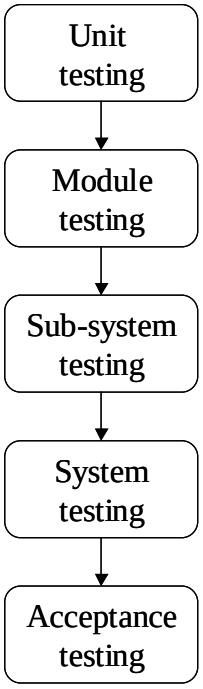
Answers EEE115

Question 1.

a.

To ensure that every aspect of the design has been tested in some way the V.V.T. phase must be related to the structure/hierarchy of the implementation phase. Systems are designed from the top-down (from a concept to many components – see partitioning and decomposition) but they are built/implemented from the bottom-up (modules built using components, sub-systems from modules, whole system from the sub-systems). Thus, applying V.V.T. inline with implementation allows errors or omissions to be more easily identified and the subsequent corrections to be incorporated within the design. The V.V.T. process, therefore, proceeds in stages where testing is carried out incrementally in conjunction with system implementation to test every element of the design at successive stages of assembly. While in any particular case the hierarchy has its own form, it will trace the following general pattern in which the testing process consists of five stages.



1. Unit testing - Individual components are tested to ensure that they operate correctly. Unit testing treats each component as a stand-alone entity that does not need other components during the testing process.
2. Module testing - A module is a collection of dependent components. A module encapsulates related components so can be tested without other system modules.
3. Sub-system testing - This phase involves testing collections of modules which have been integrated into sub-systems. Sub-systems may be independently designed and implemented, and the most common problems which arise in large systems are sub-system interface mismatches. The sub-system test process should

concentrate on the detection of interface errors by rigorously exercising these interfaces.

4. System testing - The sub-systems are integrated to make up the entire system. The testing process is concerned with finding errors which normally result from unanticipated interactions between sub-systems and components. It is also concerned with validating that the system meets its functional and non-functional requirements.

Acceptance testing - This is the final stage in the testing process before the system is accepted for operational use. It involves testing the system with data supplied by the customer (or his agent such as the marketing department) rather than simulated data developed as part of the testing process. Acceptance testing often reveals errors and omissions in the system requirements definition. The requirements may not reflect the actual facilities and performance required by the user and testing may demonstrate that the system does not exhibit the anticipated performance and functionality.

b.

System testing is expensive. For some large systems, such as real-time systems with complex timing constraints, testing may consume about half the overall development costs. So, careful planning is necessary to get the most out of testing and to control testing costs.

Test planning is concerned with setting out standards for the testing process rather than describing product tests. The major components of a test plan are:

1. The testing process - a description of the major phases of the testing process, as described above.
2. Requirements traceability - users are most interested in the system meeting its requirements and testing should be planned so that all requirements are individually tested.
3. Testing schedule - an overall testing schedule and resource allocation for this schedule. This, obviously, is linked to the more general project development schedule.
4. Test recording procedures - it is not enough simply to run tests. The results of the tests must be systematically recorded. It must be possible to audit the testing process to check that it has been carried out correctly.
5. Hardware and software requirements - this section should set out software tools required and estimated hardware utilisation.
6. Constraints - constraints affecting the testing process (such as staff shortages) should be anticipated in this section.

