The Systems Life Cycle

Introduction

Systems analysis is, as the name states, the **analysis of systems**!

The systems that we are talking about are the **systems within organizations and businesses** - systems of communication, financial systems, manufacturing systems, etc. - basically the **systems** that make the organisation or business **work**.

A person who analyses systems is known as a **Systems Analyst**.

Often systems analysts are **employed** by organizations or businesses to help them **improve their systems** and so become more **efficient**, and for businesses, more **profitable**.

Why carry out Systems analysis?

- A System is computerised and may be out of date -- > it needs replacing
- Small company has a computer system but its use if very limited + could be improved to allow many more aspects of the business to become computerised.

In the case of this course **scenarios** are used so that instead of just studying the theory, real life applications can also be seen.

Our Scenario

A Building Supply Company: **Biashara Street Building Supplies** in Nairobi, Kenya.
This company buys the following things from big companies and then sells them to smaller companies.

i. Bricks
ii. Cement
iii. Gravel
iv. Sand
v. Roof Tiles

It is run by two directors; Peter Kimanthi and Irene Kibaki. It employs two secretaries, three sales people and two truck drivers.

This company has ONE computer which is used for creating word-processed letters, Sending emails and Keep record of the company's customers on a data base.

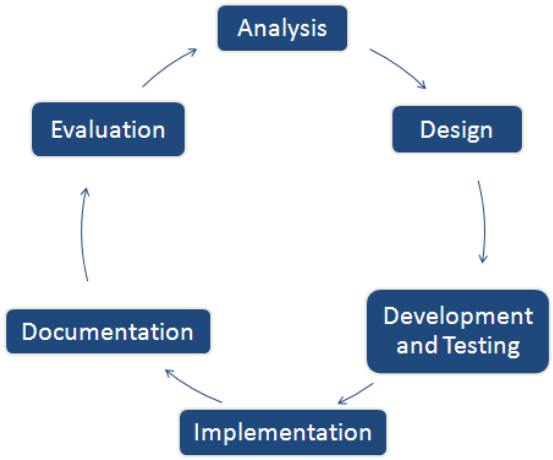
The two secretaries:

- Josephine: Deals with workers' personal information and also Keeps the customer details
- Mary: Keeps information about number of hours worked and also Processes orders from the customers.

Peter and Irene believe that they could increase the use of their computer, so they call in a systems analyst (Daniel) to look at how the computer is used and advise them if their business would improve if they increased the use of their computer.

The Stages of Systems Life cycle

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 analysing the existing system is
  - ✓ To see how data flows around the system.
  - ✓ 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.

The System analysis stage

Systems analysis involves examining the existing system in detail. Particularly,

- ✓ 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.

After these steps, the systems analyst then needs to:

- ✓ identify suitable hardware and software for a new system
- ✓ Identify user and information requirements.

1. Collection of Information

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, pay slips etc. If, after the analysis, it is considered necessary, they will be replaced with ones produced by the computer system.

This method of collection of information helps the analyst to:

1. identify the inputs and outputs of the system as it currently operates
2. Assess the processing that the computer system will need to carry out.
3. Calculate the number of documents that are processed
4. The Volume of data on all the documents
   (This would help the analyst decide on the size of memory.)
5. The types of input and output devices needed to cope with this volume of data.

Questionnaires

With large groups of people, a questionnaire is a quick and simple way to gather information.

When using a questionnaire, keep in mind that the way the question is asked can influence the way people answer the question.

For example: If you want to know what details the bosses (Peter and Irene in our scenario) need to see on pay slip, the question

"What details of the workers and their pay would you like to see on your workers' payslips? " will not produce precise information as the question phrased this way:

"Which of the following information would you like to see on a workers payslip:

- *worker's name*
- *Worker's number*
- *Rate of pay*
- *Hours worked*
- *Income tax rate*
- *Income tax paid*
- *Any other information"*

Advantages of using questionnaires

- An effective questionnaire once created allows you to give out many copies.
- Questions can be answered quickly.
- Answers are more honest as the questions can be answered anonymously.
- Fairly cheap method of gathering data.
- Answers to the questionnaires can be entered into a computer automatically using an OMR (Optical Mark Reader).

