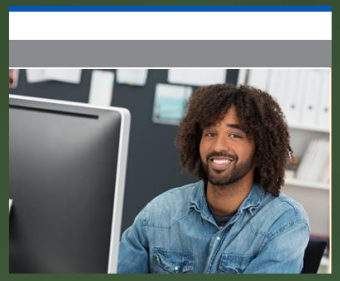
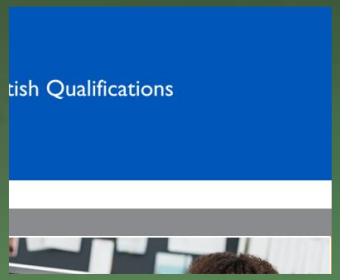
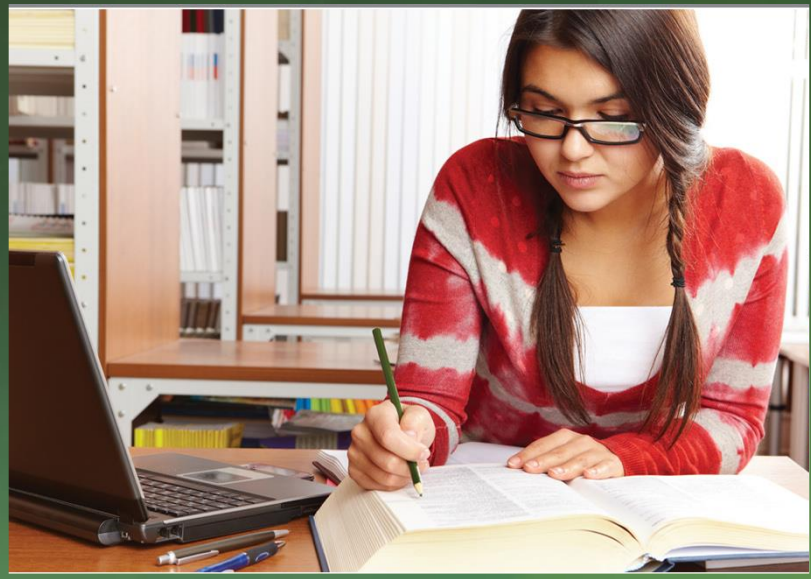
Awarding Great British Qualifications







Computer Systems

Topic 7:

System Testing

Scope and Coverage

This topic will cover:

- Testing Terminology and Concepts
- Types and Categories of Testing
- Test Plans, Standards and Documentation
- Software Testing Issues
- Hardware Testing Issues
- Repair, Replacement and Recycling
- Economics and Eco-Issues

Learning Outcomes

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

- Test and document a computer system

What is Testing?

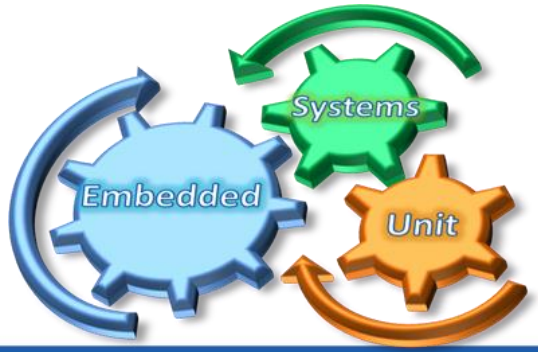
A procedure intended to establish the quality, performance, or reliability of something, especially before it is taken into widespread use.

https://www.lexico.com/definition/test

What is System Testing?

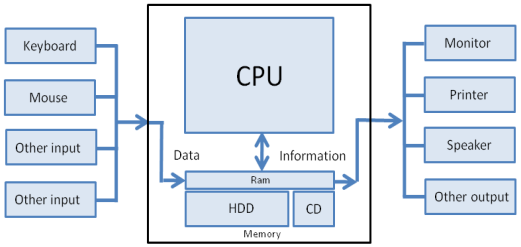
A standard set of procedures done on a complete system to evaluate the compliance with specified requirements

- Within this module, it means testing a **computer system** to ensure that it does what it is meant to do.
- Testing requires a systematic approach to ensure that all aspects are covered.
- Formal test plans and documentation of results are required.

