Cambridge

OL- IGCSE

ICT

CODE: (0417)

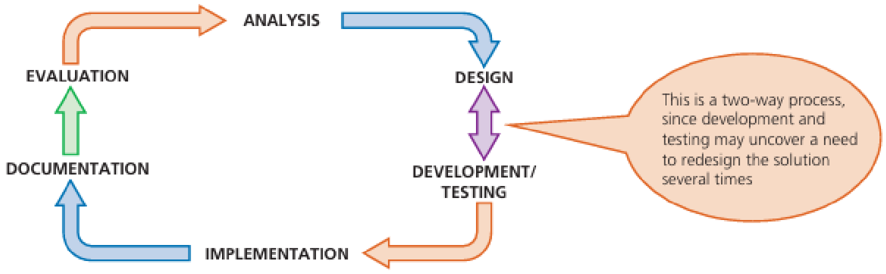
Chapter 07

The systems life cycle

The changes will involve an ICT solution, which means a **systems analyst** needs to be brought in to oversee the whole upgrade process.

Once a new system is agreed and it has been fully tested, it is then installed. It then needs to be fully evaluated and any changes made where necessary. Therefore, a cycle of events take place until a fully-working system is signed off and handed over to the management team. This whole process is called the systems life cycle.

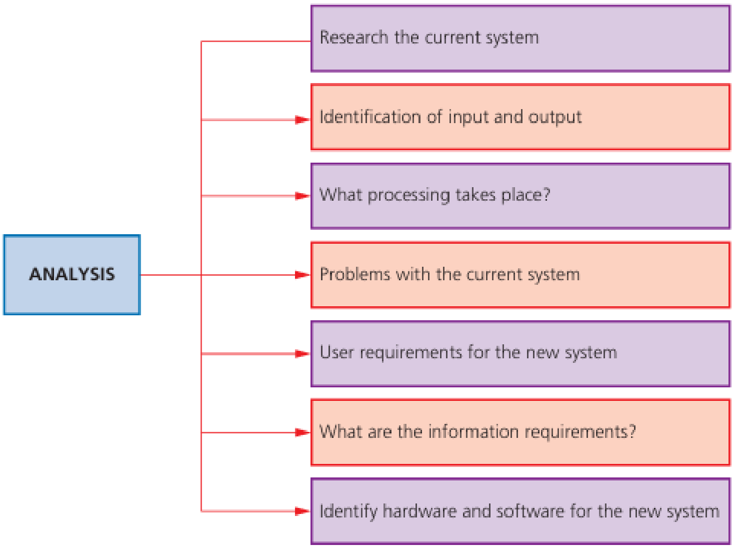
There are many stages in the systems life cycle, which have been summarised in Figure 7.1.


ANALYSIS
DESIGN
DEVELOPMENT/ TESTING
IMPLEMENTATION
DOCUMENTATION
EVALUATION
This is a two-way process, since development and testing may uncover a need to redesign the solution several times

▲ **Figure 7.1** Systems life cycle

7.1 Analysis

The first stage in the process is the analysis of the current system. Figure 7.2 shows the stages in analysis.


ANALYSIS
Research the current system
Identification of input and output
What processing takes place?
Problems with the current system
User requirements for the new system
What are the information requirements?
Identify hardware and software for the new system