Disadvantages

- People often do not complete or return the questionnaire.
- Unclear questions cannot be explained as people are left to answers the questions alone.
- You may get incorrect data if people have misunderstood a question.
- It is hard to ask very technical or specific questions on a questionnaire.
- The questionnaires may take a lot of time to produce

Interviews

The systems analyst can interview **key people (**like managers and representatives of the workers) within the system to find out how it works because it takes a lot of time to interview all workers.

Interviews allow lots of very detailed information to be gathered.

Advantages	Disadvantages
They are flexible; the interviewer can move away from the 'script'.	It takes a lot of time to organise an interview
Questions can be changed to suit the one being interviewed (different questions for a director compared to a regular employee).	Interviewees may not give very accurate answers; they may try to provide the interviewer with answers that they want to hear.
Questions can be explained if they are misunderstood	It takes a lot of time to complete interviews.
More complex questions can be asked which will give more detailed findings.	

Observation

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.

Observation will enable the systems analyst to see the process as a whole.

Advantages	Disadvantages
The analyst can see the process as a whole, *i.e.* it gives him/her the full picture of how the system operates.	"Hawthorne Effect"- some people know that they are being watched so they may change the way they work.
A data flow diagram can be produced which help determine the inputs, outputs and processing.	

Choice of method:

The method of collection of information depends on the following:

1- The type of information that is being collected.
2- The practicality of using the method in the situation/scenario.

For Example: If there were lots of employees it would take very long to personally interview each one, so another method (like questionnaires) can be used.

In our scenario of **Biashara Street Building Supplies,**

- ✓ Peter & Irene can be interviewed as they are the owners of the company, so specific needs for the new system can be established.
- ✓ The Two secretaries are observed as they both do different aspects of work.
- ✓ Sales people are busy all day so it would be easier to give them a questionnaire. They would give them the opportunity to fill out the questionnaire. This is much better than having a rushed interview.
- ✓ Van drivers will not be in the office most of the time so it would to also give them questionnaires.

2. Establishing 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.

The analyst will use the information that they obtained from examining the documents used in the current system to produce documentation of the system.
This stage is often done whilst producing a data flow diagram.

Each section of the system is examined to see what specific

- Inputs
- Outputs
- Processing is required

For example: in a payroll system, 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.

3. Recording Information about the Current System

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 other 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:

- ➢ Data flow diagrams (Most popular)
- ➢ Flow charts

Data Flow diagrams (DFD)

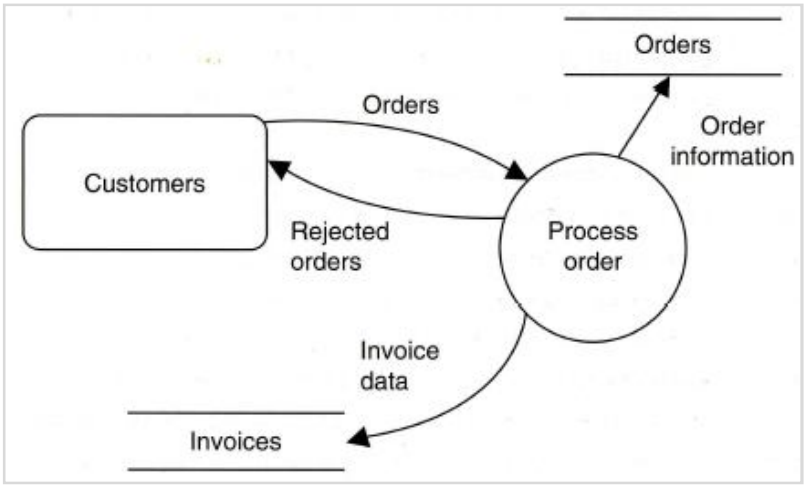
Data flow diagrams are a graphical method of recording the inputs, outputs and processing that have been identified.

