Chapter 4 AS The systems life cycle

You already have some basic knowledge of the stages in the system life cycle:

- feasibility study
- analysis
- design
- development and testing
- implementation
- documentation
- Evaluation.

In this chapter you will learn about ICT systems and how they created and developed. You will also be introduced to the scenario-based examination papers that are use by CIE.
You will learn in more detail about :

- the stages of the systems life cycle
- the different methods of researching a situation
- the steps involved in designing a new ICT system
- developing and testing a new ICT system
- implementing a new ICT system
- the development of the documentation of a new ICT system
- Evaluating a new ICT system.

Stages of the systems life cycle:
With modern day computing it is more likely that there is one of two possible situations:

1. One is where a system is computerised but may be out of date and need replacing.
2. The other is where a small company has a computer system in place but it is very limited and could be improved upon to allow many more aspects of the business to become computerised.

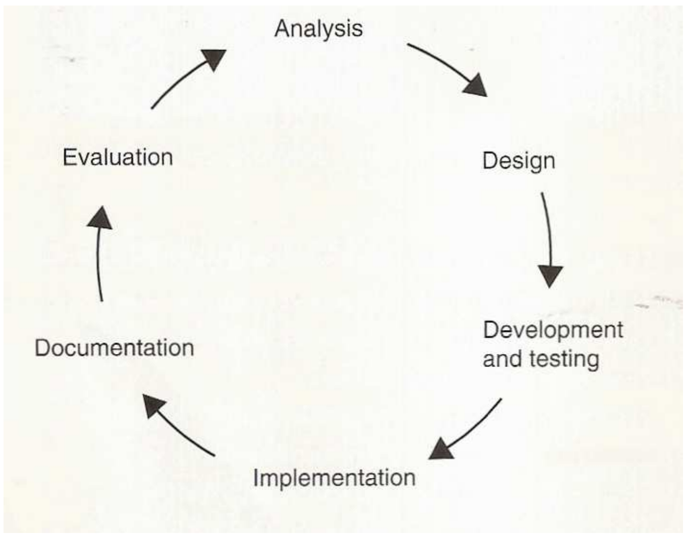
The examination papers for this course are based on scenarios — ‘scenario’ is just another name for situation — that are developed specifically for the examination. Because the subject is Applied ICT we look at how ICT is applied in real situations rather than just studying the theory of it. We are going to use the scenario approach to look at the systems life cycle.

We are going to look at a building supply company scenario and consider the need for an improved ICT system. Let us consider a fictional company called Biashara Street Building Supplies in Nairobi, Kenya. The company buys in bricks, cement, gravel, sand and roof tiles from big companies and then sells them in smaller quantities to local house builders. The company is run by two directors called Peter Kimanthi and Irene Kibaki. It employs two secretaries, three sales people and two truck drivers. It has a computer, but this is mainly used for creating word-processed letters and sending emails to customers. Its only other use is to keep records of the company’s customers on a database. One secretary, **Josephine**, deals with the workers’ personal information and is also in charge of keeping customer details. The other secretary, **Mary**, keeps information about the hours worked by the workers and also processes orders from customers. Irene is in charge of the paying of the workers.

Chapter 4 AS The systems life cycle

Peter and Irene feel that they **could make better use of their computer system and need a systems analyst to look at how the computer is used**, and to advise them whether or not their business would improve if they made increased use of the computer They have invited Daniel Mathuru(**system analyst)** to perform this role.

Let us look at how Daniel will approach the situation.
The systems life cycle is so named because it is a circular process (see diagram). **There is no real start and finish point**, because after a new system has been evaluated this leads to further improvements being needed and so the whole process starts over again.



The purpose of analyzing the existing system is to see how data flows around the system. If you think of the system as being the company itself, in simple terms analysis is **to see what data comes in to the company**, how it is processed and what outputs they produce. Only after this has been determined will the analyst decide on what sort of computer system should be implemented.

4.1 Systems analysis

Systems analysis involves examining the existing system in detail, in particular:

• collecting information on how the existing system works
• establishing its inputs, outputs and processing
• Recording information, for example in the form of data flow diagrams
• identifying problems.

Having carried out these stages, the systems analyst then needs to:
• identify suitable hardware and software for a new system
• identify user and information requirements.

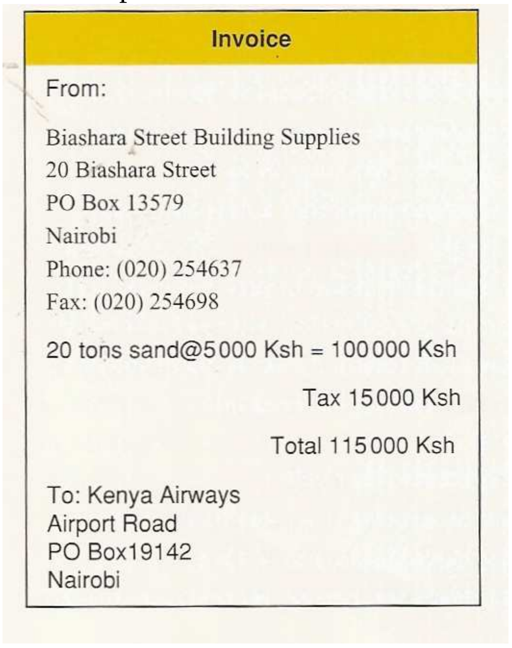
Collecting information

The first stage of systems analysis is collecting information about the current system. There are four methods of doing this:

• Examination of documents
• Interviews
• Observation
• Questionnaires.

Examination of documents

All documents that are used within a system need to be examined. The documents may be, for example, bills, invoices, letters, order forms, payslips etc, after the analysis, if it is considered necessary, they will be replaced with ones produced by the computer system.
An example of a document:


Invoice

From:

Biashara Street Building Supplies
20 Biashara Street
PO Box 13579
Nairobi
Phone: (020) 254637
Fax: (020) 254698

20 tons sand@5000 Ksh = 100000 Ksh

Tax 15000 Ksh

Total 115000 Ksh

To: Kenya Airways
Airport Road
PO Box19142
Nairobi

This method of collecting information will help the systems analyst **to identify the inputs and outputs of the system as it operates at the moment**. He can then assess the processing that the computer system will need to carry out. Using the payslip as an example:

• The new variable each week is the number of hours worked — this is the input
• The outputs are all the figures that appear on the payslip
• The processing is the use of the input and other, stored figures to perform calculations to produce the outputs.

The other use the systems analyst will make of these documents is to calculate the number of documents that are processed and the volume of data on all the documents. This is simply the amount of data on one document multiplied by the number of documents of that type. This would be calculated for each type of document. The systems analyst would use this information to decide on the size of memory as well as the types of input and output devices needed to cope with this volume of data.

Questionnaires

This is perhaps the most common form of collecting information. The way questions are phrased is very important, as the way a question is asked can influence people's answers in the questionnaire. For example, Daniel may want to know what information Peter and Irene would like to see on a computer-produced payslip. A question along the lines of:

What details of the workers and their pay would you like to see on your workers' payslips?

Is less likely to help the systems analyst to produce a payslip design than if the question was asked in this way:

Which of the following information would you like to see on a worker's payslip:

O worker's name
O worker's number
O rate of pay
O hours worked
O income tax rate
O income tax paid
O any other information?