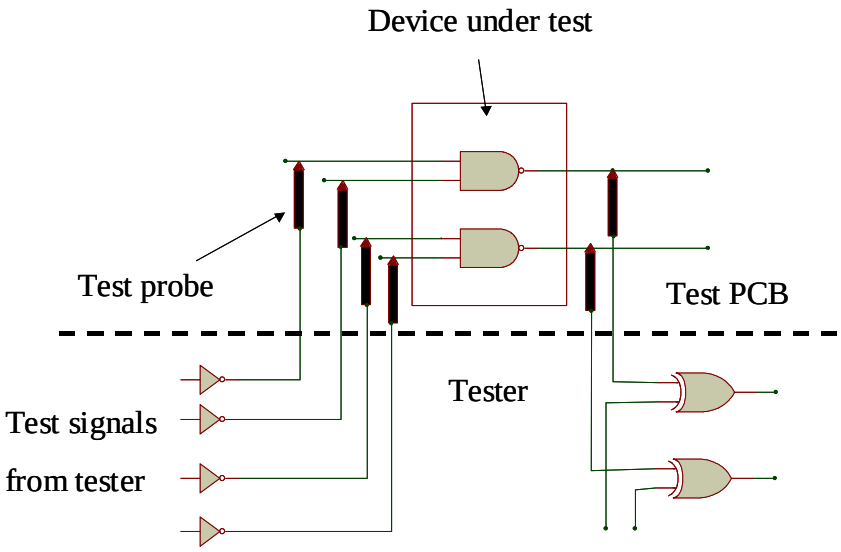
Test plans are not just management documents. They are also intended to be used by engineers and technical staff involved in designing and conducting the system tests. In addition, it must be recognised that the test plan is not a static document and should revised as the system evolves during the design implementation and testing phases.

c.

Since the function of the filter is reject unwanted disturbances one would test to ensure these disturbances are removed. To do this one would test the frequency response of the filter to ensure that unwanted frequencies (i.e. anything above about 50Hz) are attenuated. Owing to the nature of this filter these tests would have to be performed under different loading conditions to ensure consistent performance. In addition to these tests on load testing would have to be performed to ensure the filter components are adequately rated for the system. One may also decide to made linearity tests owing to the presence of a inductor and the possibility for it to saturate.

d.

In many instances, defects are due to opens or shorts on the printed circuit board (PCB). In industry, PCBs are tested using dedicated automated test machines. The Bed-of-Nails tester is machine injects test signal on a PCB or component by the use of test probes. The idea is, a test signal is injected into the device/component under test (D/CUT) and the test machine automatically measures the output response, see below.



Bed-of-Nails testers can be used to test analogue and digital circuitry. In fact, it is possible to test chips in dual-in-line packages (DIPs) with 0.1 inch (2.5 nun) lead spacing on low-density circuit boards using a tester with probes that contact test points underneath the board. Mechanical testing becomes difficult, if not impossible, with board track widths and separations below 0.1 mm, package-pin separations of 0.3 mm or less, packages with 200 or more pins, surface-mount packages on both sides of the board and multilayer boards.

e.

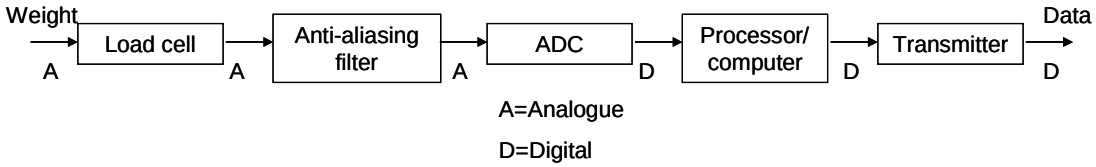
As the number of inputs to a logic gate increases, so does the number of tests required to ensure it is operating correctly. For example, with 2 inputs 4 tests can fully characterise the system. If a gate had 24 inputs over 16 million tests would be required. If a chip contained many 24 input gates it is not unforeseeable that over a million tests would be

required. This problem is known as "the combinatorial explosion". This problem becomes much worse when circuits incorporate sequential logic as well.

If we are to avoid incomprehensively large systems (w.r.t. testing) we need to reduce the complexity to a reasonable level. One of the easiest ways to do this is to segment the circuit into small components that can be tested individually. So we need to partition the circuit in order to avoid having very large numbers of inputs and/or outputs. As an aside, it is worth remembering that this technique is just a manifestation of the general principle of "divide and conquer".

f.
BST is an industry standard testing interface developed for testing digital IC's, associated circuitry and the PCB based on boundary scan test techniques. Using a 4-wire interface test data is loaded onto the IC under test. A predefined serial bit-pattern is then applied to each input pin of the IC and corresponding output pins are then clocked out to test machine or PC using a shift register. The interface also provides facility for cascading a number of IC with JTAG interfaces together so that several chips can be tested using just one interface thereby simplifying the testing of more complex systems.