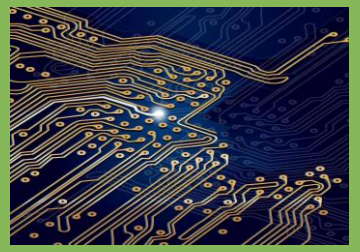


Testing Hardware & Software

- Computer systems comprise hardware and software
- Testing hardware and testing software demand different types of test
- Hardware is made up from *physical* devices
  - It can suffer from wear and tear
  - It can be damaged – by accident or design
  - It could be unfit for purpose
  - It could become obsolete or incompatible
- Software is made up from *logical* instructions
  - It will never ‘wear out’
  - It will never become ‘damaged’ (malware?)
  - It too can be unfit for purpose
  - It too can become obsolete or incompatible



Hardware Testing (A)



- **Design Verification** (Compliance Testing)
  - Performed during *product development*
  - Usually on a small sample of products
  - Involves rapid prototyping and simulations
  - May involve 3-D printing to test different designs
  - Emphasis on speed of evaluation and iteration
  - Does the product do the required job?
  - May involve optimising circuit design too.

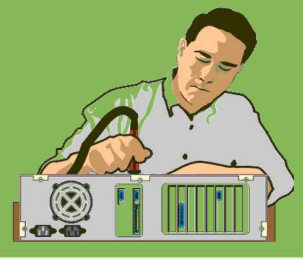
Hardware Testing (B)

- **Manufacturing (Production) Test**
  - We can now assume actual product *design* is valid
  - This test is done during *manufacturing and production*
  - Emphasis not on underlying design, but each *individual example* of that design (product)
  - All about **quality control** and satisfying metrics
  - Do the products coming off the production line meet the design requirements?

Hardware Testing (C)

- **Acceptance Testing**
  - Once produced, delivered and installed, does the product (computer) actually work as promised?
  - Are the users/customers happy?

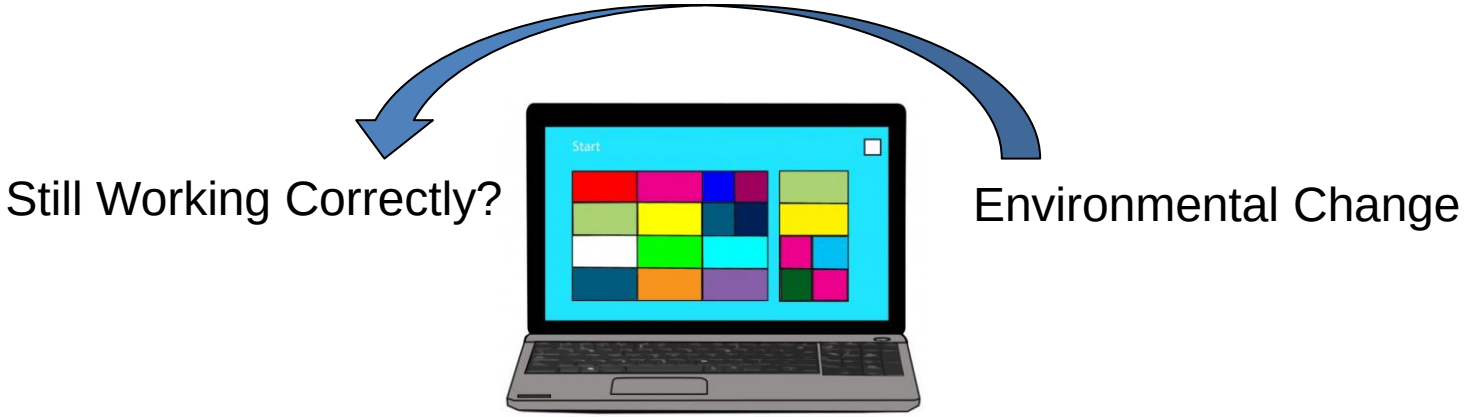
Hardware Testing (D)



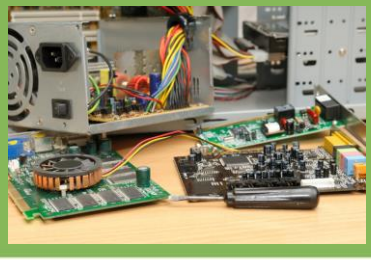
- **Service & Repair/Diagnostic Testing**
  - Over time, physical hardware will degrade or even fail due to wear and tear
  - This can be delayed or prevented by regular services and planned maintenance
  - If not prevented in time, failures will occur and so fault finding (diagnostic testing) will be needed
  - The option then is repair/replace/recycle (see later)

Hardware Testing (E)

- **Regression Testing**
  - Does the original product or component still work as expected when other things around it change?