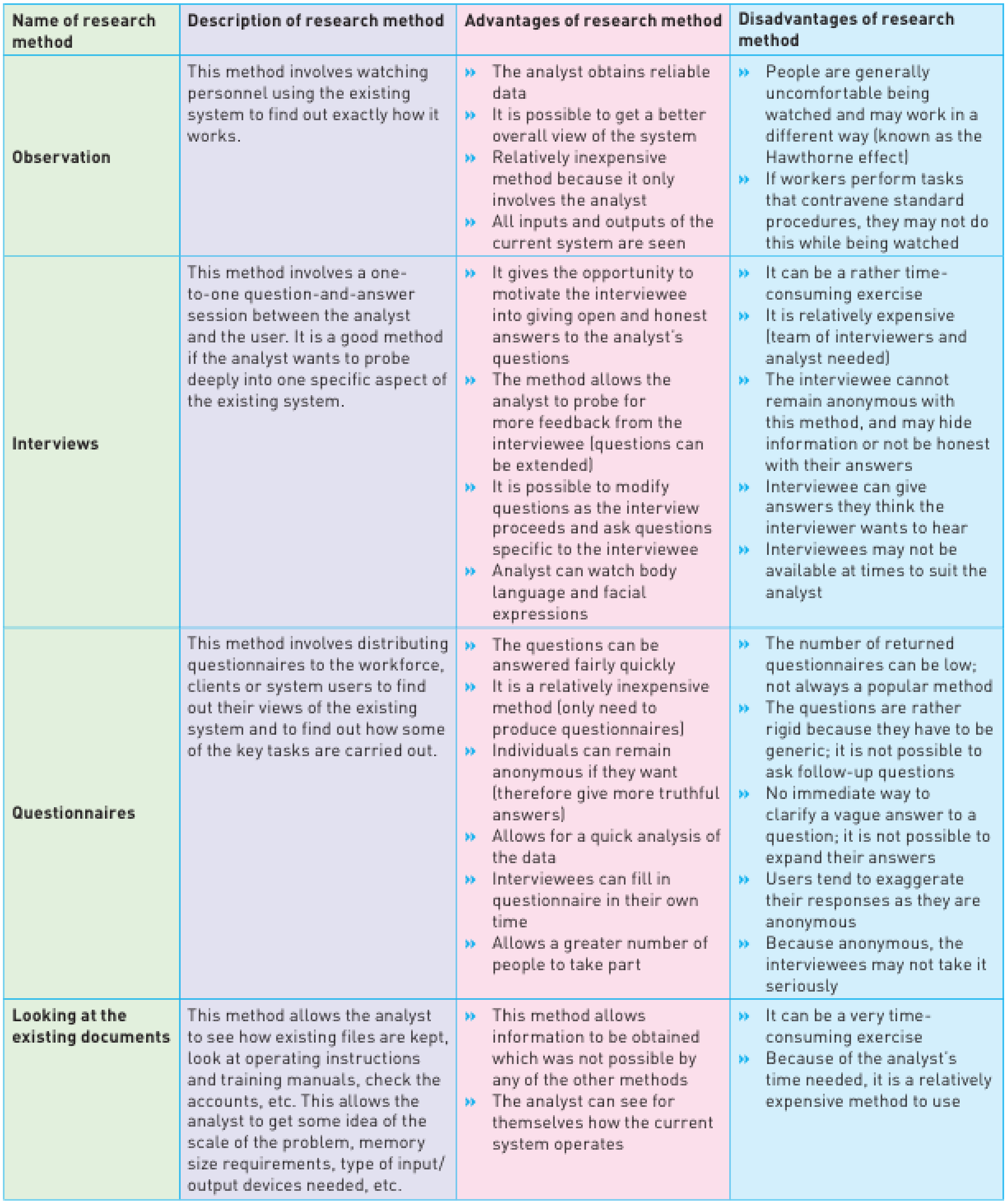
▲ **Figure 7.2** Analysis stage

7.1.1 Analyse the current system

There are four methods used to research the current system. The four methods used are:

» Observation
» Questionnaires
» Interviews
» Examination of existing documents.