A DFD consists of four components:

- Terminators (entities)
- Processes
- Flow arrows
- Stores

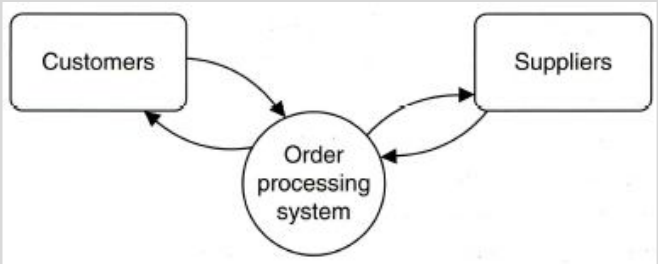
Example from our scenario

- The customer sends in an order to the company.
- 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.
- If does not, it is rejected and sent back to the customer.
- 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.



Different levels of data flow diagrams.

1. Level 0 (Context level)DFD
   This is basically a diagram showing a very generalized diagram with the terminators linked to the current system as a whole.



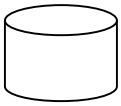
2. Level 1 DFD
   This has many more processes, with more detail about all aspects of the system.
   It would have the process boxes for receiving orders and also for producing invoices.

3. Level 2 DFD
   This would show more parts, each showing parts of the level 1 diagram in much more detail.

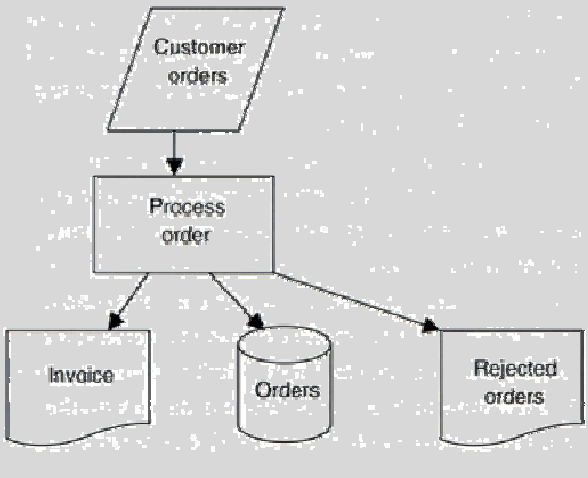
System Flow charts

This method can be used to design system solutions. For that reason, it is not frequently found in the analysis stage.
A system flowchart is designed to represent a new computerised system. The dataflow diagram represents the old manual system

The shapes used in a system flow chart are:

- Input 
- Process 
- Storage (magnetic disk) 
- Output document 

In our scenario, we would have a system flowchart looking like this:



4. Identifying Problems with the Current System

The DFDs will help the analyst to identify problems with the current system.

- Weaknesses can be identified.
- By showing the process in great detail it will be easier to identify if there are problems like job duplication or time wasting.

For example: (From the scenario)

It may be unnecessary to have both secretaries involved in entering data for the payroll process. So in the new system these two files of information may be linked together making the process easier.

5. Identifying suitable hardware and software for a new system

The actual hardware/software is not chosen at this stage.
However the analyst can then make generalised recommendations for the software/hardware having decided upon the required outputs, storage and processing requirements of the system using the data flow diagrams.

For example (in our scenario)

- Daniel knows the volume of data being input from the terminators and flow arrows. This can then allow him to determine an appropriate method of inputting data into the system.
- He can give details on the size and speed of the processor required.
- The stores in the dataflow diagram will allow Daniel to make predictions on how much data needs to be stored allowing him to recommend the size and number of storage devices.

6. 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.

