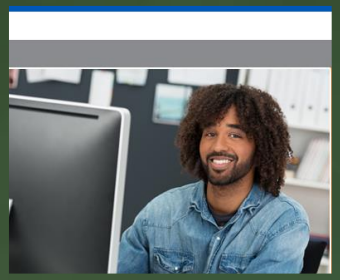
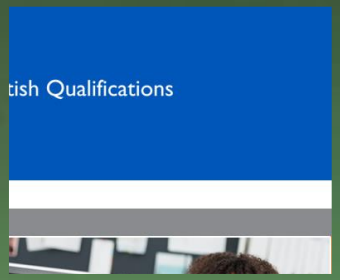
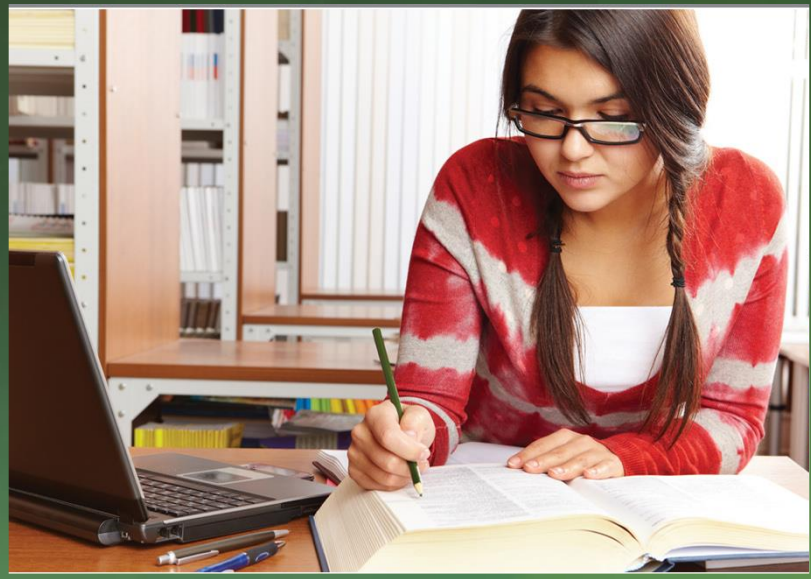
NCC education

Awarding Great British Qualifications







Computer Systems

Topic 8:

Software Maintenance

Scope and Coverage

This topic will cover:

- Software problems
- Automatic & manual updates
- Upgrades
- Utility software
- Security software
- Scheduling maintenance
- ITSM & ITIL framework
- Cloud & Utility Computing

Learning Outcomes

By the end of this topic, students will be able to:

- Perform routine maintenance tasks on a computer system
- Upgrade the hardware and software on a computer system

Software Maintenance?

- There are two broad definitions, depending on the context:
  - In **software development** the phrase *'software maintenance'* means to modify the software code to fix bugs, add features or improve performance. This is all about 'enhancing or improving' the software.
  - In **technical support**, '*software maintenance'* means to install upgrades or patches to current packages and the operating system on a computer. This is all about 'supporting' the software. *This is the definition we shall focus on in this presentation.*

Software Maintenance Process

- The *software maintenance* process can be defined as a series of steps:
  - Preparation of the maintenance tasks according to an overall plan (see the **ITIL** framework later)
  - Allocating these maintenance responsibilities to suitable IT department staff (IT technicians)
  - Implementation and documentation of the changes
  - In large organisations, this is not a trivial workload
  - Known as ‘IT Service Management’ using ITIL

Software Maintenance Process

- Plan a migration process when certain parts of the existing system must be integrated and perhaps moved to a different platform
- Deployment of the new system and the retirement of the old one
- Review of new system and ongoing maintenance

Types of Change

- Software maintenance may include the following types of change:
  - **Adaptive:** Changes to suit different conditions
    - Make the software ‘adapt’ to the new environment/needs
  - **Corrective:** Changes to remove defects
    - Identify and remove flaws/weaknesses in existing version
  - **Perfective:** Changes to improve current version
    - Make the existing software even better!
  - **Preventive:** Changes to reverse deterioration
    - The precursor to ‘corrective’ changes.