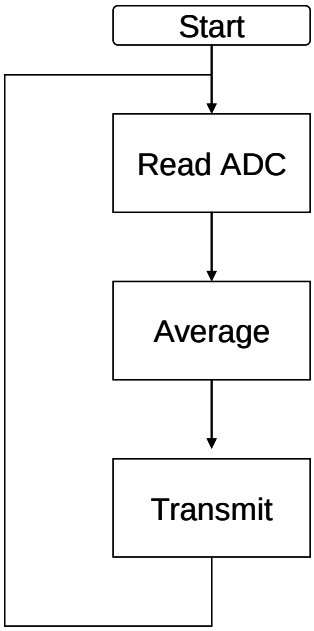
Question 2
a.



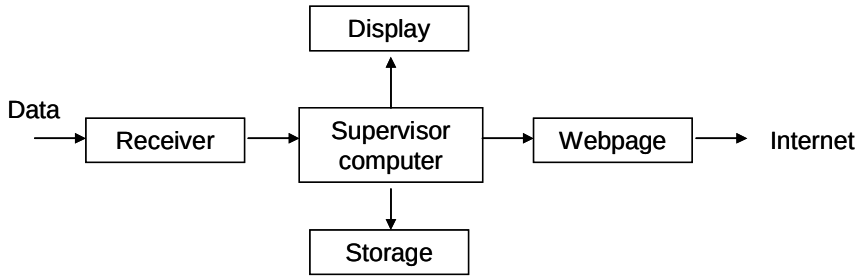
b.
Maximum frequency of interest is 10Hz. Therefore to meet the Nyquist/Shannon sampling criterion the sampling frequency should be 2*fmax=20Hz.

c.
min resolution should be 250kgs so 0.1uV*250=25uV. Min levels=50m/25u=2000. Number of levels=2^N. Therefore N>=11-bits.

d.
Main functions are:Read the ADC, calculate average, transmit average.



e.



f.

Main function are: Receive data from LDASs, process the data, display data, store data, export data to webpage.

In Pseudo code type language:

```
While 1==1 i.e. loop forever
     Index=1
     While not read all LDASs
          Data(index)=Read data from LDASs
          Index=Index+1
     End

     New_data=Process data(data)

     Display(New_data)
     Store(New_data,Time)
     Webpage(New_data)
End
```