This would tend to produce more precise information for the systems analyst than trying to get Peter and Irene to produce a list which might include information that is not really relevant. This is where the systems analyst's previous experience of designing documents such as invoices and payslips will help the directors.

An advantage of questionnaires is that although it takes a lot of time to produce an effective one, once it is produced as many copies as you want can be given out. For example, Peter and Irene might decide at the last minute that the workers ought to be asked what information they would like to see on their payslip. With any other method it would take a lot of organizing to get the information, whereas with a questionnaire it is just a matter of producing extra copies and distributing them.

A disadvantage of questionnaires is that, because they are impersonal and can be anonymous, workers might exaggerate their answers as they know there is no comeback. In our scenario, however, because there are so few workers involved, they would be fairly easily identified.

Interviews

This method is used in every situation, but because of its nature it has a very limited format. Because it takes time to complete an interview it is not possible to interview every worker. Instead, interviewing is a technique that is used with key personnel and representatives of the other workers.

An advantage with interviews is that they are flexible. With questionnaires it is very difficult to ask further questions based on the response to another question. With interviews this is straightforward. The interviewer can move away from their 'script' and ask a more in-depth question if a particular response is given. A questionnaire cannot be adapted like this without a great deal more time being spent on altering and redistributing it.

Disadvantages:

As with questionnaires, a **lot of time is spent on producing the most appropriate set of questions. It also takes a lot of time to organise an interview**.

The workers or directors have to be available at the time the systems analyst wants to interview them. This is not always possible and compromises often have to be reached. The systems analyst has to be very flexible in all this and must try to accommodate the busy working schedule of the people who are to be interviewed.

There are other drawbacks with interviews. There is a temptation for certain interviewees to give not very accurate answers. They may try and provide answers that they think the interviewer wants to hear rather than giving accurate responses. In contrast, as a questionnaire can be anonymous answers tend to be, on the whole, more accurate.

Another drawback with interviews is the time taken to complete interviews with many people. This compares unfavorably with questionnaires, where everyone can complete the questionnaire in the same amount of time instead of one after the other.

Observation

There are many situations where the three methods described above do not provide the full picture that the systems analyst requires.

All the other methods give information about an individual's role within the business but do not really provide information about how separate tasks overlap and how workers interact, or even if the methods being used are efficient.

Observation involves the systems analyst just **watching all the activities going on in the office**. For example, it may be necessary to see how the data comes in about a customer and how that is processed and used to produce an invoice. A number of tasks performed by individual workers may involve the inputs to the system, such as recording new sales to customers, while another worker maybe involved in processing that data so that an invoice can be created.

Observation will enable the systems analyst to see the process as a whole. From this, a data **flow diagram can be produced that will enable the analyst to determine the inputs, outputs and processing which exist in the current system**.

One drawback of observation is the 'Hawthorne effect' this is when some people who know they are being observed change the way they work. They may start to work more efficiently than normal, which could lead to misleading statistics being collected by the analyst.

Choice of method

The choice of method for collecting data about the existing system depends on the type of information being collected and also the practicality of using the method in the situation presented to the analyst. If, for example, there were hundreds of employees it would take too long to personally interview each one. This would also be the case where employees are spread over a number of areas, for example with companies such as banks that have separate branches in different towns. In this case, questionnaires might be a better method of collecting information from the workers.

In our scenario, there are very few employees and so, in addition to examining the documentation of the company, any of the other three methods might be appropriate. However, in reality different methods would be used for different employees:

- Peter and Irene would need to be interviewed as the owners of the company, so that their specific needs could be established. These needs might be difficult to discover in any other way.
- The two secretaries may need to be observed, as their roles involve different aspects of the work. Asking them questions may make it difficult to discover their precise roles, but these would be fairly easily seen using observation.
- The sales people are very busy all day as their job entails speaking to customers most of the time. It might be easier to give them questionnaires and collect their responses at a later date. This would mean that they had enough time to complete them, rather than having a possibly rushed interview.

• The van drivers would rarely be in the office as they spend most of their time driving between customers delivering the building materials. Again, questionnaires might be the most appropriate method for them.

It is sometimes a good idea when collecting information about an existing system to use more than one method with each type of worker, although this particular scenario does not necessarily call for such an approach. However, the very nature of the business involves the production of many orders and invoices as well as a regular, although small, need for production of payslips. Because of this, as with the majority of, if not all, companies, it would be essential to examine the paperwork and documents used in the system.

Establishing the inputs, outputs and processing in the existing system

After the systems analyst has finished collecting information about the current system he needs to identify all the inputs, outputs and processing in the existing system. By examining all the documents used in the current system it can be established which relate to information coming into the system and which relate to information going out. This will then enable the analyst to produce documentation of the system, as opposed to the company's documents that the analyst has been looking at so far. Quite often this stage is done whilst producing a data flow diagram.

Each section of the system will need to be examined to see what specific inputs, outputs and processing are required. For the payroll system, for example, the input would be the details of the workers, the processing would be the calculation of the payrolls and the output would be the payslips. Each part of the system will be examined and broken down into these three elements.

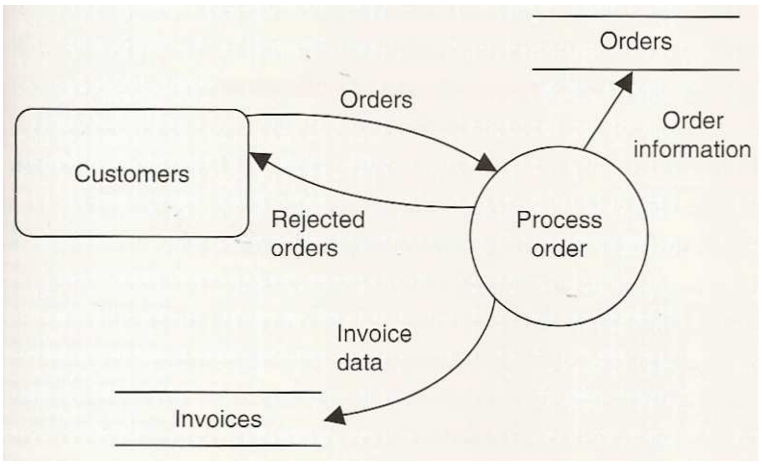
Recording information about the current system

Whilst carrying out the fact-finding methods, the systems analyst needs to record all the results in order to establish all the relevant features of the existing system. It is always important to keep accurate records of systems analysis since the system will continually evolve and systems analysts and programmers will need to develop the system even further.