▼ **Table 7.1** Methods of researching the current system



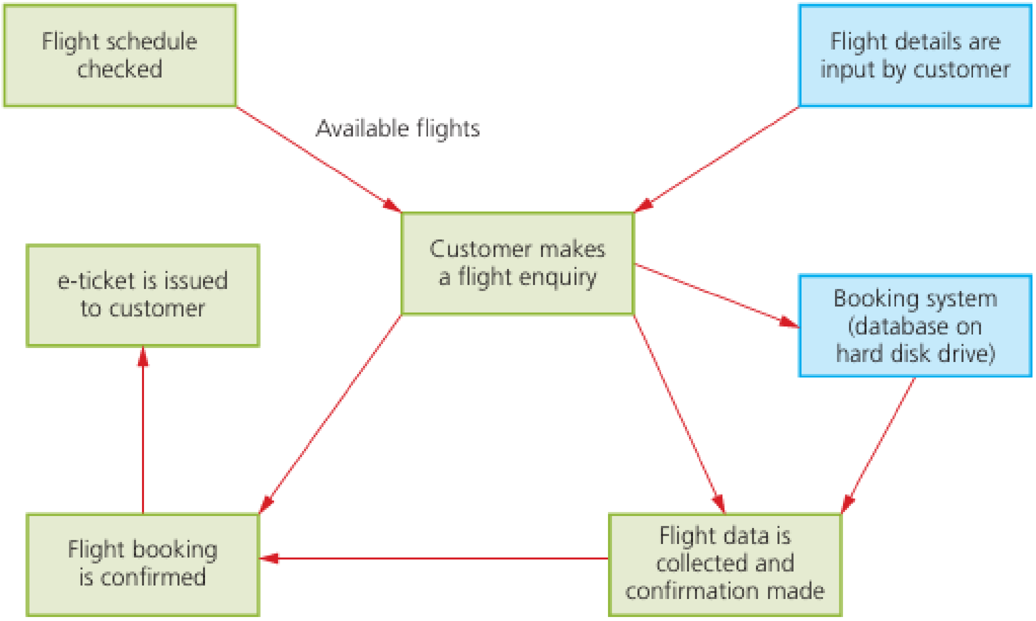
Name of research method	Description of research method	Advantages of research method	Disadvantages of research method
Observation	This method involves watching personnel using the existing system to find out exactly how it works.	» The analyst obtains reliable data » It is possible to get a better overall view of the system » Relatively inexpensive method because it only involves the analyst » All inputs and outputs of the current system are seen	» People are generally uncomfortable being watched and may work in a different way (known as the Hawthorne effect) » If workers perform tasks that contravene standard procedures, they may not do this while being watched
Interviews	This method involves a one-to-one question-and-answer session between the analyst and the user. It is a good method if the analyst wants to probe deeply into one specific aspect of the existing system.	» It gives the opportunity to motivate the interviewee into giving open and honest answers to the analyst's questions » The method allows the analyst to probe for more feedback from the interviewee (questions can be extended) » It is possible to modify questions as the interview proceeds and ask questions specific to the interviewee » Analyst can watch body language and facial expressions	» It can be a rather time-consuming exercise » It is relatively expensive (team of interviewers and analyst needed) » The interviewee cannot remain anonymous with this method, and may hide information or not be honest with their answers » Interviewee can give answers they think the interviewer wants to hear » Interviewees may not be available at times to suit the analyst
Questionnaires	This method involves distributing questionnaires to the workforce, clients or system users to find out their views of the existing system and to find out how some of the key tasks are carried out.	» The questions can be answered fairly quickly » It is a relatively inexpensive method (only need to produce questionnaires) » Individuals can remain anonymous if they want (therefore give more truthful answers) » Allows for a quick analysis of the data » Interviewees can fill in questionnaire in their own time » Allows a greater number of people to take part	» The number of returned questionnaires can be low; not always a popular method » The questions are rather rigid because they have to be generic; it is not possible to ask follow-up questions » No immediate way to clarify a vague answer to a question; it is not possible to expand their answers » Users tend to exaggerate their responses as they are anonymous » Because anonymous, the interviewees may not take it seriously
Looking at the existing documents	This method allows the analyst to see how existing files are kept, look at operating instructions and training manuals, check the accounts, etc. This allows the analyst to get some idea of the scale of the problem, memory size requirements, type of input/output devices needed, etc.	» This method allows information to be obtained which was not possible by any of the other methods » The analyst can see for themselves how the current system operates	» It can be a very time-consuming exercise » Because of the analyst's time needed, it is a relatively expensive method to use

7.1.2 Record and analyse information about the current system

Inputs, outputs, processing and current problems

The next stage in the process requires the analyst to find out:
» What input and output takes place
» What processing is done
» What problems exist with the current system
» User and information requirements for the new system.

One method the analyst can use is a **data flow diagram (DFD).**


Flight schedule checked
Available flights
Flight details are input by customer
Customer makes a flight enquiry
e-ticket is issued to customer
Booking system (database on hard disk drive)
Flight booking is confirmed
Flight data is collected and confirmation made

▲ **Figure 7.3** DFD for flight booking process

User requirements

The problem to solve is this: computer system developers do not really understand how a business works: business managers do not really know how computer systems could help them. The user requirements are designed to help with this problem:
» User requirements are therefore written by the analyst for the business managers (who are the customers).
» They are written in natural language with very few technical details or jargon.
» Their purpose is to allow the customers to check that what the analyst proposes
» The user requirements will also describe what the analyst thinks the customer does with their system.

Information requirements

» This is the information needed to support the business.
» The information requirements are made up of:
» What? (that is, the data)
» When? (that is, the timing). A systems analyst turns the information and user requirements into a functional requirements specification (that is, how the new system will be developed and implemented, including timescales).

7.1.3 System specification

The DFD and other information gathering processes allow the analysis team to identify what hardware and software is needed to run the new system.

Identify and justify hardware

» Identification and justification of which input devices are needed might be, for example:
– **barcode readers** (using barcode readers avoids the need to manually input data about goods, which is more efficient, less error-prone and less expensive in the long run

– **scanners** (these could be used if it is necessary to convert any existing paper documents into an electronic format during the implementation stage)

– **touch screens** (this may be the best and most cost-effective way of gathering information from a business customer,

» Identification and justification of which output devices are needed; for example:
– **3D printer** (if the company are manufacturing toys,
– **very large 60” (152 cm) monitors**
– **speakers**