In our scenario
Daniel has already interviewed Irene and Peter and will use this information to discover their requirements for the new system.
Using his own observations and the data flow diagram he can make generalised user requirements. This will then allow Daniel to produce a **requirements specification**. This is a list of features of the system that are required. These will be things that the user wants to be able to do with the system. For instance:

- "I want the system to find me the details of an individual customer quickly."
- "I want the system to produce the payroll overnight so I can just set it to print at the end of one day and its there for me the next morning."

The Design Stage

At this stage the analyst may involve a programmer.
A programmer is the person who will actually write the software, if new software has to be written.

The following need to be designed:

- Inputs to the system
- Outputs from the system
- Files and/or databases needed to store the data
- Processing required to produce the outputs
- Validation checks that will be needed
- Data needed to test the system

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

1. Designing data collection forms and screen layouts

Data collection forms can be either hard copy or screen based.

In our scenario, the drivers never need to use the computers so it will be better if the data about them is collected on a hard copy.

Points to note 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. (This can be achieved by putting boxes in each section to be completed. The form will be easier for the secretary to read, so fewer errors will be made when typing in the data.)
- The requirements for completing the form should be clear to workers, so that they know which sections to complete.
- The design of these forms depends on the user requirements as well as output required from system and file structures.

Factors to note when designing an input screen

- It must be easy to use
- It must be attractive to look at
- It needs to limit the possibility of inaccurate data being entered
- It must contain guidelines for workers on how to fill in any data entry forms
- It must also allow them to navigate from one screen to another without difficulty

In our scenario, Mary and Josephine must not get distracted by an over-elaborate design. So using different fonts and colours should be avoided.

2. Designing report layouts and screen displays

When reports and screens are to be designed, 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.

- Many of the features of input designs have to be included in output designs, such as being attractive to look at and not being over-elaborate.
- Screen displays should have instructions on how to move from screen to screen. Screen displays must have instructions on how to move from screen to screen.

In our scenario: Daniel has to decide on what types of outputs are required. This will be done by looking at existing documents and by examining the results of the interviews that he carried out.

The two main aspects of the way output is designed are:

1. Format of the output, i.e.
   i. Graph
   ii. Lists of records
   iii. Reports
   iv. Tables
2. Medium it is to be produced on, which can be
   i. Paper
   ii. Screen display
   iii. Sound

In our scenario: Daniel l using the information that he acquired from interviewing Peter and Irene and from the existing documents he can conclude that invoices and payslips could be produced using the existing database software in order to produce reports.

For the layout of the documents, the analyst needs to consider the following points:

- Consider who will see the documents
- Consult with the owners of the company.

In our scenario: Peter and Irene will want their customers to gain a good impression of the company.

- They would want to have their company advertised favourably and it must contain all the required information in an orderly and easy-to-read format.
- It must also be ensured that the content of the invoice matches that of the customer and company needs.

<u>The screen output</u>

- It must be simple as possible (E.g., the internal users of the system don't need to see adverts within the system itself.)
- The screen output must not contain any extra material other than that required.
- Each output screen must have a consistent theme so the user doesn't get confused.
- Instructions on how the user is to navigate between screens must be included on the screen.

3. 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 down the parts of the systems flowchart into algorithms or program flowcharts.

The systems analyst will have to design a structure that will have **two** files.

