

## **ICT5 - Future Proofing**

When a system is developed there will be some provision for future expansion and development. The system designers will attempt to allow for increased business volume and they will also try to anticipate future development that may alter the way in which the system will need to operate. Factors such as the need to incorporate new hardware elements - such as portable hand scanners or to provide increased functionality or simply to handle more data at a higher speed will require flexibility of the system.

It would be far too expensive to build a current system in such a way as to cater for anticipated future needs. There are several reasons for this.

- Future needs can only be surmised. Designers could find that capability anticipated is not required or is not sufficient when the time comes.
- It is not sensible to plan for future demands with current hardware and software. At the very least, in a few years time, prices will have fallen. More likely better hardware and software will be available.

An organisation cannot however afford to be constantly replacing and updating in Information System as technology and software develops and as the organisation itself changes. Apart from the high costs of a new system, the organisation will have a heavy investment in staff training and skills. A new system will often require a staff training programme which will be a direct cost, added to this will be the indirect costs of loss of productivity as staff learn to use the new system and the time spent in changing moving data from the old to the new system when changeover occurs.

For these reasons a system will be designed to meet current needs with some provision for anticipated future need so that the lifetime of the new system is made as long as possible.

For only a small overhead some future proofing can be built into a system by building in facilities for expansion or later upgrading. Sockets can be included for additional connections that might be needed later, the system might be designed to handle data from a faster or more expensive peripheral which is not currently needed or supplied. In this way the system can be added to or upgraded as the need arises. A modular design makes this easier since modules can be unplugged and replaced by a later or more powerful version at a later date. The overall cost of upgrading is normally higher than getting it right to start with but the cost is spread over a longer time period and only the upgrades and additions required will be put in place removing the danger of mistakenly planning future needs at the initial design stage.

Once a system is in place and working then there are powerful factors opposing major changes to it. Not least of these is the staff expertise and training invested in the current system. For this reason many new systems will be backward compatible. This means that the new system will be able to process data produced by the old system. An extreme of this is when the new system emulates the old system.

## **Emulation**

Emulation is when hardware or software behaves like some other piece of hardware or software. A printer from manufacturer A might emulate a completely different printer from manufacturer B. This is often done to open up B's market to A.

A second reason for emulation however is to allow a system to be replaced without the immediate need to retrain staff or convert data to a new format. A new system may emulate an old system so that the staff do not see any major difference. This allows the organisation to take advantage of more powerful or more reliable hardware and software while avoiding some of the overheads involved in changeover. However there are problems involved with emulation:

- Emulation will not allow users to make full use of the new system - since its whole purpose is to behave like the old. (It may be possible for new functions to make full use of the new system's power).
- Some of the new systems power or increased capability will be being used to maintain the emulation.
- Emulations are rarely 100%. There are likely to be some problems where the new system does not behave like the old or where there are basic incompatibilities.

One purpose of emulation is to allow software written for one hardware platform to run on another. In this case an emulator program is run which examines each instruction of the program and interprets it for the new hardware platform. This approach means that software running under the control of an emulator will often run more slowly than on its native platform - unless the emulator is running on a much more powerful machine.

## **Credits**

These notes first appeared on <http://www.colchsfc.ac.uk/ict/> and they probably own the copyright or something