Fault Detection



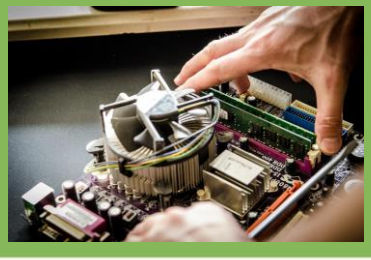
- Strictly speaking, ***fault detection*** means identifying that there *is* a fault
  - It is useful to be able to detect a fault *before* it becomes a problem
  - E.g. a noisy fan may be about to fail, which could lead to overheating and serious damage
  - In practice, most fault detection is done by the user reporting a problem with their PC
  - We are more interested in **fault diagnosis**

Be safe: Data Backup

IMPORTANT

- If the machine will start, back-up the hard disk
- If the machine will not start, take the hard disk out first and back it up using another machine
- Hardware can always be replaced - most data cannot!
- **BACK UP DATA FIRST**

Fault Diagnosis



- **Fault diagnosis** is the identification of the *nature* of a fault and its cause (not that there *is* a fault)
  - Requires a systematic approach
  - Get a clear description of the problem
  - Get a history of the problem
  - Check the most obvious and likely causes first
  - Has this problem been seen before?

Problem History

- A series of questions will enable you to find out about the problem.
  - What are the signs that there is a fault?
    - i.e. what has gone wrong
    - E.g. my monitor is blank
  - How long has this fault been happening?
    - E.g. I have had no image since I switched the computer on today
  - What were you doing at the time the fault started?
    - E.g. I was switching the computer on at the start of the day

Problem History

- Have you had this problem before?
  - Usually the answer is no

- Note that many users do not have the technical knowledge to give clear and accurate descriptions of more complex problems.
  - You may have to visit the user to determine the nature of the problem before you can diagnose and fix it.

Common Causes

- Always check common causes first
  - Is the device plugged in and switched on?
    - Is there a power light on the device and is it on?
  - Is the device connected to the PC?
    - Peripheral connecting cables often come loose
  - Has your PC been moved?
    - Often related to the previous two points
  - Do other devices in your office still work properly?
    - Checking for local power failure
  - If the PC has frozen or crashed, will it restart?

Hardware Fault Finding

- The easiest approach to finding a hardware fault is by substitution:
  - Identify the likely faulty component based on the signs and symptoms
  - Replace that component with a known good one
    - *Do not forget the health and safety issues covered in lab sessions.*
  - Test to see if the fault is still present

Hardware Fault Finding

- Some problems may be caused by a number of different components.
  - e.g. network connection faults may be the NIC, the patch cable, the wiring in the wall, the switch etc.
  - Narrow down the fault by seeing if other machines still work (in which case the switch works – but an individual port can fail, so swap ports).
  - Replace the patch cable, try a different PC in the wall socket, then switch the NIC if possible.

Power On Self Test (POST)

- Every PC does a POST when switched on
- This checks various items, such as systems memory, systems buses, and devices available for booting
- POST has evolved from the original simple system used on early IBM PCs to a complex set of tests
- POST may indicate its result with a series of beeps and if possible messages on the screen

Power On Self Test (POST)

- Different manufacturers use different beep codes:
  - 1 short beep is a POST pass, 2 short beeps indicates an error, with an error code on screen
- Not all POST error reports make sense.
  - “Keyboard missing – press F1 to continue” – this is a genuine error report
- Lots more on POST at:
  - https://www.computerhope.com/issues/ch000607.htm

Internal Problems

- Although modern motherboards have many devices integrated, issues can still arise with devices mounted on the motherboard
  - Make sure that all plug in devices and connectors are correctly aligned and firmly seated
  - Memory and expansion cards (including graphics card) can become loose, particularly if a machine is moved
  - Power and other internal cables can become disconnected, particularly after working inside the PC

Internal Problems

- Swap-in a known good memory or expansion card to identify faulty component.
- Label all faulty components as soon as you find them.