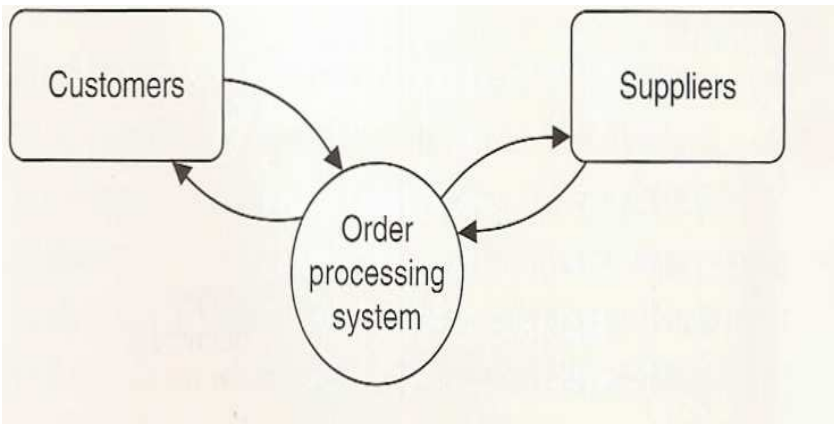
There are a number of ways of formally recording the flow of data, but the use of data flow diagrams is the most popular. Data flow diagrams are a graphical method of recording the inputs, outputs and processing that have been identified.

Data flow diagram consists of four components:
Terminators, processes, flow arrows and stores. Look at the very basic data flow diagram below showing how Peter and Irene deal with customer orders.

1. It shows that the customer sends in an order to the company.
2. It is checked to see if it has all the information required such as customer name and address and an order for an existing product.
3. If does not, it is rejected and sent back to the customer.
4. If it does, the order is processed and the order information is printed and filed. In addition, an invoice is generated and filed ready to send to the customer.



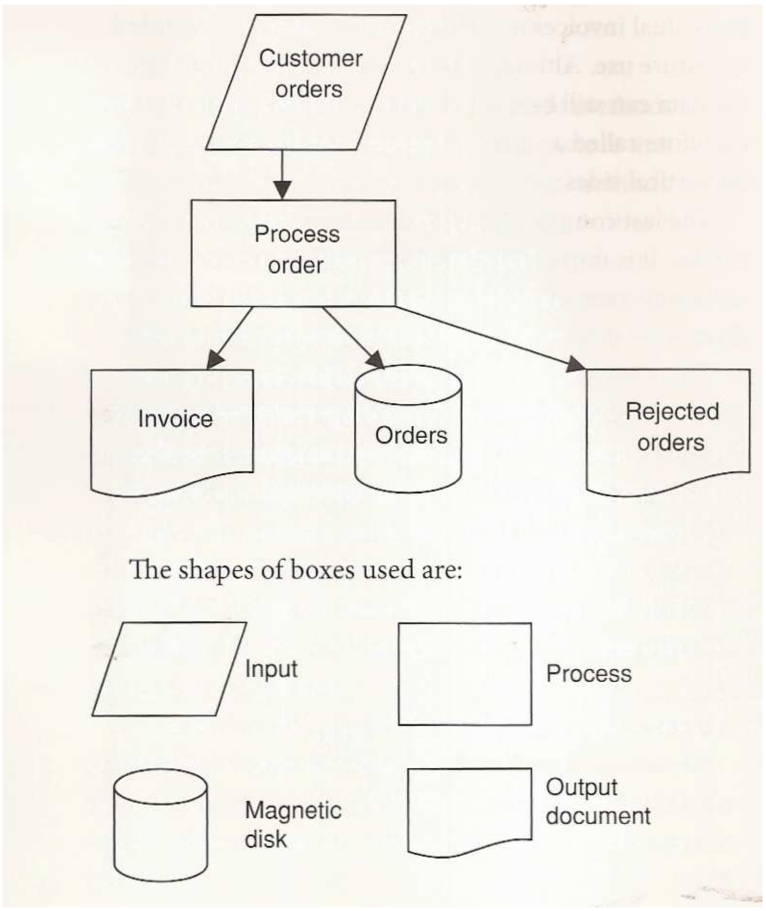
The order has come from the customer; customers are not part of the process order system and so are placed in a terminator. When data flows from or to somebody or somewhere outside the system, that somebody or somewhere is called a terminator. Here we are using a rectangular box with curved corners for a terminator.
The data from the customers are the orders. This information is processed to produce the invoices. The process order is put in a process box, shown with a circle.
The actual data output from the system, such as the individual invoices and the printed orders, are recorded for future use. Although this is not kept on computer, the data can still be viewed as being stored. Such data is therefore called a store and placed inside a rectangle with no vertical sides.
The last component is the data flow. These are the arrows. It is important that the direction of data flow is accurately recorded and that each arrow is labelled to show what data is flowing at that point in the diagram.
There are different levels of data flow diagram. The **context** level or **level 0 data flow diagram** is basically a diagram showing a very generalized diagram with the terminators linked to the current system as a whole. A very basic one might look like this.



The next level is **level 1**. It would have many more processes, with more detail about all aspects of the system. It would have the process boxes for receiving orders and also for producing invoices. The next level would be **level 2** — there would be many of these, each showing parts of the level 1 diagram in much more detail.

Another way of representing the data flow diagram for processing customer orders (shown above) is by using a system **flowchart** as in the diagram below. This, however, is generally a method of designing a systems solution and so is not found very frequently in the analysis stage. Notice that the orders that are considered to be a store in the data flow diagram have to be represented as being stored on a storage medium in a computerised system.

The invoices, which will actually be sent to the customers, are now considered to be output documents ?this is because the system flowchart is designed to represent the new computerised system, whereas the data flow diagram represents the old, mainly manual system. In a manual system, keeping information field on paper is equivalent to storing the data on a computerised system.



Identifying problems with the current system:

Another use of data flow diagrams is to help the systems analyst to identify problems with the current system. By examining the system in great detail, many weaknesses can be identified. In the payroll example above, it might seem unnecessary to have both secretaries involved in entering data into the payroll process. In a new system, these two files of information would need to be linked to make processing easier. By showing the stages of a process in great detail it is easier to see where there is job duplication or unnecessary time wasted in gathering data. After the data flow diagrams have been produced, they are examined together with the other results of analysis such as interview transcripts and questionnaire analysis to identify the problems with the current system.

Identifying suitable hardware and software for a new system

The actual hardware and software will not be recommended at this early stage in the systems analysis. However, having decided upon the required outputs, storage and processing requirements of the system using the data flow diagrams, the systems analyst will be able to make generalised recommendations for the software and hardware.

Daniel will know the volume of data being input to the system from the terminators and flow arrows in the data flow diagram. This will allow him to determine an appropriate method of input to the system. He will have looked at all the processes that occur and this will give him an idea about the size and speed of the processor required. The stores in the data flow diagram will give him an idea of how much data needs to be stored and this will help him recommend the size and number of storage devices. The terminators and flow arrows coming out of the system will also indicate the quantity and format of the output.

Identifying the user and information requirements

When the new system is developed it will be essential to involve all workers in the process of design. The new system must meet the needs of the people who will be using it.