Identify and justify software

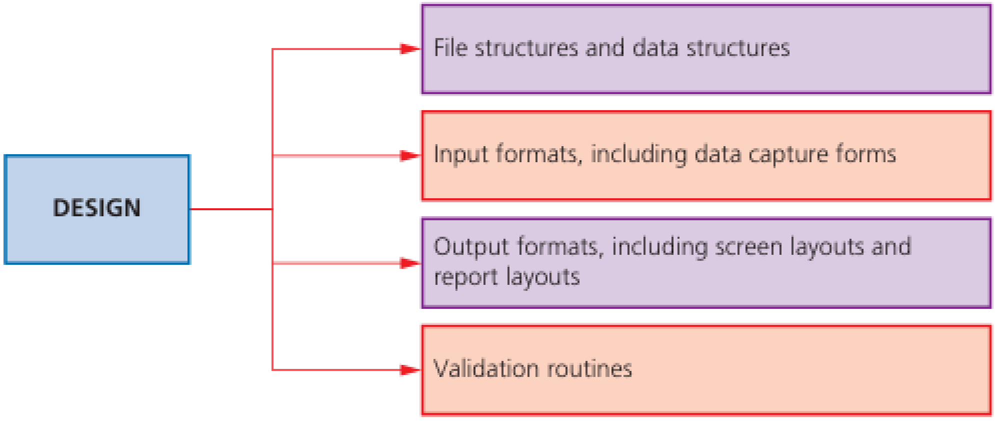
» Identification and justification of which software is required; for example:
– **operating system** (which operating system is the most appropriate to meet the company needs)
– **applications software**: – off-the-shelf software, which would save a lot of development time and costs, but may require compromises in how the company runs; off-the shelf software (such as Word or Excel) also has a huge user-base in case of problems and a minimum of training will be required, because the software is well known
– bespoke software (written specifically for the company)
– this will require considerable time and money to develop, but will exactly meet the company’s requirements; it will also require considerable training in using software unknown to the employees, and there will be no user base to seek help (they will have to rely on the software development technical team, which could be expensive).

» Storage requirements also need to be considered; for example:
– **size of storage** (how many bytes of storage are required for the systems to run now, and in the future)
– **type of storage** (which storage type is the most suitable for the company: hard disk drives, solid-state drives or even magnetic tape drives)
– the choice could depend on:
– data access and data write speeds– number of read-write operations (there is still some doubt about the longevity of SSD if it has to endure large numbers of read-write operations)
– type of access – can it be serial access to the data (all read in order) or does it need to be direct (no need to read all the data in order)
– if huge amounts of storage are required and data access time is not that important, magnetic tape may still be the best option.

7.2 Design

7.2.1 Design

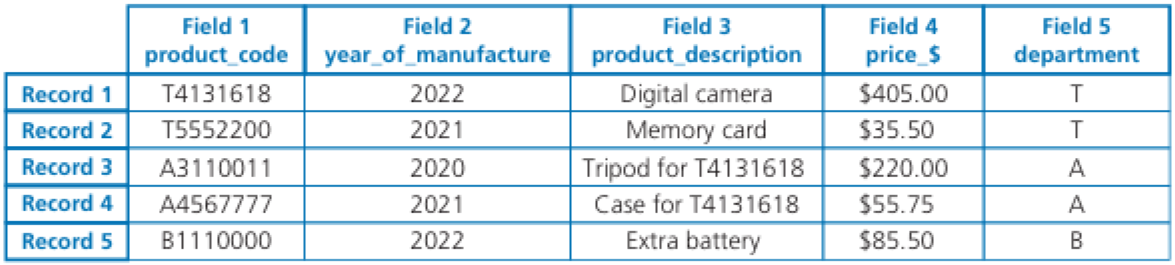
Once the analysis of the existing system has taken place, and the systems analyst has a better idea of the scale of the problem, then the next stage in the process is design. Figure 7.4 summarises the design stage.


DESIGN
File structures and data structures
Input formats, including data capture forms
Output formats, including screen layouts and report layouts
Validation routines

▲ **Figure 7.4** Design stage

file structures and data structures

A file is made up of a number of **records**, and each record is broken up into **fields**. One of the fields must be unique and will act as the **primary key field** – this is to allow each record to be uniquely identified. An example of a record (in the file) is shown in Figure 7.5.



	Field 1 product_code	Field 2 year_of_manufacture	Field 3 product_description	Field 4 price_$	Field 5 department
Record 1	T4131618	2022	Digital camera	$405.00	T
Record 2	T5552200	2021	Memory card	$35.50	T
Record 3	A3110011	2020	Tripod for T4131618	$220.00	A
Record 4	A4567777	2021	Case for T4131618	$55.75	A
Record 5	B1110000	2022	Extra battery	$85.50	B

▲ **Figure 7.5** Typical file structure (showing records and fields)

Each field in the record now needs to be totally defined as follows:

» Field name (for example, **product code**)

» Field length (what is the maximum number of characters that need to be stored)

» Data type (see below)