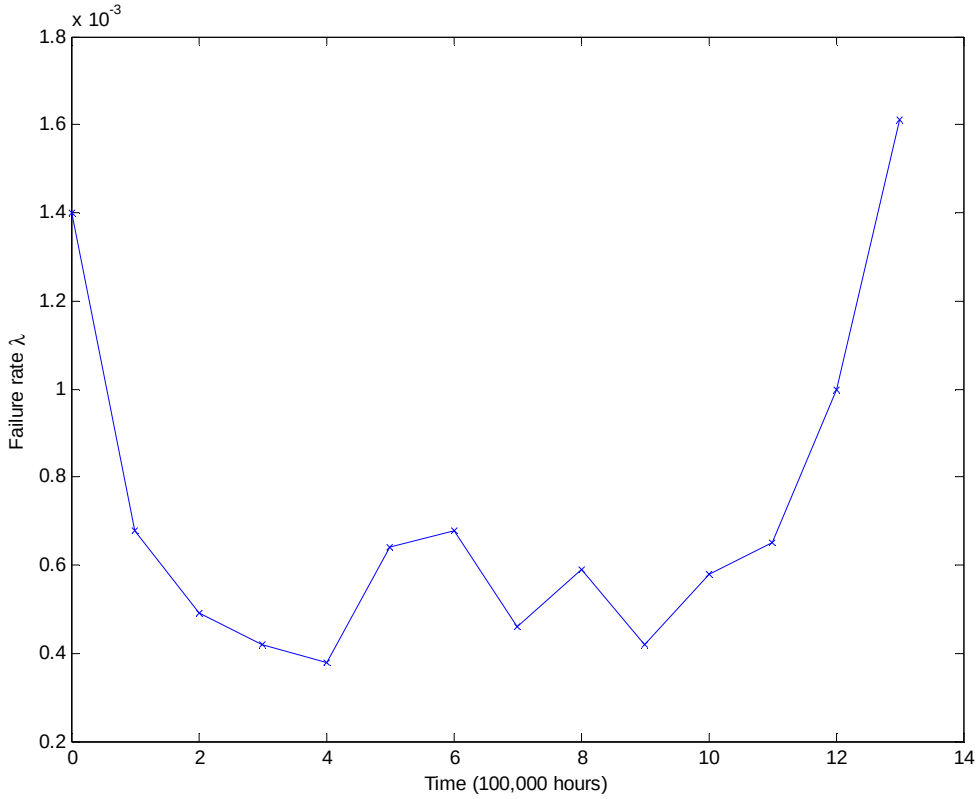
Q3.
a)
Ra=0.6778, Rb=0.1738, Rc=0.0970, Rd=Ra, Re=0.6456, Rf=0.4966

Rp1=Rb+Rc-Rb*Rc=0.2539
Rs1=Rd*Re=0.437
Rp2=Rs1+Rf-Rs1*Rf
Rtot=Ra*Rp1*Rp2=0.1234

b)

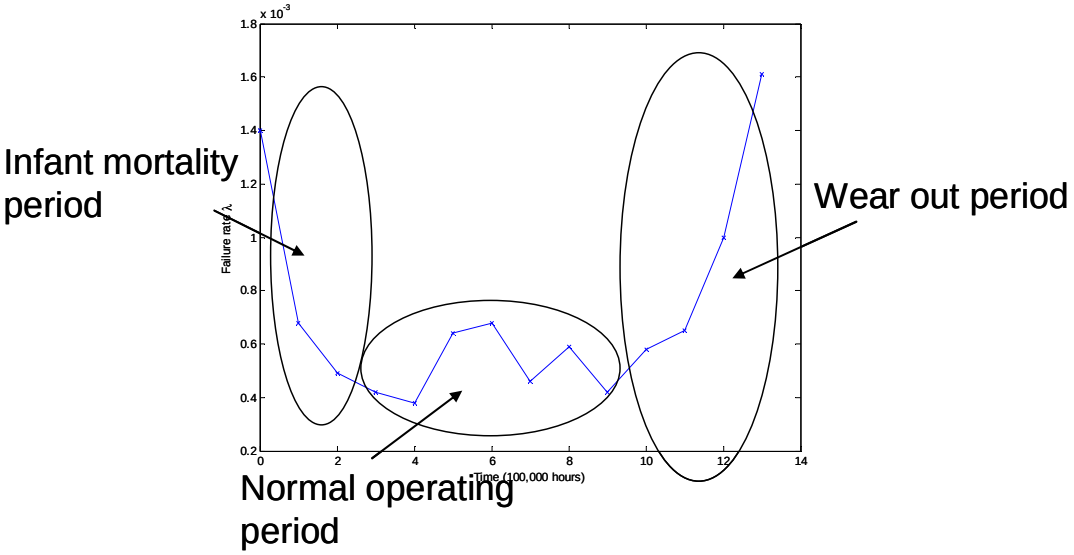
The Arrhenius equation, $e^{-\frac{Ea}{kT}}$, states that a reaction rate of some process is proportional to an activation energy *Ea*, some physical constant *k* (such as Boltzmann's constant = $8.62 \times 10^{-5} eVK^{-1}$) and absolute temperature *T*. If the temperature under which a test is taken is increased the reaction rate also increases. This is the justification for "burn-in" testing where component, unit or system is operated at elevated temperature (or voltage and current) levels to accelerate the failure process. One must always be mindful though to ensure increasing the operating temperature does not introduce extra failure modes into the system that cannot be conveniently modeled by the Arrhenius equation.