The process of collecting information about the existing system will have been very important to see exactly what job each worker is doing. Daniel will have interviewed Peter and Irene to discover their requirements for the new system. In addition, he will have recorded his observations and used the data flow diagram to come to a conclusion about the user requirements in general. From this, he will have produced a requirements specification. This will be a list of the features of the system that are required. It will contain general requirements such as what the user wants the overall system to do, i.e. produce the payroll, deal with orders, file customer information, and so on. It will also include specific requirements such as, ‘I want the system to find me the details of an individual customer quickly’ or, ‘I want the system to produce the payroll overnight so I can just set it to print at the end of one day and it’s there for me the next morning:

4.2 Design

Having analyzed the existing system, the next stage is to design the new system. The systems analyst, Daniel, may involve the use of a programmer (or programmers) at this stage. A programmer is a person who will actually write the software, if new software has to be written. Daniel, together with the programmer, if needed, will have to design:

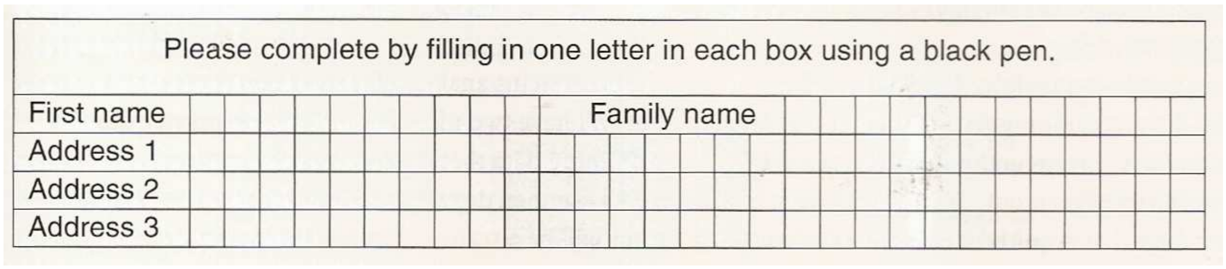
- the inputs to the system
- the outputs from the system
- the files and/or databases needed to store the data
- the processing required to produce the outputs
- any validation checks that will be needed
- the data needed to test the system.

In addition, the analyst will need to specify the hardware and software needed to form the system.

Designing data collection forms and screen layouts

Great care must be taken in producing data collection forms and screen layouts. Data collection forms can be either hard copy or screen based. In our example, the drivers have rarely needed to use the computers. It might therefore be best, if data is to be collected about them for entry to the personnel file, that they complete hard copy data collection forms.

When designing a data collection form, the analyst must make it easy both for the worker to fill in and for the secretary to read the information. One of the most common methods for ensuring this is to put boxes in each section to be completed. The rule is to put one box for every character of required input. The use of these boxes will mean that the worker is likely to make fewer errors when filling in the form. Also, as the form will be easier for the secretary to read, errors are less likely when typing the data in. The requirements for completing the form should be clear to workers, so that they know which sections to complete. An example of a basic form is shown below.



Please complete by filling in one letter in each box using a black pen.

First name		Family name	
Address 1			
Address 2			
Address 3			

The design of these forms will depend on the user requirements as well as output required from system and file structures.
When designing input screens, the systems analyst or programmer must consider a number of factors:

1. Each screen will need to be set out in such a fashion that it is easy to use.
2. It will need to be attractive to look at.
3. It will need to limit the possibility of inaccurate data being entered.

In our scenario, Daniel will need to make sure that Mary and Josephine are not distracted by an **over-elaborate screen design**. **The use of lots of different colours and fonts should be avoided, otherwise they will find it difficult to focus on their work and this may lead to them making mistakes.**

4. The screen designs must contain guidelines to workers on how to fill in any data entry forms as well as allowing them to navigate from one screen to another without any difficulty.

5. Decisions regarding the type of input devices may well be taken at this stage.

The choice of input design will influence the choice of input devices. For example, it might be decided that each type of building material should be given a barcode. This would mean that one of the input devices would be a scanner or other hardware capable of inputting a barcode.

Designing report layouts and screen displays

When designing report layouts and screen displays, the systems analyst and programmer will be heavily influenced by the views of the users and what the systems analyst has agreed are the users' requirements. Daniel will decide on what outputs are required by looking at existing documents and examining the results of their views that he carried out.

The two main aspects of the way output is designed: are the format of the output and the medium it is to be produced on. The formats are likely to be one or more of:

- graphs
- lists of records
- reports
- Tables.

The medium that will be used will be one or more of:
to see that they advertise the company

- paper
- screen display
- Sound.

Daniel will have spoken with Peter and Irene about their requirements. This and the examination of existing documents will have probably suggested to him that the invoices and payslips could be produced using existing database software to produce reports. For other parts of the system, there will need to be other types and forms of output.

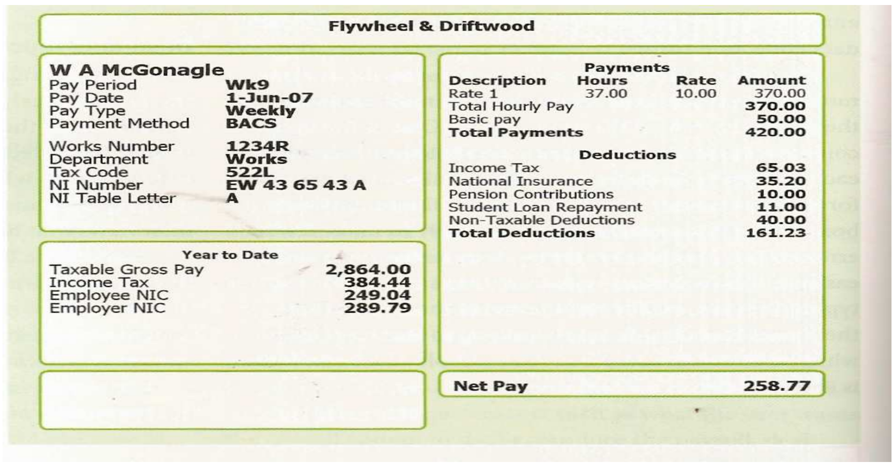
For the layout of the documents, the systems analyst needs to consider who will see them. There will also need to be consultation with the owner(s) of the company. Peter and Irene in our scenario will want their customers to gain a good impression of the company. They will want favorably and contain all the required information in an orderly, easy-to-read format. Furthermore, when designing invoices Peter will know that the style and content of the invoice will have to match the needs of the customer as well as the company.

As far as screen output is concerned, the usual rule is to keep it as simple as possible. The users of this system will be the secretaries and sales people and they will not need to have the company advertised to them. It is not necessary for screen output to contain any extra material other than that required.

Each screen of output must have a consistent theme so that the user does not get confused by changing appearance, instructions on how to navigate between screen should be included on the screen, as stated above there are different available formats, and these need to be relevant to output produced and what the user is comfortable with.

Designing the required data/file structures and processing

Although the steps outlined above are in order of how they should happen in theory, in practice the design of the processing would probably occur at the same time as designing the files and databases. In order to produce a data structure, the systems analyst will have to produce a systems flowchart or similar. The programmer will break then down parts of the systems flowchart into algorithms or program flowcharts.
If we look at a typical UK payslip, we can see that the processing is fairly straightforward.
The type of payslip we will be looking at in this section is shown below, although the payslip for our scenario is simpler than this.