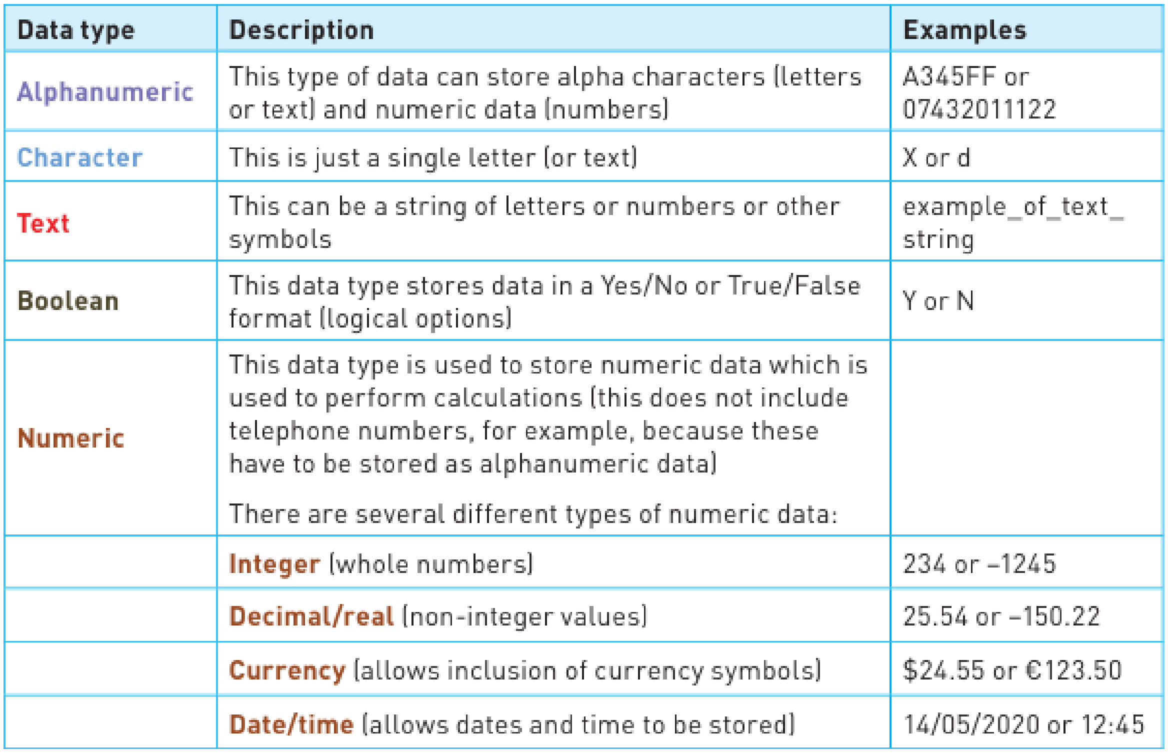
» Is any code being used (in this example, the codes T, A and B are being used – coding saves space in the file because only a single character is being used; this also speeds up entry and also reduces errors)

» The primary key field here will be **product code** because it is unique.

Data types

We will now consider data types. The data types in Table 7.2 are the ones you are most likely to come across:

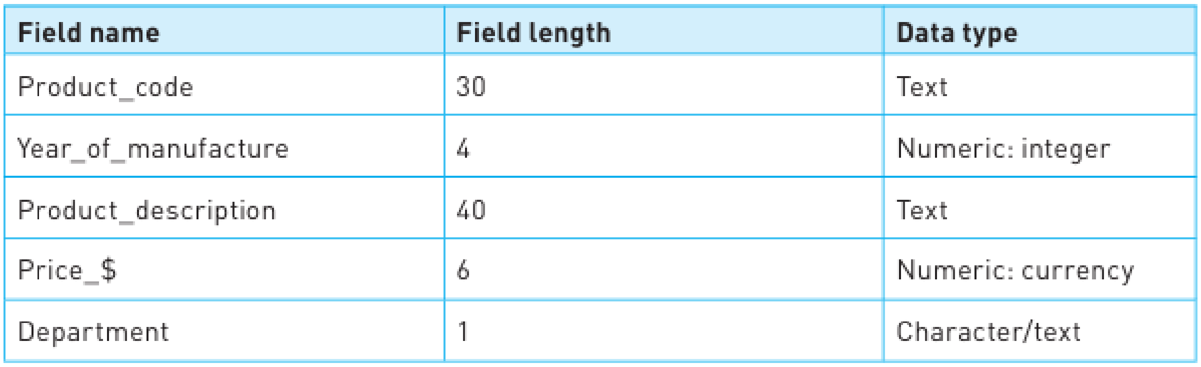
▼ **Table 7.2** Data types



Data type	Description	Examples
Alphanumeric	This type of data can store alpha characters (letters or text) and numeric data (numbers)	A345FF or 07432011122
Character	This is just a single letter (or text)	X or d
Text	This can be a string of letters or numbers or other symbols	example_of_text_string
Boolean	This data type stores data in a Yes/No or True/False format (logical options)	Y or N
Numeric	This data type is used to store numeric data which is used to perform calculations (this does not include telephone numbers, for example, because these have to be stored as alphanumeric data) There are several different types of numeric data:	
	Integer (whole numbers)	234 or −1245
	Decimal/real (non-integer values)	25.54 or −150.22
	Currency (allows inclusion of currency symbols)	$24.55 or €123.50
	Date/time (allows dates and time to be stored)	14/05/2020 or 12:45

It is now possible to complete the file structure for the example in Figure 7.5.