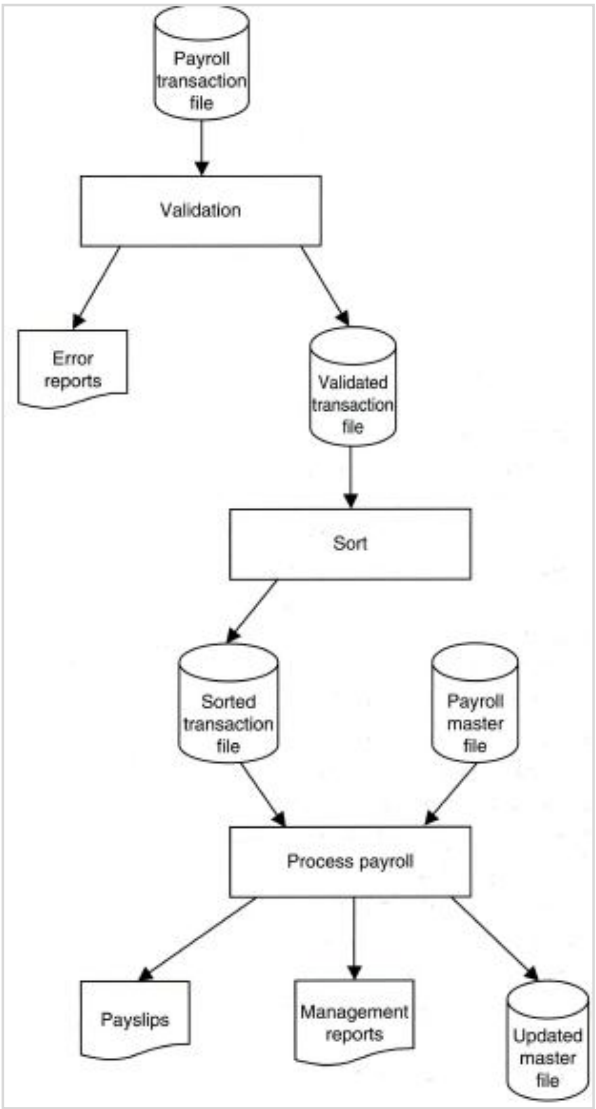
- **Master file**: containing data that does not change often, such as name, works number, department, hourly rate, and so on.
- **Transaction file**: containing the data that changes each week, such as hours worked.

These two files will be processed together to produce the payroll.

Before the transaction file can be used,

- It is checked for errors
- It is then sorted in the same order as the master file
- The transaction file is stored on a magnetic disk even though the data is in sequential order.

The systems analyst will now decide on individual file structures and whether any programming is required.



This is a system flowchart showing how the transaction file and the master file can be used in this process.

Attributes of the Master file

The systems analyst 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 create a test plan.

4. Designing Validation Routines

It must be ensured that the number of errors made during data entry is very small. One method used is Validation.

Validation is the process of determining the accuracy of data. The process is used to ensure that data conforms to given specifications.

In order to be sure that the data entered is valid, a number of validation routines have to be put in place. The number and types of validation routines depend on the form of input and the file structures that are being used in the system.

Keep in mind that all fields cannot have a validation check. For instance, there are so many variations of a person's name, so it's hard to put validation checks on such a field.

Types of validation checks.

- Range check: Used on data containing numeric data (e.g. workers can work for a minimum of 0 hours and a maximum of 9 hours)
- Length check: this can be used for text fields (e.g.
- Format/Picture check: check if a specific field has a specific format. For instance it must contain 4 numbers followed by a letter.
- Invalid character check: input is checked in order to ensure that it is digits or all text
- Check digit: this is when a calculation is performed on a long string of digits which produces an extra digit which is added to the end of the string of digit.

Other ways of reducing errors:

- Reduce the volume of data. For instance, using coding; using 'y' instead of 'yes'. Less data is entered so fewer errors are made. This also means that there will be simpler and shorter validation routines.
- Use direct data entry methods. For example barcode reading or optical mark reading.

5. Specifying the required hardware and software

The required hardware and software is decided upon at this step. A supplier is chosen basing on:

- cost,
- reliability and
- After-sales support that can be offered

At this stage the analyst makes general recommendations about the size and type of hardware and the type of software.

- The volume of data will determine the choice of the output devices. E.g. If it is a heavy volume of data then an inkjet printer would not be suitable.
- The order in which the data will be output will affect the choice of storage devices. E.g. if the data needs to be accessed in an indexed sequential manner, then a magnetic disk can be chosen for storage.

The analyst will also need to choose the type of software. The analyst will have to choose from the following:

- **Off-the-shelf software**, which is created for use by a large range of customers - it tends to be quite general-purpose.
- **Purpose-built software** (Custom-written software), which is designed and written specifically for one customer.