Flywheel & Driftwood

W A McGonagle

Pay Period	**Wk9**
Pay Date	**1-Jun-07**
Pay Type	**Weekly**
Payment Method	**BACS**
Works Number	**1234R**
Department	**Works**
Tax Code	**522L**
NI Number	**EW 43 65 43 A**
NI Table Letter	**A**

Year to Date

Taxable Gross Pay	**2,864.00**
Income Tax	**384.44**
Employee NIC	**249.04**
Employer NIC	**289.79**

Payments

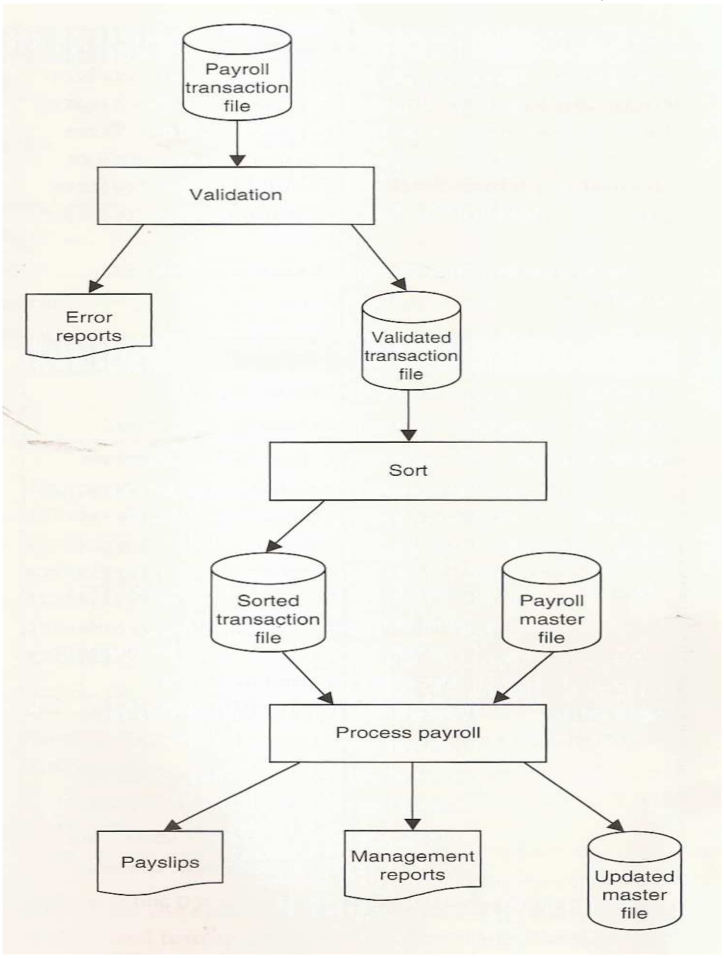
Description	**Hours**	**Rate**	**Amount**
Rate 1	37.00	10.00	370.00
Total Hourly Pay			**370.00**
Basic pay			**50.00**
Total Payments			**420.00**

Deductions

Income Tax	**65.03**
National Insurance	**35.20**
Pension Contributions	**10.00**
Student Loan Repayment	**11.00**
Non-Taxable Deductions	**40.00**
Total Deductions	**161.23**

Net Pay **258.77**

The systems analyst will have to design a structure that will have two files. One will be the master file containing data that does not change often, such as name, works number, department, hourly rate, and so on. The other will be a transaction file containing the data that changes each week, such as hours worked. These two files will be processed together to produce the payroll. HOW?
Before the transaction file is used in combination with the master file, it will need to be checked for errors and sorted in the same order as the master file (as discussed in Chapter 2). Magnetic disks are now used much more than magnetic tapes, which tend only to be used for backing up systems. The transaction file will therefore be stored on disk, even though it holds data in sequential order.



System flowchart showing the processing

Having decided on the need for these files, the systems analyst will then decide on individual file structure and whether any programming is required. **He will need to look at the following attributes of the master file:**

• Field names
• Field types
• Field lengths
• Validation rules
• Field descriptions
• Selection of key field.

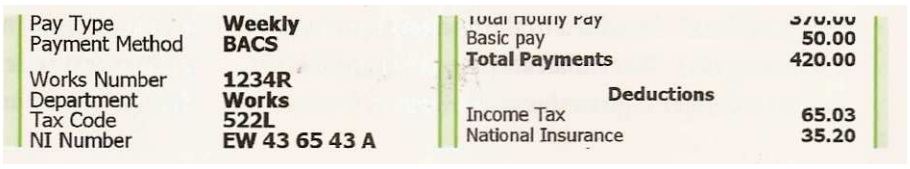
The analyst will also want to design a test plan: the files will need to be tested, as you will see in the development section.

Designing validation routines

When using a computer system, data entry is probably the most time-consuming process compared with processing, storing or outputting data. It is therefore important to try and ensure that the number of errors, which will lead to retyping some of the data, is very small. This can be done by using validation.
In order to ensure that data input to the system is valid it is essential to incorporate as many validation routines as possible into the system. The number and types of routines or checks will obviously depend on the form of input and also the file structures that are being used in the system. Not every field can have a validation check. For example, there are so many variations of peoples' names that this would be very difficult to validate. Some fields will be calculated fields and so will not require a validation check that has not already been built into the calculation. We will only be looking at validation checks which can be used on single items of data individually.

Below are some of the fields on a typical payslip, with examples of data. Many will be stored on the master file but when the master file was set up the input data would have had to be validated and any new records would also require validation.

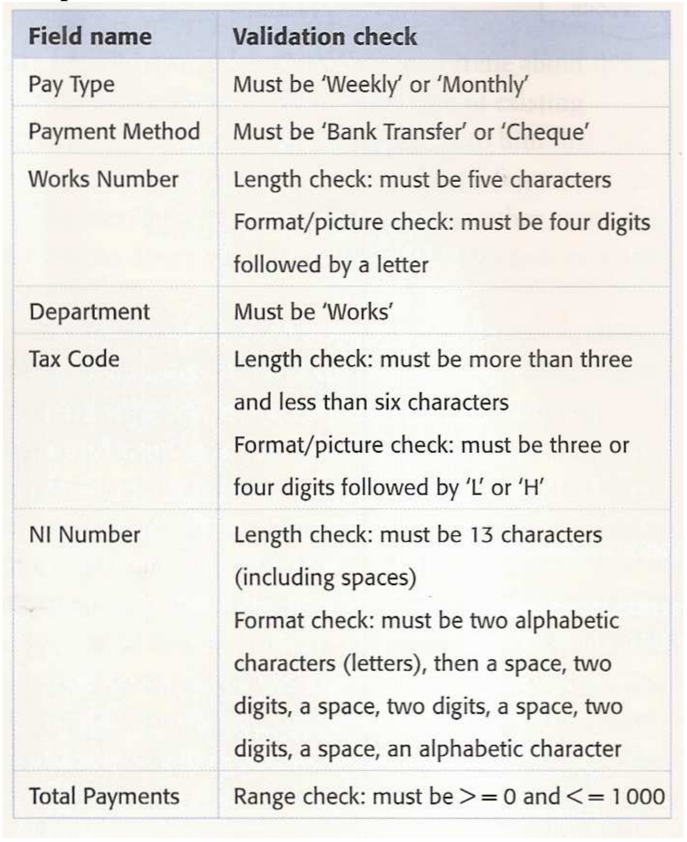
Fields which contain numeric data tend to have range checks designed for them. Using the payslip below, we can see that Total Payments are £420.00. It is unlikely that any worker will receive more than £1 000.