System Documentation

- If your PC was bought as a complete unit, you should have a full set of documentation with it
- If you built your own PC from components, you should still have the documentation for each individual component
- Manufacturers' websites also carry extensive documentation for components/complete systems in the form of 'data sheets'

System Documentation

- Check with this documentation to ensure that configuration settings and connections are correct
- Manufacturers also provide online help forums and support – surprisingly detailed and helpful
- Also, search the Internet for help on particular problems

Test Plan

- A test plan is a document that details a systematic approach to testing
- All testing should follow previously designed test plans
  - Ensures that the correct things are being tested
  - Allows repeatable testing (of same items/issues)
  - Provides documentation for future monitoring and fault finding

IEEE 829 Test Plan

- Specifies the form of a set of test plan documents
  - Focuses on **software** testing (covered soon)
  - Can be modified for whole system testing
  - Like a lot of similar standards, 829's scope is large
  - May be excessive for testing a single PC
  - Essential for major IT projects
    - http://www.coleyconsulting.co.uk/IEEE829.htm
    - http://reqtest.com/testing-blog/how-to-write-a-test-plan-2/
    - http://www.acutest.co.uk/latest-from-us

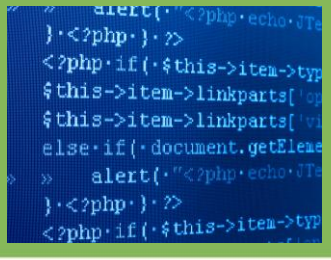
IEEE 829 Test Plan Structure

- Test plan identifier
- Introduction
- Test items
- Features to be tested
- Features not to be tested
- Approach
- Item pass/fail criteria
- Suspension criteria
- Resumption requirements
- Test deliverables
- Testing tasks
- Environmental needs
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies
- Approvals

Example Test Plans

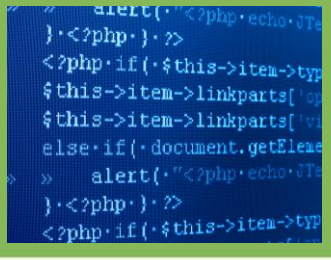
- Sample test plans…
  - http://www.softwaretestingmentor.com/test-case-template/
  - https://www.toolbox.com/tech/enterprise-software/blogs/testing-a-sample-test-plan-022405/

Software Testing



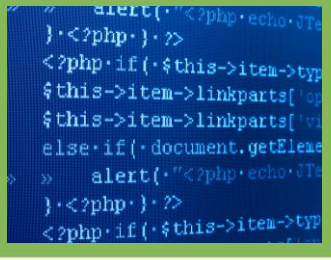
- Unlike hardware, software and the surrounding software can change regularly and rapidly
- Hardware is stable and not designed to change often
- Software is designed to be modified and updated a lot
- Software is also more complex (excepting a CPU)
- Hence, testing of software is more challenging
- Software testing has more types and stages

Software Testing



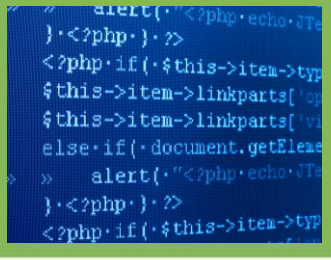
- Many types:
  - **Unit Testing**: Software is written as modules/units – each one with inputs and outputs – so each 'unit' can be tested in isolation. The basic building block of testing.
  - **Integration Testing**: Applications consist of large collections of interacting units – so the fact that they are now interacting (integrated) means that the application as a whole needs to be tested.
  - **Functional Testing**: Does the software as a whole satisfy the original requirements? Does it meet its functions?
  - **System Testing**: Does the completed application work correctly when placed in a larger or differing environments – like different operating systems or different hardware platforms?