Maintenance Model

- A maintenance model can be described as a series of stages that are inter-related and provide several cyclical iterations…
  - Idea/Concept
  - Specification
  - Design
  - Implementation
  - Testing
  - Training
  - Documentation
  - Release
  - User acceptance

Software Problems

- There are only **two** types of software problem:
  - The software *does not work at all…*
    - The software keeps crashing or freezing
    - The software is producing incorrect outputs
    - This means the logic/code is *simply wrong*
  - The software *does not do the required job*…
    - Even though it ‘works’, it does not work *as planned*
    - This means user requirements are not being met
    - This means the logic/code is *solving the wrong tasks*
    - The failure lies in the analysis & design stages
    - There is no failure in the software code itself

Not Working (At All)

- By this we mean that the software does not operate correctly when in use...
  - May be bugs in the specific software
  - May be incompatibility with other software
  - May be incompatibility with the hardware
  - The software is useless and unusable

Not Working (At All)

- Diagnose by reproducing the fault to get clear and accurate details and then searching the Internet for further information
  - Manufacturers often have knowledge bases which document many known issues with their software
  - User groups may also have knowledge bases or forums with helpful information
- The usual fix is to install the latest version, patch or service pack

Not Doing Required Job

- By this we mean that the software does work in a technical sense BUT it lacks some features or facilities that we require…
  - Maybe the original requirements have changed?
  - Maybe the original requirements were never grasped!
  - Maybe the software product being used could never do what we needed?
  - Start by searching the Internet to ensure that the required function can not be done in some other way with the current software.

Not Doing the Required Job

- Check to see whether a new version of the current software does the required job and if so, upgrade
- Investigate replacing the current software with a package that does the required job
- Re-visit the (correct) user requirements and start designing and building again!

Software Changes

- Once you have identified the problem and the solution, a **software change** will be required. This may be:
  - Installation of a minor patch
    - Either manually or automatically
  - Installation of a new version of the current software
    - Either by a major upgrade or a new installation
  - Installation of completely new software
    - i.e. switching to a different product

Updates and Patches

- An update or patch is a minor revision of the software
  - Incorporating bug fixes
  - May add minor features
  - Usually free
  - Usually downloaded either automatically or manually…
    - A Windows service pack is an update
    - An anti-virus update

Automatic Updates

- *Automatic* updates are done by the operating system or package, which checks with a manufacturer's website at regular intervals, downloading and installing updates as they appear
  - The most commonly seen one is Windows Update
  - Automatic - so it saves time and prevents people forgetting to check
  - May introduce new problems
  - Can be annoying, particularly if you are in a hurry!

Manual Updates

- Manual updates are done by the user or administrator.
  - You can control when it is done
  - Often in response to a problem
  - Fixing rather than preventing
  - Needs intervention from user or administrator

Upgrades

- Generally refers to a *significant* change in the software, often adding new features
- Usually charged for?
- Can be anything from a minor change, adding a new feature, to a complete re-write of the software that completely changes how it works.
  - See the various versions of Microsoft Office for examples of both

Utility Software

- There are a number of routine maintenance tasks that can be performed using **utility software**, either from within the operating system or via third-party
  - **Disk defragmentation**
    - Over a period of time, data on the disk gets fragmented (spread-out) and this slows down disk performance
    - Most operating systems come with disk defragmentation software which should be run regularly to avoid this problem

Utility Software

- **Disk clean-up**
  - Over time, disks get filled up with temporary files and deleted files that are still in the Recycle Bin
  - When many updates have been done, there may be many older versions of some software files on disk
  - A regular clean-up of the disk can free-up significant space

Security Software

- Security software, in particular anti-virus software, needs regular updates to remain effective
- New viruses and other threats are coming out all the time – so it is necessary to keep the **virus definition files** up-to-date
- This should be done as an automatic update that cannot be blocked or interrupted

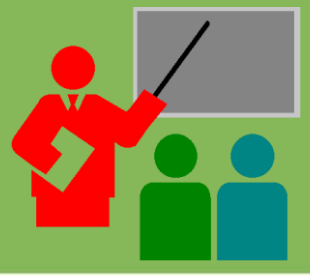
Security Software

- On a large network, this can be centrally co-ordinated, with a server downloading the updates and then distributing them to the clients on the network
- A specific member of IT staff should be tasked with checking every day that this is working properly
  - With other staff in place when the key person is not at work!

Parallel or Replace?

- For any *significant* change to software, you need to decide whether to directly replace the old version with the new, or offer them both in parallel…
  - Replacing may save software licence fees and lead to the new version being adopted by users more quickly
  - Replacing may lead to incompatibility problems with current documents, other systems and users
  - Running parallel systems long-term increases support costs and slows down migration to the new system
  - Running parallel systems allows users to migrate documents if necessary and to learn the new system

Training



- Whichever system you use to migrate to new software, training is vital
  - Productivity is increased as users can use the new system straight away
  - Support costs are reduced as users know how to drive the software
  - Users are more comfortable with the new system, so are more likely to use it

Need to Upgrade?

- Every time an upgrade is suggested or available, this question needs to be asked…

  “Does the upgrade give us something we really need?”

Need to Upgrade?