c)
To plot failure rate one must first calculate the rate of change between adjacent date points $\lambda = \frac{dFailures}{dt}$. Expect the plot to be a line of best fit not point-to-point as the one shown below. Also looking for axes labels and units.



d)

Expect the periods to coincide with areas outlined below but owing to interpretation and line fitting they may be slightly different.

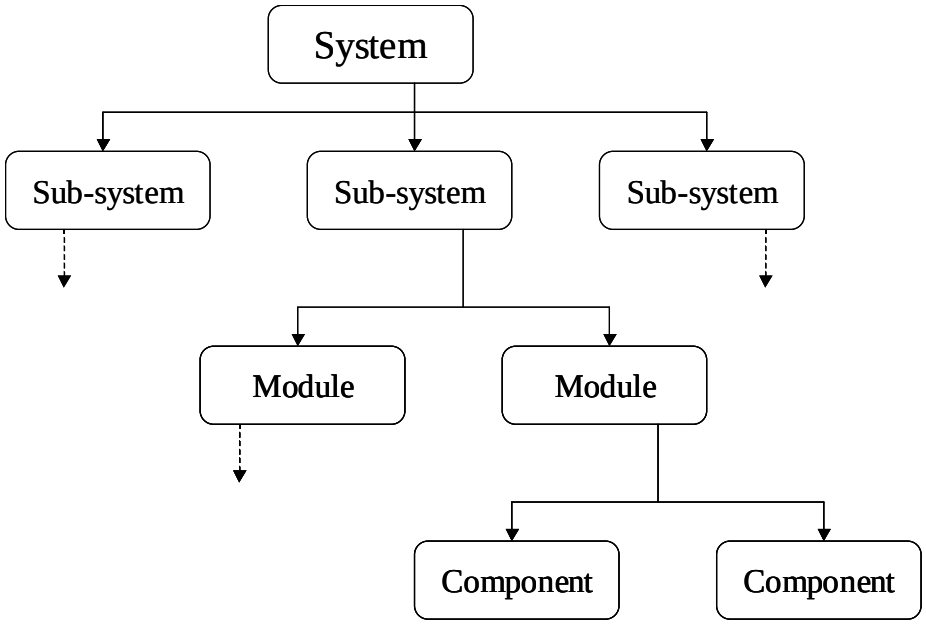


e)

This work be done by drawing a line of best fit through the normal operating period to give a constant failure rate. Typically the rate would be in the range 0.00053-0.00061.

f)
Dependent on e) but somewhere between 0.25 and 0.31.

4.
a.



System

System or system specification is the top-level of our hierarchy. At this level the system is described only in a terms of its external behaviour. Details about the internal behaviour of the system are left to the subsequent levels.

Sub-system

Sub-systems represent divisions in the functionality/requirements of the overall system. Often, sub-system definitions directly relate to the system specification statements.