▼ **Table 7.3** Field lengths and data types



Field name	Field length	Data type
Product_code	30	Text
Year_of_manufacture	4	Numeric: integer
Product_description	40	Text
Price_$	6	Numeric: currency
Department	1	Character/text

7.2.2 Validation routines

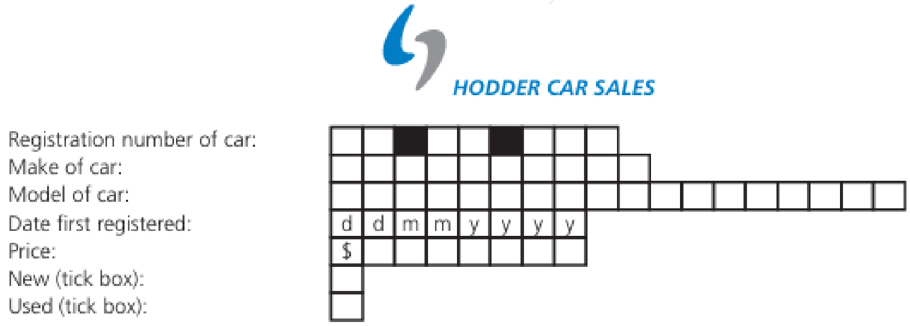
When data is input into a computer system, there is a need to check that the data is acceptable. **Validation** is the process where data entered into a computer is checked to see if it satisfies certain criteria. It is an automatic check carried out by the computer as part of its programming.

▼ **Table 7.4** Validation checks (routines)

Validation check	Description	Examples
Range	This checks to see if the data input lies between an acceptable upper value and an acceptable lower value	Limiting a temperature range from 10 to 50 degrees Celsius
Type/character	This checks to see if the data entered is of the correct type (i.e. letter or number only)	A person's name should not contain numbers A person's height should not contain letters
Length	This checks to see if the data input contains only the required number of characters	If a password contains eight characters, then an input with seven characters or nine characters, for example, should produce an error message
Format	This checks to see if the data input is in the correct format	Ensures the date is entered in a format such as dd/mm/yyyy (e.g. 10/12/2023)
Presence	This checks that data has been entered into a field and it has not been left empty	For example, when using an online form, a person's telephone number may be a 'required field'; if no data is entered, this should give rise to an error message
Check digit	This is an extra digit added to a number which has been calculated from the other digits	Check digits can identify three types of error: 1 if two digits have been transposed during input; for example, 13597 instead of 13579 2 an incorrect digit has been entered; for example, 13559 instead of 13579 3 a digit has been missed out or extra digit added; for example, 1359 or 135799 instead of 13579 (in all three cases, the check digit (usually the last digit) would not be 9 if an error had been made)

7.2.3 Input formats (data capture forms)

Data capture forms are often used to input data into a computer. These forms ensure data is input into the computer in the correct format.



HODDER CAR SALES

Registration number of car:

Make of car:

Model of car:

Date first registered: d d m m y y y y

Price: $

New (tick box): □

Used (tick box): □

▲ **Figure 7.7** Paper-based data capture form

A computer-based data capture form is slightly different. These often have the following features:

» Use of text boxes to capture key data clearly

» Use of on-screen help when completing the form

» Use of drop-down/combo boxes where there are limited choices

» Use of radio buttons and tick boxes requiring a single click of a mouse to select

» Automatic validation of data as it is entered

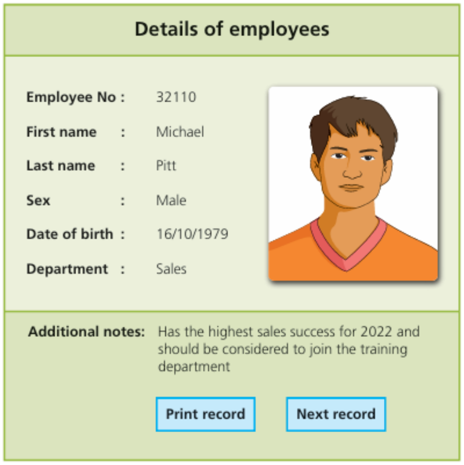
» Control buttons (such as next form, clear entry, save, etc.)