Pay Type	**Weekly**	Total Hourly Pay	370.00
Payment Method	**BACS**	Basic pay	**50.00**
		Total Payments	**420.00**
Works Number	**1234R**	**Deductions**	
Department	**Works**	Income Tax	**65.03**
Tax Code	**522L**	National Insurance	**35.20**
NI Number	**EW 43 65 43 A**		

The range check we could put on this field is that Total Payments must be less than or equal to £1000. Text fields can have length checks put on them. The Works Number has five characters and cannot be more or less than this. There are other validation checks which can be used such as a format or picture check. This would be used to check that the Works Number has four numbers followed by a letter.

The possible validation checks are shown in the table below:



Field name	Validation check
Pay Type	Must be 'Weekly' or 'Monthly'
Payment Method	Must be 'Bank Transfer' or 'Cheque'
Works Number	Length check: must be five characters Format/picture check: must be four digits followed by a letter
Department	Must be 'Works'
Tax Code	Length check: must be more than three and less than six characters Format/picture check: must be three or four digits followed by 'L' or 'H'
NI Number	Length check: must be 13 characters (including spaces) Format check: must be two alphabetic characters (letters), then a space, two digits, a space, two digits, a space, two digits, a space, an alphabetic character
Total Payments	Range check: must be >= 0 and <= 1 000

In our scenario it will be simpler as there will be no fields such as Works Number, Department, etc., and there are many fewer deductions in terms of pension contributions, and so on.

Other validation checks that could be used in other scenarios are:

- **Invalid character check**, when the input is checked to make sure it is all digits or all text, depending on the requirements of the field
- **Check digit**, when a calculation is performed on a long series or string of digits to produce an extra digit; this is then added to the end of the string; the computer re-calculates when data is entered to check whether it gives the same result.

Other ways of reducing errors when inputting data

The programmer or systems analyst will need to consider ways of reducing errors in addition to designing validation checks.

Another way of reducing errors is to reduce **the volume of data that has to be input**. For example, the use of coding can help in reducing the number of errors made when entering data. Many students get confused over the use of the word 'coding It should not be used in the sense of 'encoding' as in encryption, where 'encoding' is the correct term. It is often used in the sense of programming code. However, the interpretation of the word, in this instance, is totally to do with input and storage of data. Students should understand that what is meant by coding data is reducing the number of letters in a word, such as 'Y' instead of 'yes' and 'M' instead of 'male When data is shortened in this way, less data is entered and fewer mistakes are made. You are less likely to make spelling mistakes by typing in the letter 'F' than if you typed in the whole word 'female it also means that simpler and shorter validation routines can be used.

Another way of reducing errors is to use direct data entry methods such as barcode reading and optical mark reading. These lead to a reduction in the number of errors when compared with manual typing.

Specifying the required hardware and software

The final stage in the design of the new system will be to decide upon the required hardware and software. Now that Daniel has completed the analysis stage of the cycle, he needs to specify exactly what hardware and software is to be purchased. A supplier will be chosen based on cost, reliability and the after-sales support that can be offered. In the analysis stage of the systems design cycle the systems analyst would have made general recommendations regarding the size and type of hardware as well as suggesting some software requirements. Now, he will need to be specific. He will already have made a record of the existing software and hardware being used by the company.

We have already seen in Section 4.1 that the volume of data will determine the choice of output devices. If there is a heavy volume of data then it would mean that an inkjet printer, for example, would not be as suitable as a large volume output device such as a fairly large laser printer. The order that data will be output in will affect the choice of storage devices. If the data is to be stored in an indexed sequential fashion, because the workers' payroll data may need to be accessed directly as well as sequentially, then a hard disk drive will be recommended rather than a magnetic tape drive.

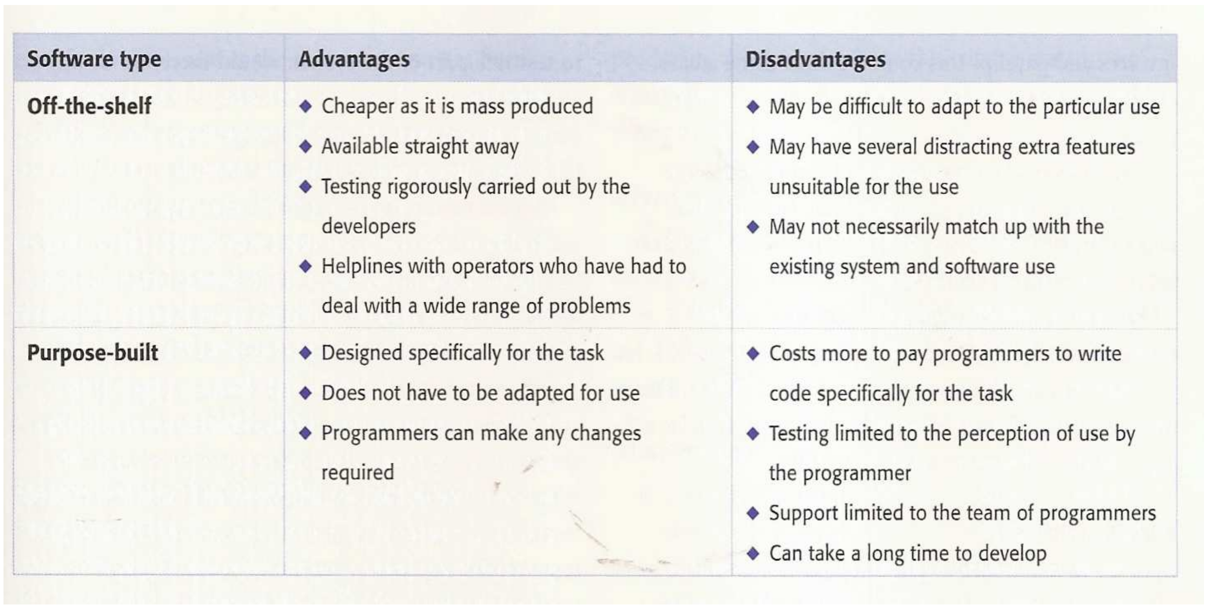
The systems analyst must also choose the software. In our scenario, Daniel may decide that the existing software will be sufficient and that it only needs to be adapted to provide the solution to all the system requirements. Because of the small size of the company, it might be appropriate for Daniel to adopt this strategy. It is likely that other companies dealing with more customers and employing more workers might need to have software written especially for them. However, he will need to be sure that the existing software is capable of producing word-processed documents that can be mail merged and that the various aspects of payroll and invoice production can be completed successfully using the existing database software.

For a larger organisation, it may be necessary for them to employ a programmer to write software specific to their use. There is number of reasons why a company would want to have software specially written for their own purposes: for example, they may need web pages designed which require specialist programming skills; or their database needs may be more complex than that of small companies.

Generally, therefore, there are two choices:

- Off-the-shelf software, which is already written and available
- Purpose-built software, which has to be specially written to solve the problem.

There are many big software houses that produce off-the-shelf software, for example database software, invoicing software, accounting software, payroll software and many more types of, predominant business-oriented software. The table below gives the advantages and disadvantages of each type of software.



