Another common reason for updates is to fix things that previous updates broke – often a company will release an update that has problems, and then release a 'hotfix' – an update that only fixes what the previous update broke.

P2 - Describe the potential risks of installing or upgrading software

While updates to software usually fix problems and add extra functionality, this is not guaranteed, and sometimes the opposite is true.
Problems can arise from updates when they are not tested properly, or unforeseen circumstances arise.

Essentially, some updates are incompatible with a users system, and this can cause a wide range of problems, ranging from a failed update to loss of data to a 'bricked' (irrevocably broken) system, depending on the severity of the incompatibility.

The most common issue is that a logic error slipped through the testing phase, and when the user encounters it the program crashes. An example of this was when the Google Chrome browser updated the way it processes links, causing some links to lock up the browser.

Other problems could be that the update does not preserve the users settings or data, as the update overwrites the previous version rather than adding to it.

For applications that deal with low-level systems, such a program that interacts with hardware directly rather than going through the OS, an error in the program can physically damage the hardware. An example of this would be the '1970' bug in iOS. When the user set the date to the 1st of January 1970, every process on the device would fail, causing the device to become unusable. This issue could not be fixed by further updates, as the system clock was affected, and could not be changed as any attempt to do so failed, along with every other process.

M1 - Explain the advantages and potential disadvantages of installation or upgrade of software

Advantages of updating software include;

Increased functionality
Updates often add extra features or options to software, enabling it to do more.

Bug fixes
Many updates are just to fix problems with the previous version.

Performance improvements
Sometimes parts or all of a piece of software is re-written to be more efficient and run quicker.

Increased compatibility
Updates often add support for interacting with more software or hardware.

Disadvantages include;

Bugs
Sometimes updates have new errors and problems that can cause problems or even complete inoperability with the software.

UI changes
This is particularly common in newer software. User Interface changes can make software seem unfamiliar and difficult to use. An example of this would be windows 8 – many people struggled to use it as it the UI was different from windows 7.

Reduced support
Sometimes developers drop support for certain systems, either to increased performance on newer systems or to reduce code maintenance. This means users with older systems can no longer use the software. Apple is particularly notorious for this – the majority of apps in the App store require iOS 8 or later, but 4th generation device (iPhone 4, ipod touch 4th gen, etc) do not support anything higher than iOS 6, meaning they cannot get new apps and other updates. They usually drop a generation of devices every years to force customers to buy new products.

M2 - Explain the requirements in preparing for a software installation and upgrade

Users need to do several things in order to ensure they can install new software or upgrade existing software.

Necessary components
This refers to hardware and software that is required for the software to run. For example, a program meant for Windows may require the latest version of the .NET framework, a core part of the Windows OS.

Minimum Requirements
All software requires hardware to run on, and this hardware must be powerful enough to run the software smoothly. While most systems can run the vast majority of software, some software (or an update) requires more powerful hardware to run smoothly and stably. The most common example of this is games – modern games require powerful GPU's CPU's and lots of memory to be playable.

Compatibility
Not all software is compatible with all systems – for example, a .DEB program will not run on Windows, and a .EXE will not run on Linux. Users may need to choose the correct version of the software for their system. However, most download websites can automatically choose the correct version for the user (although they still offer the choice in case they get it wrong).

D2 - Evaluate the risks involved in the installation or upgrade of software and explain how the risks could be minimised

Many problems with upgrades and installations have been mentioned in M1 & M2. There are numerous ways of reducing the risk from an upgrade or installation.
The first, most effective and most common method is to back up any important data – this could be a system-wide update (such as Apple's Time Machine software), or copying data related to the program to another folder or drive.
Prevention is better than cure, and this holds true for updates too. It is important to check that the software is compatible, and pay attention during the installation process to ensure that system settings and data are not unintentionally modified.
Some software comes with (or is disguised) malware. Users should always download installers from reputable sources, decline 'extra' software during installation, and use the file-checking feature of their antivirus to stay safe.