» Double entry boxes (with verification rules) to check correctness of key data (for example, when keying in an email address).

7.2.4 Output formats – screen layouts and report layouts

Screen outputs should be designed:
» To make sure the size of all the output fields is correct
» So that any instructions/descriptions are clear
» So that the full screen is utilised (avoiding large areas of 'nothing')
» So that colours and fonts (size and type) make the output clear.

Reports (often the output from a database search) should clearly show all the fields that were included in the search criteria. The output is usually in the form of a table – the example in Figure 7.9 outputs a list of all sales managers over 40.

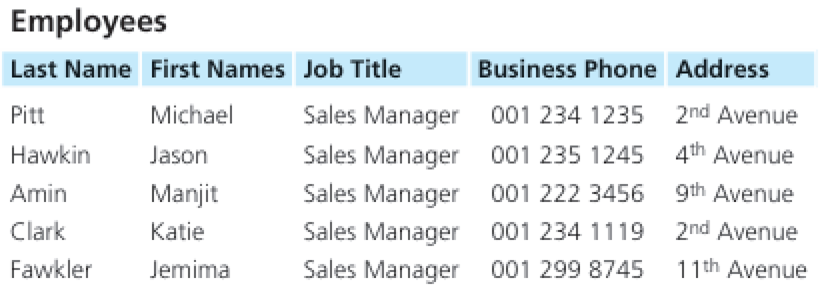


Details of employees

Employee No : 32110
First name : Michael
Last name : Pitt
Sex : Male
Date of birth : 16/10/1979
Department : Sales

Additional notes: Has the highest sales success for 2022 and should be considered to join the training department

▲ **Figure 7.8** Screen output example



Employees

Last Name	First Names	Job Title	Business Phone	Address
Pitt	Michael	Sales Manager	001 234 1235	2nd Avenue
Hawkin	Jason	Sales Manager	001 235 1245	4th Avenue
Amin	Manjit	Sales Manager	001 222 3456	9th Avenue
Clark	Katie	Sales Manager	001 234 1119	2nd Avenue
Fawkler	Jemima	Sales Manager	001 299 8745	11th Avenue

▲ **Figure 7.9** Report example

7.3 Development and testing

7.3.1 Testing The need for testing

» If the system contains files (for example, a database) then the file structure would need to be finalised at this stage (for example, what type of data is being stored in each field, length of each field, which field will be the key field, how will the data files be linked, etc.).

» Because it is important that the correct data is stored in files (etc.) there are certain techniques that need to be adopted to make sure the data populating the files/database is at least of the right type and that it conforms to certain rules. Validation routines and verification methods are used to ensure this happens.

» Obviously, any system being developed will have some form of user interface. The types of hardware have already been considered; how these are used to actually interface with the final system now needs to be identified.

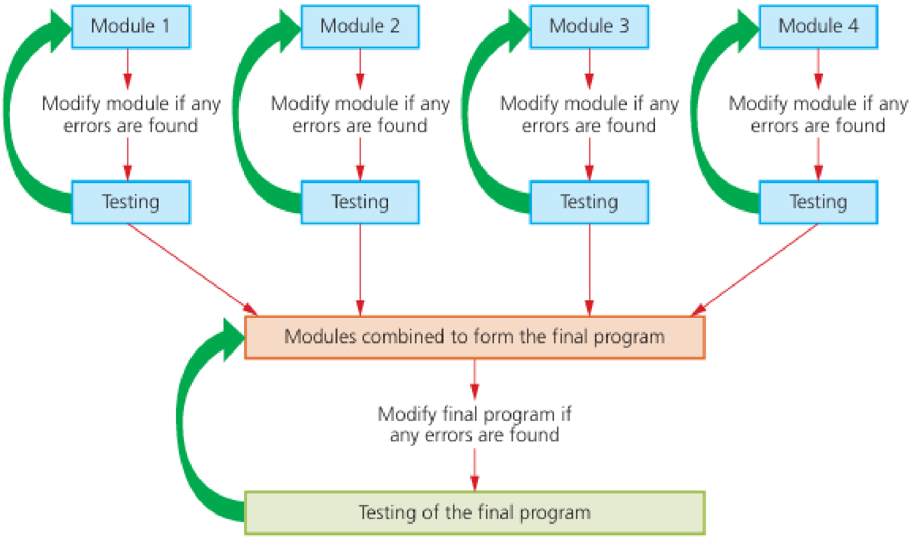
Test designs Test designs cover how a system is to be tested. Table 7.5 shows the test designs that need to be considered, and how we can ensure the following aspects can be achieved:
» Testing the data structures
» Testing the file structures
» Testing the input methods
» Testing the output formats
» Testing the validation rules.