Software Testing



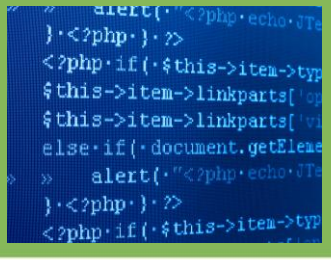
- Many types:
  - **Stress Testing**: What happens when adverse loads or extreme conditions are placed on the system? Could include high volumes of data, extreme data values, high volume of transactions etc.
  - **Performance Testing**: Does the finish system deliver its functionality in acceptable or within specified performance limits, which may be contractual or specified in a Service level Agreement (SLA). For example, an ATM application must respond within 0.5 seconds etc.
  - **Usability Testing**: How good is the system from the human user perspective? Is the application and its interface easy to navigate? Is it easy to learn? Is it user-friendly?

Software Testing



- Many types:
  - **Acceptance Testing**: Maybe the customer themselves wants to run their own tests with their own staff? Just like buying a car or house, you need to check it out for yourself.
  - **Regression Testing**: As stated, software interacts with other software and if that other software is changed, has it affected your software? Does the application still work after other changes are made? ‘Regression’ means to return to a previous state.
  - **Beta Testing**: This is when you deliberately issue a non-finished version out to clients (potential end-users) or the general public and get them to test it for you!

Software Testing



- Many types:
  - **Black Box Testing**: More of a philosophy than a testing category. Whatever size of unit is being tested (single unit, collection of units, whole system), black box testing takes the view that we only need judge/test inputs against outputs. How inputs become outputs is of no concern. Also refereed to as ‘functional’ testing – because it tests only the function provided, not the underlying mechanisms. It looks at the WHAT, not the HOW.
  - **White Box Testing:** Not surprisingly, the opposite of black box testing. This approach does take into account the inner workings and thus the HOW as well as the WHAT.

Repair/Replace/Recycle?

- Whenever anything fails, we have three choices:
  - **Repair**: Identify the fault and, if possible, fix the component while retaining the whole device
  - **Replace**: If it is too complex, time-consuming or uneconomical to repair the device, throw it away and but a brand-new device
  - **Recycle**: If we cannot repair the device or we cannot afford to replace the device with new, we could recycle an existing second-hand replacement

Economics & Eco-Issues

- The option you choose depends on:
  - **Economics**: The normal driver. Which is the cheapest option? The phrase ‘TCO’ (Total Cost of Ownership) factors in the whole lifecycle of a product, including on-going maintenance and final disposal of it…
    - https://www.business-case-analysis.com/total-cost-of-ownership.html
    - http://www.purchasing-procurement-center.com/total-cost-of-ownership.html
    - http://www.gartner.com/it-glossary/total-cost-of-ownership-tco
  - **Legislation**: Maybe the law makes you take certain actions – like WEEE and EU waste disposal law…
    - http://ec.europa.eu/environment/waste/index.htm
    - https://www.gov.uk/topic/environmental-management/waste
    - http://ec.europa.eu/environment/waste/weee/index_en.htm

Economics & Eco-Issues

- Your decisions may also be swayed by:
  - **Culture**: Maybe you are proud to be seen as a community-minded, socially-aware, progressive organization who respects the environment and pushes the green recycling agenda? You hate to throw things away! Then again, maybe the bottom-line and hard cash is everything? You couldn't care less about 'green' issues!
  - **Logistics**: Despite your best intentions, it may just not be feasible to repair those old computers. Maybe the supplier has gone out of business?

Economics & Eco-Issues

- Your decisions may also be swayed by:
  - **Technology**: Much as you like to go green and protect the environment, the existing computers are just too old and not up to the job? They cannot be upgraded for technical reasons.
  - **Competition**: Perhaps it is technically possible and even financially viable to repair and retain the computers – but all your competitors have better IT kit, offer much better services and are stealing your market share! Buy new, or die!

References

- http://softwaretestingfundamentals.com/
- http://www.softwaretestinghelp.com/

Awarding Great British Qualifications

Topic 7 – Systems Testing

Any Questions?