

## 1.1 Prompts for change

So why install or upgrade software? There are a few possible drivers for change.

### Changes to the computer system

Changes to a computer system have a **knock-on effect**.

When hardware is changed, **compatibility issues** between the software and hardware may occur.

In order to resolve this it may be necessary to **upgrade** the existing software to a newer version which has **better compatibility**.

For example, changing a computer system's graphics card may make it difficult to run specialist graphic manipulation software.

Changes to other elements of a computer system can also have an effect, e.g. **upgrading to a new operating system** (such as Windows Vista\*) may cause **incompatibilities** with older software **applications**.

### Fix the bugs

Although the majority of software is subjected to a thorough quality assurance (**QA**) process to test for 'bugs' and unexpected problems before commercial release, many products are bought that contain errors which cause the software to crash (stop working), **lock-up** (stop responding) or **behave unreliably**.

When these mistakes are discovered, it is likely that the software developer (or publisher, if different) will release an **updated version** or a '**patch**' which can **fix the problems** with the software's code.

In the most **severe cases** (e.g. **security flaws**), users are directed to perform these upgrades **as soon as possible**. This is a common feature of -updates for operating systems, web browsers and server software.

### Newer version

Software is sometimes upgraded simply because a new version is released. Newer versions typically:

- work faster and more efficiently;
- offer new features and functionality;
- have better compatibility;
- Resolves issues reported by users of the previous version.

Most modern software is **Internet aware**: this means that it is possible for the software to connect to a remote server to check to see if a newer version is available. If an update is available, the user is typically given the opportunity to download it. Of course, it is up to the user to decide if they want to perform (some or all) of the updates found.

It is always worth remembering that some updates can introduce system problems that didn't exist before

## Needs have changed

Software is purchased for a reason: it **fills a need** identified by a user or an organisation. It is quite common for a user's needs to **change or grow** beyond the capabilities of the software they currently use. For example, a website development team may be happy to use **a basic HTML** editor to create **simple websites**. However, as customer demands become **more complex** they consider purchasing a more sophisticated website design application that offers **a fuller set of features** in order to produce a **better product**.

Wanting **additional functionality** is perhaps the most **common reason** to upgrade or install new software.

Users who have **officially registered** (their software can often upgrade on a regular basis for reduced rates).

## Company policy

Many organisations often have **policies** that control **which** particular piece of software is used for common business applications such as word-processing, spreadsheets, email, web browsing, databases etc. Doing so ensures that any **documents** (or data) created can be **easily shared** between their employees without compatibility issues.

They will also have a company-wide program of **installation and upgrade**, designed to roll out changes to their employees' computer systems in a planned, controlled and systematic manner.

Perhaps two of the most common company-wide changes that are made are updates to office productivity suites (e.g. Microsoft Office) and the operating system itself (e.g. upgrading to a new release of Microsoft Windows or Mac OS).

## 1.2 Justification for change

In order to justify a change it is necessary to present an argument that describes the trade-off between the cost of purchasing new software and the benefits which are likely to be gained.

In most organisations, this is known as a **business case**.

## Key questions to answer in any business case

1. What is the software called?
2. What function does the software serve?
3. Why is the software needed?
4. How much does the software cost?
5. Will extra training be required for employees to use it? If so how much?
6. How will the employees' work improve?
7. Are there any risks?

Typical benefits of using new software include:

- **Increased productivity**;
- **improved efficiency**;
- **extra functionality may automate existing tasks;**
- **improved customer service**;
- **extra functionality may open up new lines of customer business or services;**
- **Improved data sharing and communication within the organisation.**

It is of course quite possible that management may decide that the potential costs (and risks) **far outweigh** the potential benefits; in this case, the software installation (or upgrade) is **not** sanctioned.

### 1.3 Risks

It is almost inevitable that from time-to-time the installation or upgrade of software **will** go wrong.

The results of this can have a **profound effect** on the user of the computer system and a wider impact on their organisation.

**Possible impacts** include:

- potential **loss of service** (for users, i.e. employees and customers)
- **loss of income** while services are incapacitated; '
- introduction of **incompatibility issues**;
- new version of software may be **unstable**;
- users and customers may **not like** it; preferring the previous version.

## Risk reduction measures

A number of different techniques can be employed to **reduce the risk** of software upgrades and updates. These include:

- performing a limited **pilot deployment** of the new version; **creating backups** (of data and software);
- treating a **'bookmark' position** for the operating system so that settings can be rolled back. This is often called a **'system recovery point'**
- installing the upgrade (or update) at a **low-risk time** (e.g. at the weekend so that it can be tested).

It is a sobering thought perhaps that most software developers advise users to **'upgrade' at their own risk'**. This is typically because the computer systems market uses such an **incredibly diverse** mix of hardware and software that it is **almost impossible** to test a new piece of software with **every possible permutation**.

## Braincheck

Answer the following questions

1. What is the difference between software and hardware?
2. What is the difference between software and firmware?
3. What is the difference between an upgrade and an update?
4. Give three reasons for updating/upgrading software.
5. Which process provides justification for an update/upgrade?
6. Give three risks associated with updating/upgrading software..
7. Give three techniques that can be used to reduce upgrade risk;