Modules

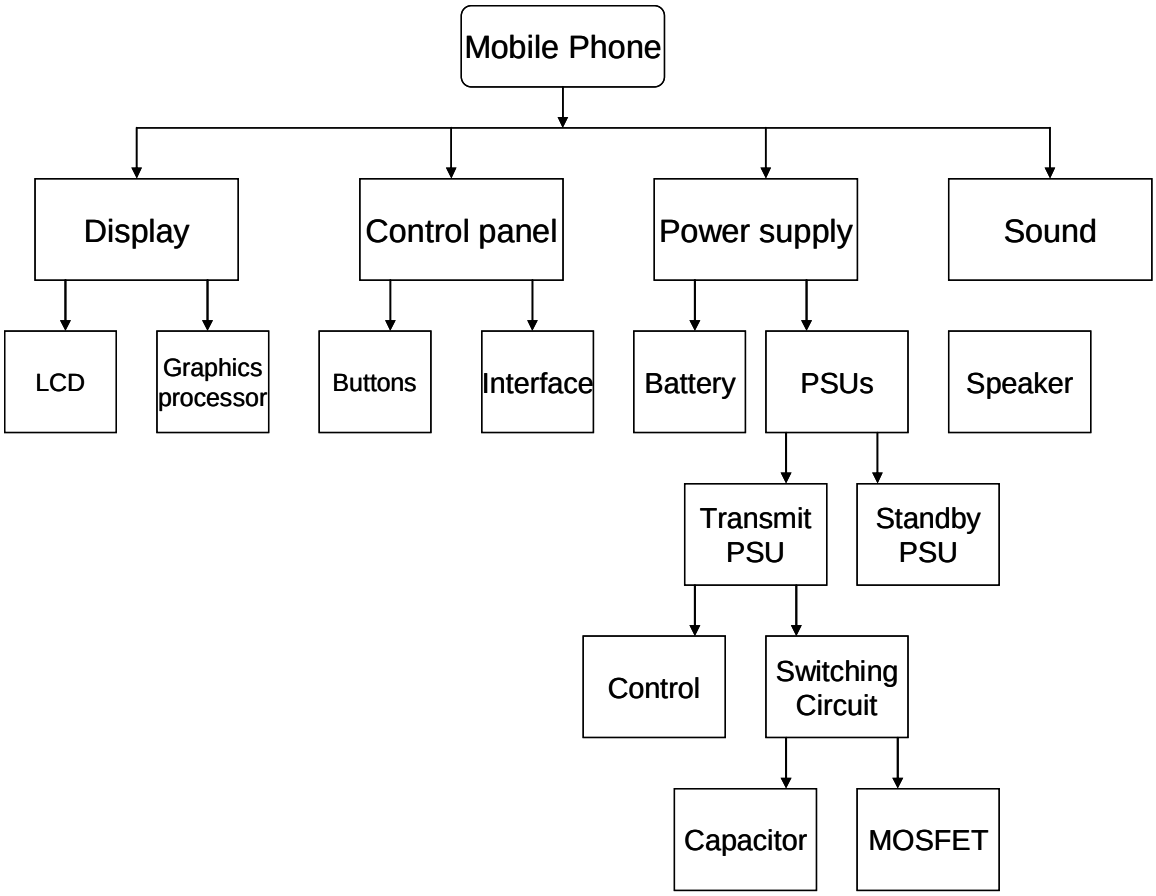
Modules are high-level functional blocks required to construct a sub-system.

Component

Components are the lowest level building blocks for the system. Electronic components and programming language functions fall into this category.

Mobile phone

Could look something like:



b.

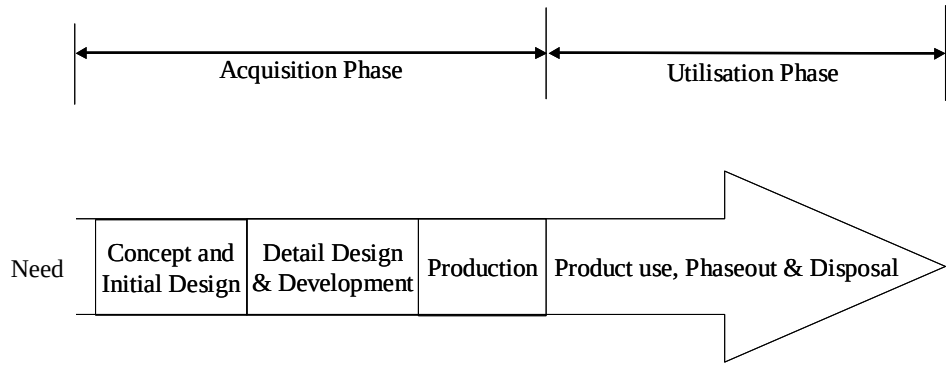
Abstraction is the process of filtering out certain details about a system (sub-system or element) to obtain a more generic description of the system's behaviour. This means an engineer can benefit by transferring knowledge gained in analysing one system to other similar systems even though their final usage may be completely different.