Software type	Advantages	Disadvantages
Off-the-shelf	◆ Cheaper as it is mass produced ◆ Available straight away ◆ Testing rigorously carried out by the developers ◆ Helplines with operators who have had to deal with a wide range of problems	◆ May be difficult to adapt to the particular use ◆ May have several distracting extra features unsuitable for the use ◆ May not necessarily match up with the existing system and software use
Purpose-built	◆ Designed specifically for the task ◆ Does not have to be adapted for use ◆ Programmers can make any changes required	◆ Costs more to pay programmers to write code specifically for the task ◆ Testing limited to the perception of use by the programmer ◆ Support limited to the team of programmers ◆ Can take a long time to develop

4.3 Development and testing

Having designed the system required, the next three stages are:

- creating the system
- testing the system
- improving the system.

Each time the system is changed as a result of test results, it will need testing again, so the second and third of these stages maybe repeated several times, until the system is completed.

Creating data structures and program modules

After the data structures have been designed, they will need to be created using the software or programming language recommended by the systems analyst. The programmer will produce the program code needed to solve the problem. The nature of the problem will determine the amount of programming that is required. Some small organizations such as Peter and Irene's may require very little as it may be possible to adapt existing software in order to produce the outputs required. For example, if the main requirement is the production of invoices and payslips this could be done using a basic database package, a spreadsheet package and a word processor

However, some organizations and companies are so large that this may not be practical. Instead, their requirements may be so complex that they need to have software written especially for them.

In our scenario, Daniel will need to create file structures for customers, their orders and the payroll. He will have already designed these and selected the software and so, unless he needs to employ a programmer, he will now use the software to create these files.

Testing strategies

In order to make sure that the system works as it is intended the system has to be tested. It is important that the systems analyst produces a test plan. The test plan will consist of a list of test data together with the results expected to be produced by the system (expected results). The systems analyst will then make a note of the results which the system in fact produced when this data was used (actual results). There will also be a note made by the analyst of any comments if there are differences between the actual results and the expected results.

The two main ways of testing a system are by using test data and live data, both of which will need to be carried out.

Test data

First of all, let us look at the three types of test data:

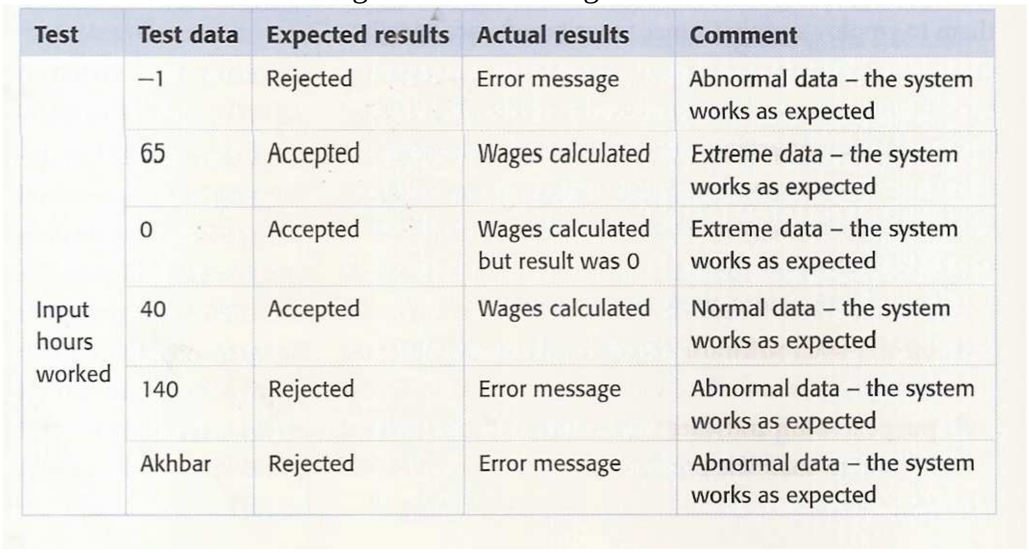
• Normal data
• Extreme data
• Abnormal data.

Normal data is data that is acceptable or valid to the system. This is data which should lot produce error messages from the system. For example, if we look at the payslip example we would not have anybody working more than 65 hours in a week and the lowest number of hours would be 0 in the event that somebody was ill all week. Normal data to test this part of the system would therefore be any number between 0 and 65, including 0 and 65. If data such as this is entered and error messages are produced then there is a problem with the system.

Extreme data is only used where a range of data is input. For names and addresses, for example, there would be no extreme data. Where a range is used, extreme data are the values at either end of the acceptable range. For our payslip example, extreme data would be 0 and 65 only.

Abnormal data is data which is not acceptable or valid. For example, in a numeric field in a database we would not expect alphabetic characters to be entered. Where a range of data is used, numbers outside the range would be considered abnormal. In our payslip example, any negative number or any number greater than 65 or any item of text would be considered abnormal data. Examples of abnormal data here would be -1, 66, 140, 'Akhbar'

Every aspect of the new system will need to be tested with different types of test data.
A test plan used for the above data might look something like the table below.



Test	Test data	Expected results	Actual results	Comment
Input hours worked	–1	Rejected	Error message	Abnormal data – the system works as expected
	65	Accepted	Wages calculated	Extreme data – the system works as expected
	0	Accepted	Wages calculated but result was 0	Extreme data – the system works as expected
	40	Accepted	Wages calculated	Normal data – the system works as expected
	140	Rejected	Error message	Abnormal data – the system works as expected
	Akhbar	Rejected	Error message	Abnormal data – the system works as expected

If the entry of 65 had produced an error message then the validation check would need to be looked at again. The validation check may have been written as <65 instead of < 65. Extreme test data is very important as it helps to identify this sort of mistake.

Live data

This is data that has been used in the existing system. It will be used because the outputs are already known. In our example, Daniel will use the hours worked by the workers in past weeks. Because

there will be records of the payslips produced from these figures, it will be easy for him to see that the correct output has been produced.
He will select a week in the year where there may have been special circumstances such as a public holiday.
He will also choose another week where it was quite an average week. He will then run these sets of data on the new system and compare the results with the payslips already in existence for those two weeks.
If there are differences between the results, using the new system and the existing payslips, amendments will need to be made to the system.

Improvements that could be needed as a result of testing
The next step is to correct any mistakes. Before the system is implemented, the analyst will correct any faults that were identified as a result of testing. If we look at the example for hours worked being 65, it might be necessary to change the validation check as suggested. Checking the data validation, calculations and fife structures should be reasonably straightforward as the test plan will show where there are differences between the expected and actual results of the system.
However, the output from the system may still not have been exactly as expected when live data was used. The next step will be to see where and when the differences occurred. This can be achieved by a process called single stepping. Certain software allows you to run the system one step at a time so that the exact point where the differences occurred can be clearly seen in the programming code. That section of code can then be amended to produce the correct result.