Advantages and disadvantages

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

Examples of off-the-shelf-software:

- Database software
- Invoicing software
- Accounting software
- Payroll software

Note: (For examinations, you MUST use generic names like these, not brand names. You lose marks if you do)

The Development and Testing stage

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 may be repeated several times, until the system is completed.

1. Creating the system (data structures and program modules)

Once the data structure has been designed, it needs to be created using the software that has been recommended by the systems analyst.

The programmer produces the program code that is needed to solve the problem.

The nature of the problem determines how much programming is required. For example, a Small Company, whose main requirement is production of invoices and payslips may need a basic database package, spreadsheet package and word processor. A big company may require whole brand new software to be written for them.

In our scenario,

- Daniel needs to create a file structure for the customers, their orders and the payroll.
- It is most probable that by this stage Daniel will have already designed the structures and has already selected the software for this.
- Now he has to use the software to create the files.

2. Testing Strategies

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.

There are two main ways of doing this:

1. Using test Data
2. Using live Data

Types of data used for testing

1. **Normal data** is data that is acceptable. It should not produce an error when input into the system.
   Example:
   In a payslip no-one should have worked more than 65 hours or less than 0 hours. So data that lies within that is entered and if an error message is produced it means that there is a problem with the system.
2. **Extreme data** is used only when a range of data is input.
   Things like name and address will not have any extreme data.
   But in the case of a payslip where the most amount of hours worked could be 65 and the lowest could be 0; entering 65 or 0 would be considered as extreme data.

3. **Abnormal data** is data this not acceptable.
   For example:
   You wouldn't want alphabetic characters in a numeric field.
   - When data range is used; any number outside that range will be considered as abnormal.
   - For instance with the payslip; any number that lies outside the range of 0 and 65 hours will be considered as abnormal data.

Test plan

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

Live Data

This is data that has been used in the existing system.
It is useful to use it because the outputs of the system are already known.

In our scenario,
- Daniel can use the hours worked by the workers in the past week and then using this he can check whether or not he has the correct output from the system.
- He will then select a week where there was a special circumstance such as a holiday and then check the output from that week.
- Then he can choose a week that was somewhat an average week.

These sets of data will be run on the new system and their output will be compared to the ones that already exist. If there are any discrepancies between the data; amendments need to be made.

3. Improvements that could be needed as a result of testing

Before implementation; the system needs to be cleared of any errors that were identified in the testing stage.

- This may include altering the validation check. Checking the data validation, calculations, and file structures is reasonably straightforward as the test plan will show where there are differences between the expected and the actual results.
- In case the output from the system still doesn't have exactly what is expected when the live data is used, an approach called **single stepping** is used. This is basically when the software allows you to run the system one step at a time so the exact point where the differences arise can be identified.

The Implementation Stage

After the system has been developed the systems analyst will have to get the system up and running. The implementation of the new system occurs when the old system is replaced by the new one.

Methods of implementation

1. **Parallel running:**
   This is when the new system is run along side the old system, meaning that the results on the new system can easily be checked against those of the old system.

Advantages	Disadvantages
Workers can be trained to use the new system gradually while it is being implemented	2 sets of workers need to be paid to keep both the systems running.
If the new system fails the old system is always available as the back up.	It takes a longer time to implement than any other method

2. **Direct change over**
   The existing system is replaced by the new one instantly. This method should only be used if the new system has been tested thoroughly. There is a major risk taken here because if the old system is shut down it cannot be reintroduced.

Advantages	Disadvantages
Cost is less than parallel running because only one set of workers needs to be paid.	If a problem arises there is no back up system
It is a quick method of implementing the system.	It may be difficult to make improvements to the new system and keep it working

3. **Phased implementation**

This is when the new system is introduced to the system one part at a time. This way any problems that arise can be over come and as soon as the system is working fine another aspect can be introduced. This carries on until all aspects have been introduced into the system.