c.

$$\frac{V_{out}}{V_{in}} = \frac{j\omega CR}{1 + j\omega CR} = \frac{j\,\omega/\omega_c}{1 + j\,\omega/\omega_c}$$

d.

In this concept, and many others, the system life cycle is split into two main sections, the acquisition phase and the utilisation phase. The life cycle begins at the acquisition phase where a need is identified. From here, a concept is developed that finally results in the manufacture of a product or service. The product is then put into service, called the utilisation phase, and after a sufficient amount of time, the product is with drawn from service and disposed.



Systems Life Cycle

The waterfall model, represents the discrete stages of the design process as a sequence of activities following one another other in a defined hierarchy. Each stage of the design process is defined as:

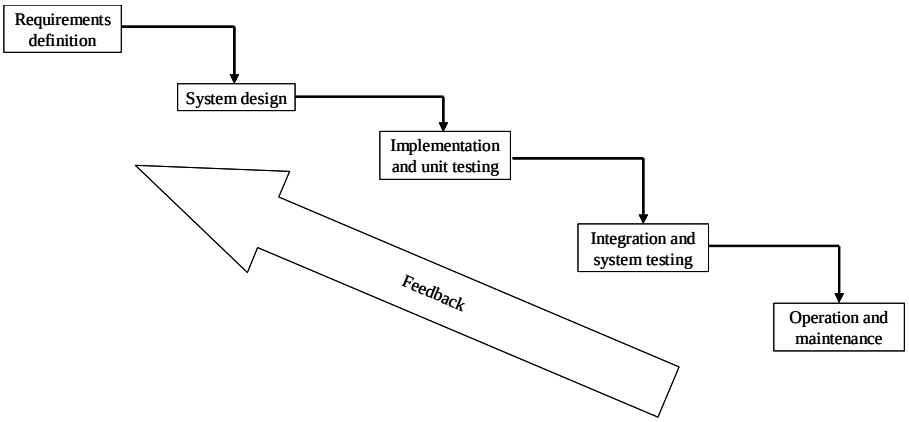
1.Requirements capture
Systems users requirements are defined here in detail.
System specification is developed.

2.System design
The actual system is designed here. Usually involves partitioning the whole system into smaller, more tangible elements/modules suitable for implementation.

3.Implementation and unit testing
Individual modules (units) from the design stage are constructed and tested in isolation to ensure each unit meet the required specification.

4.Integration and system testing
Here the individual system modules are integrated to form the whole system. The whole system is tested to ensure the system meets the specification.

5.Operation and maintenance
The system is installed/commissioned and put into practical use. Normally, throughout the life of a system, the system is maintained to remove errors not initially detected during the testing phases or to correct for errors or omissions from the specification or design stages. In addition, the system may be continually updated, as the end-user requires extra functionality.



Waterfall model

As can be seen, each activity feeds information forward to the next and although it is not explicitly shown, information is fed back to the previous stage(s) to update the entire system design (including the specification) in light of extra information that was not at first obvious during the initial design phase. In Fig. 4 the formulation, analysis and interpretation steps that are generally conducted during each design phase is clearly shown along with the feedback interactions within each phase and between each design step.

e.

WEEE and ROHS define legislation that restricts the use of certain materials and components. In order for certain products to be sold commercially they must adhere to this legislation. The system design must be sensitive to these needs and ensure that their design (or parts of it) is compatible with the legislation. This can mean anything from obtaining certificates of conformity to scheduling tests. Thus, the requirements, design and implementation phases are considerably affected by these restrictions.