- Do the advantages of the new system outweigh the upgrade costs?
  - Both in purchasing the new version and rolling it out
  - Remember **Total Cost of Ownership** (TCO)?
- It is usually not necessary to go for every upgrade
- Even if the sales department would like you to!
  - Often every other version, or even one in three, will give a suitable upgrade path

Scheduling Maintenance

- In general, maintenance should be carried out when it will *not disrupt users' normal activities*
- This usually means that significant upgrades must be done *overnight or at weekends*
- See ITIL and cloud computing later

Scheduling Maintenance

- If systems are in use 24/7, then parallel roll-out may be required, where the new system is enabled and users are switched individually as they log out and back in
  - This will require extensive testing of the new system before it is rolled out to users
- If systems are used less heavily at certain times of year, it is better to do major upgrades at quiet times
  - E.g. in a school, doing major upgrades during the longer vacations

Information

- When doing any significant changes, it is vital that users are consulted *before* the changes take place
  - Check when is the best time to do the upgrade
  - Identify the users affected
  - Implement training before the new system rolls out
    - E.g. in a school, check when a particular system is not in use (vacations?) before doing an upgrade
  - This leads to a subject area called ***Change Management***
    - https://www.cipd.co.uk/knowledge/strategy/change/management-factsheet
    - https://www.mindtools.com/pages/article/newPPM_87.htm
    - https://www.apm.org.uk/body-of-knowledge/delivery/scope-management/change-management/

Maintenance Precautions

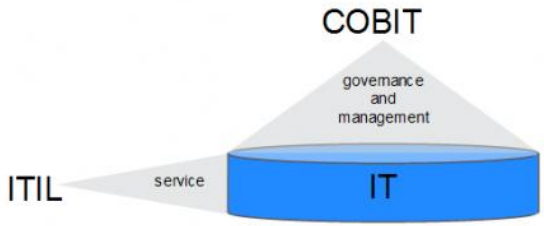
- Back-up all user data
- Back-up or record all system settings
- Establish an application checkpoint
- (System Restore point in Windows)
- Do the upgrade on one system
- Then test before rolling out
- Keep a copy of the update files in case you have to do a full re-installation and then roll forward

ITSM & ITIL

- Software maintenance is a very complex process to manage
- Certainly in large organizations – planning is essential
- It is one aspect of **IT Service Management** (ITSM)
  - https://www.itgovernance.co.uk/itsm
- ITSM stresses the delivery of IT ‘services’ to internal & external clients
- It supersedes the old ‘IT Department’ philosophy
- ITSM is a *discipline* and you can gain ITSM professional certification
- ITSM ‘best practices’ are brought together under the **ITIL** *framework*
  - https://www.axelos.com/best-practice-solutions/itil
- There is even an international *standard* for ITSM:
  - https://apmg-international.com/product/iso-20000

ITSM & ITIL

- There is also the related **COBIT** framework
  - https://www.itgovernance.co.uk/cobit
  - https://www.isaca.org/resources/cobit
- COBIT stands for:
  - **C**ontrol **Ob**jectives for **I**nformation and Related **T**echnologies
- COBIT is all about IT governance, risk-management and compliance
- It is thus pitched at senior managers and corporate decision-makers
- It is more strategic in outlook than ITIL (which is operational in outlook)



Cloud Computing

- This presentation has focused on software maintenance and its effective management, including planning, scheduling and paying for it all
- Wouldn't it be nice if you could off-load all this stress and work onto somebody else?
- Well, now you can!
- The dominant trend in IT now is 'utility' computing
- This means treating IT as a 'utility' like gas, electricity and water
- No need to buy-in expensive hardware, software and IT staff
- Simply use a 'pay-as-you-go' (utility) model and download all your IT services from a remote 'cloud' platform operated by a third-party

Cloud Computing

- You can subscribe to a vast array of IT services from the cloud:
  - IaaS (Infrastructure as a service)
  - PaaS (Platform as a Service)
  - NaaS (Networking as a Service)
  - SaaS (Software as a Service)
  - DaaS (Database as a Service)
  - DRaaS (Disaster Recovery as a Service)
  - GaaS (Gaming as a Service)
- The term '**XaaS**' sums this philosophy up (*Anything as a Service*)
- MS Office 365 is the best known example of cloud-based software
  - https://www.microsoft.com/en-gb/microsoft-365/buy/compare-all-microsoft-365-products

References

- https://www.itgovernance.co.uk/itsm
- https://www.axelos.com/best-practice-solutions/itil/what-is-itil
- http://www.isaca.org/cobit/pages/default.aspx
- https://www.itgovernance.co.uk/cobit
- https://azure.microsoft.com/en-gb/overview/what-is-cloud-computing/
- https://www.ibm.com/cloud-computing/learn-more/what-is-cloud-computing/

Awarding Great British Qualifications

Topic 8 – Software Maintenance

Any Questions?