Advantages	Disadvantages
If the introduced part of the system fails all the other parts are still working	Slow method of implementation compared to direct changeover
Workers have time to get used to the new system	If the new system doesn't work properly there is no way of going back to the old system

4. **Pilot running**

This method is usually taken up by large organisations. The new system is implemented in one of the branches of the organisation while other branches continue with the existing system.

Advantages	Disadvantages
If the system doesn't work; not all the branches are affected	Slow method of implementation
Later branches benefit by learning from the mistakes made in earlier branches	

The Documentation stage

When a system is ready to be implemented, documentation has to be produced for the new system.

Types of documentation:

1. Technical Documentation:
   - It is produced for systems analysts and programmers. It helps when the system either needs to be further upgraded or developed.
   - It also very helpful should any errors occur in the system and they need to amend the system to get rid of these errors.

1. User Documentation:
   - It helps the users to operate the new system. It could be a tutorial that helps the user 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.

System documentation provides a detailed overview of the whole system. It includes things like:

1. Test plans and test results
2. Results of the systems analysis; including things like dataflow diagrams
3. What is expected of the system
4. The overall design decisions; things like choice of hardware and software, file input and output structures.

Program documentation has to be produced for the pieces of programming code which have been written. It includes:

- Description and purpose of the software explaining what the software does, its features, and reasons as to why certain pieces of software were chosen
- The input and output data formats that have been used
- The program flowcharts that were produced at the design stage
- The program listing; copy of the code used with annotations explaining what the code does
- Notes that may help any future programmer to make modifications to the system

Designing and developing elements of user Documentation

User documentation is provided to help the user actually use the system.

The system analyst produces user documentation for different reasons

- Since the user of the system will not be familiar with the system, they will need help with various parts of the system until they are familiar with it.
- It will save the analyst time in the long term. They will not have to be contacted on regular basis to show users how to do certain things.

The user documentation includes:

1. Screenshots; and descriptions of how to use the software
2. The purpose of the system
3. Input and output formulas
4. Hardware and software required to run the system
5. Examples of the sample runs of the system
6. What to do when an error occurs
7. Troubleshooting guide or list of FAQ's

The Evaluation stage

After the system has been developed, tested and implemented, it must be evaluated.

Stages of evaluation

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?

The evaluation process

A system needs to be evaluated in terms of the efficiency, ease of use and appropriateness of the solution.

The evaluation process involves

- using test results to evaluate the solution,
- obtaining feedback from users,
- identifying limitations of the system
- Assessing the benefits of proposed improvements.

Using test results to evaluate the solution

The test results will help the systems analyst to make judgements.

In our scenario:

- Daniel will have recorded the results from his tests in a table.
- Comparisons can be made between the actual results and the expected results. If the results are not as expected changes will have to be made.

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.

For example,

- The systems analyst could observe users performing set tasks and record their progress using video recording.
- 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.
- He could interview users to gather their responses about what they thought of the system and how easy it was to use.
- The systems analyst could also use their findings to see whether the system needs changing.
- He could also 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.

Identifying limitations of the system

The analyst will have determined:

- Whether the new system has met its original objectives
- How easy to use the new system is
- If the users have accepted it and are happy to work with the new system
- Whether there are new features or requirements that the users want

This will have given him a fair idea of any limitations in the system.

- Some limitations could be minor; for example, distracting colours on the user input screen
- Some limitations can be major; for example, unable to produce connected output, like old customer records with new customer orders.

Making improvements to the new system

In order to make improvements, the systems analyst will

- Evaluate the results of testing against the requirements specification
- Evaluate the results of user testing.
- Interview users to get their opinions on the limitations there are with the new system. They will also be asked about any extensions or improvements 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.
- Once the limitations have been decided upon whether or not the good points of the system compensate for the limitations. It will also be decided whether or not the improvements should be included, and which improvements can be included.

Once the improvements have been made, the system needs to be developed, tested and evaluated again.

Revision Past paper questions

You need to do as many revision papers as possible.

Years between 2008-2013