4.4 Implementation

After the system has been developed the systems analyst will want to get the system up and running. His next step will be to choose a method of implementing the new system. There are four methods to choose from: parallel running, direct changeover, phased implementation and pilot running. They all have their own advantages and disadvantages.
Parallel running
Parallel running is running the new system while the old system is still running. This means that the results from the new system can be checked against those of the existing system. When the new system is consistently producing the same results as the existing system, the existing system can then be stopped and replaced by the new system.
Advantages:
• A workers can be trained to use the new system gradually while it is being implemented.
• If there are any problems with the new system and it has to be stopped, there is still the old system as a backup.
Disadvantages:
• Two sets of workers have to be paid to keep both systems working.
• It takes a lot longer to fully implement than any other method.
Direct changeover
In direct changeover, the existing system is replaced by the new one instantly. The existing system is stopped and the new system starts running immediately. This method can only be used when the new system has been thoroughly tested. There are risks associated with this method, as once the old system is closed down it cannot be reintroduced.
Advantages:
• The cost is less than parallel running as only one set of workers needs to be paid.
• It is a very quick method of fully implementing a new system.
Disadvantages:
• If there are problems, there is no backup system.
• It can be difficult to make improvements to the new system and keep it working.
Phased implementation

Phased implementation involves the introduction of the new system one part at a time. It could be that the production of invoices is done by the new system whilst other aspects like the payroll and processing of orders carries on as before. Any problems with the new method can be overcome and when the system is working perfectly another aspect can be moved onto the new system such as processing orders. This approach continues until all aspects have been transferred to the new system.

Advantages:
• If the new system does not work as intended with one aspect, the other aspects of the work can carry on as normal.
• Workers have time to get used to the new system.
Disadvantages:
• It is a slow method of implementation compared with direct changeover.
• If the new system doesn't work properly, it is not possible to fall back on the old system
Pilot running
Pilot running is the method adopted by large organisations. The new system is implemented in one branch of the organisation whilst the other branches continue with their existing system. Workers from other branches can be taught on the new system before it is introduced to their branch.
Advantages:
• If the system does not work properly, not all branches are affected.
• The later branches benefit by learning from the mistakes made in earlier branches.
Disadvantages:
• It is a slow method of implementation.

4.5 Documentation

When a system is ready to be implemented, documentation has to be produced for the new system. This documentation will take one of two forms: technical or user.
Technical documentation is produced specifically for systems analysts and programmers. It is meant to help when the system needs further development or upgrading. It also very helpful should any errors occur in the system and they need to amend the system to get rid of these errors.
User documentation is provided to help users operate the new system. It can take the form of a tutorial that helps users work their way through the system.
Developing elements of technical documentation
Technical documentation consists of systems documentation and program documentation. Together, these will relate to information about the structure of any data files, document templates and spreadsheet workbooks.
The systems documentation provides a detailed overview of the whole system and includes:
• test plans and test results so that systems analysts can see the results of these — this means that when they find an error in the system they will be able to use this data again to check if they have successfully removed the errors
• the results of the systems analysis, including elements like data flow diagrams — this should help anybody who wants to develop the system
• what is expected of the system
• overall design decisions such as the choice of hardware and software as well as file, input and output structures.
Program documentation also needs to be produced for those pieces of program code that have been written. It includes:
• a description and the purpose of the software — this will explain what the software does and its features, as well as the reasons for choosing those pieces of existing software that were used instead of the programmer having to write code
• the input and output data formats that have been used
• the program flowcharts that were i5roduced at the design stage

• the program listing — this will be a complete copy of the code used as well as annotations explaining what each module of code does
• notes that will help any future programmer to make modifications to the system.
Designing and developing elements of user documentation
User documentation has a different function. It is provided to help the user actually use the system. There are a number of reasons why the systems analyst needs to produce this. Firstly, the users of the system will not be at all familiar with the system and so will need help with various parts of the system until they are familiar with it. It will also save the analyst time in the long term, as if the documentation is effective they will not be contacted on a regular basis to show users how to do certain things. The user documentation will include:

• screenshots, as well as descriptions of how to use the software to save a file, perform a search, sort data, print data, add records, delete records and edit records
• the purpose of the system
• the input and output formats
• the hardware and software needed to run the system
• examples of sample runs of the system so that the user can tell if they are using the system in the correct way
• what to do when errors occur
• a troubleshooting guide or a list of Frequently Asked Questions.

4.6 Evaluation

After the system has been developed, tested and implemented, it must be evaluated. There are a number of stages in the evaluation process.
A system is usually evaluated against a set of criteria:
• Is the system reliable and robust?
• Does the system do what it was intended to do?
• Is the system easy to use?
• Is the new system efficient?
• Is the solution appropriate?

A system needs to be evaluated (in terms of the efficiency, easy of use and appropriateness of the solution. evaluation process involves using test results, obtaining feedback from users, identifying limitations of the system and assessing the benefits of proposed improvements.)

Using test results to evaluate the solution

As we have seen above, the test results will help the systems analyst to make judgments. In our example Daniel will have recorded the results of his testing in the form of a table from which an excerpt was shown above. Comparisons will have been made of the actual results with the expected results. If the results are not as expected, Daniel would use the comparisons and comments to make any refinements which may be needed. For example, if a worker's wage should be $300 for a particular week but the output wage was $30, Daniel would need to check the relevant calculation and see if the formula used had the decimal point in the correct place. Other comments in the comparison table would also help in this process.

Obtaining feedback from users

In order to see if the system is working as it should, users must be consulted over the new system. The way users' responses are recorded may differ from one evaluation to another.

First of all, the systems analyst could observe users performing set tasks and record their progress using video recording. For example, Daniel could record Josephine and Mary working on the new system.
Alternatively, he could get a user to perform a task and measure the time it takes them to carry out the task compared to the old method. Daniel could time Irene over the running of the payroll system and compare it with the original method. He would then be in a position to make his conclusions based on the time it took Irene to produce the required output and consequently make a report on the efficiency of the new system.
Another method is for the systems analyst to interview users to gather their responses about what they thought of the system and how easy it was to use. The systems analyst could use their findings to see whether the system needs changing. This might be an appropriate method for Daniel to use with Peter.
Finally, the systems analyst could hand out questionnaires to all the workers to ask them about their thoughts on the new system with regard to how easy they found it to use. These results could be analysed statistically.

Identifying limitations of the system

The systems analyst will have discussed how successful the new system has been in meeting the original objectives as specified in the requirements specification. He will have gained information about how easy the system is to use. He will have seen if the users have accepted it and are happy to work with the new system. He will also have recorded any extensions to the system that users have said they would like. This will have given him a fair idea of any limitations in the system. These could vary from the minor, such as the colours used on the input screen distracting the user, to major limitations such as not being able to produce connected output, for example the past records of the customer together with the current orders of the customer.

Making improvements to the system

In order to make improvements, the systems analyst evaluates the results of testing against the requirement specification. He also needs to evaluate the results of user testing. Users are interviewed for their opinions on the limitations there are with the new system. They will also be asked about any extensions to the system they would like to see. In order to do this, the analyst identifies users who are typical of the workforce and the tasks that they might perform. They are then interviewed as a result of performing these tasks.
Having identified the limitations, the systems analyst must decide with the users whether the good points of the system compensate for these. They will need to decide whether any extensions to the system that the users have identified should be included. It may be that the system needs to be improved in view of the limitations that have been identified. After the improvements have been made, the system will need to be developed, tested and evaluated again.

Solve Examination questions page 76

Done by IT instructor
Ahmad Hirzallah
A.H.S.S.