▼ **Table 7.5** Testing designs

Data structures	The test design should determine how we can test that all data is in a correct format or has been stored in the correct way (for example, whether tables hold data correctly)
File structures	The testing design should test that the file structures function correctly (i.e. data is stored in the correct format and can be correctly retrieved when required)
Input formats	The testing design should determine how we can test that all data can be entered into the system correctly (for example, if a date is to be entered does the input format permit this date to be entered correctly)
Output formats	The test design should determine how we can test that screen outputs and reports are all in the correct format (for example, are the output results clear, complete and correctly match the input data)
Validation routines	The test design should determine what data is needed to test to see if all the validation rules work (for example, does the system correctly reject unreasonable data being input)

Test strategies

» Software is often developed in modular form. This method allows it to be broken down into smaller parts (known as modules). Each module is developed separately by a programmer (or team of programmers).
» Each module needs to be tested separately to see if it functions correctly. Any problems resulting from the testing require the module to be modified and then tested again.
» Once the development of each module is completed, the whole system needs to be tested as a whole (with all modules functioning together).
» All of this may lead to a need to improve the input and output methods, file/ database structures, validation and verification methods, etc. and then fully test everything again.


Module 1
Module 2
Module 3
Module 4
Modify module if any errors are found
Modify module if any errors are found
Modify module if any errors are found
Modify module if any errors are found
Testing
Testing
Testing
Testing
Modules combined to form the final program
Modify final program if any errors are found
Testing of the final program

▲ **Figure 7.10** Module testing

Test plan, test data and live data

Once the testing designs and strategy have been determined, it then becomes necessary to formulate a test plan for each module.

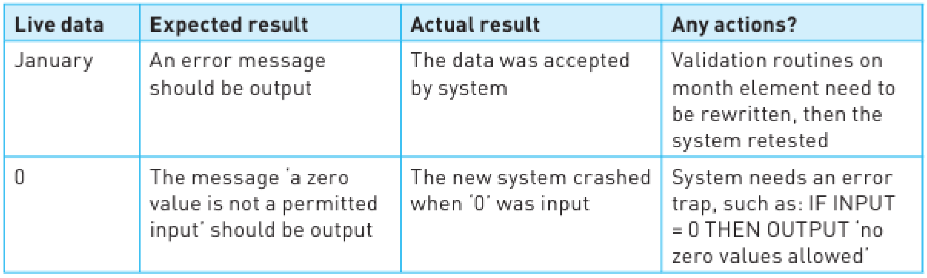
The example we will use is inputting a date into a database field. The entered data must take the format dd/mm/yyyy and all data must be numeric.

» Normal – this is data which is acceptable/valid and has an expected (known) outcome; for example, the month can be any whole number in the range 1 to 12

» Extreme – this is data at the limits of acceptability/validity; for example, the month can be either of the two end values i.e. 1 or 12

» Abnormal – this is data outside the limits of acceptability/validity and should be rejected or cause an error message;

▼ **Table 7.6** Live data comparison table

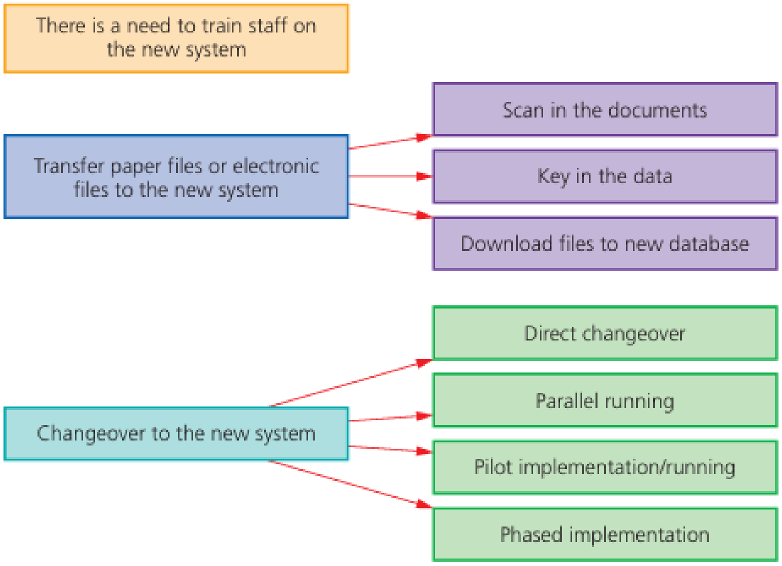


Live data	Expected result	Actual result	Any actions?
January	An error message should be output	The data was accepted by system	Validation routines on month element need to be rewritten, then the system retested
0	The message 'a zero value is not a permitted input' should be output	The new system crashed when '0' was input	System needs an error trap, such as: IF INPUT = 0 THEN OUTPUT 'no zero values allowed'

7.4 Implementation

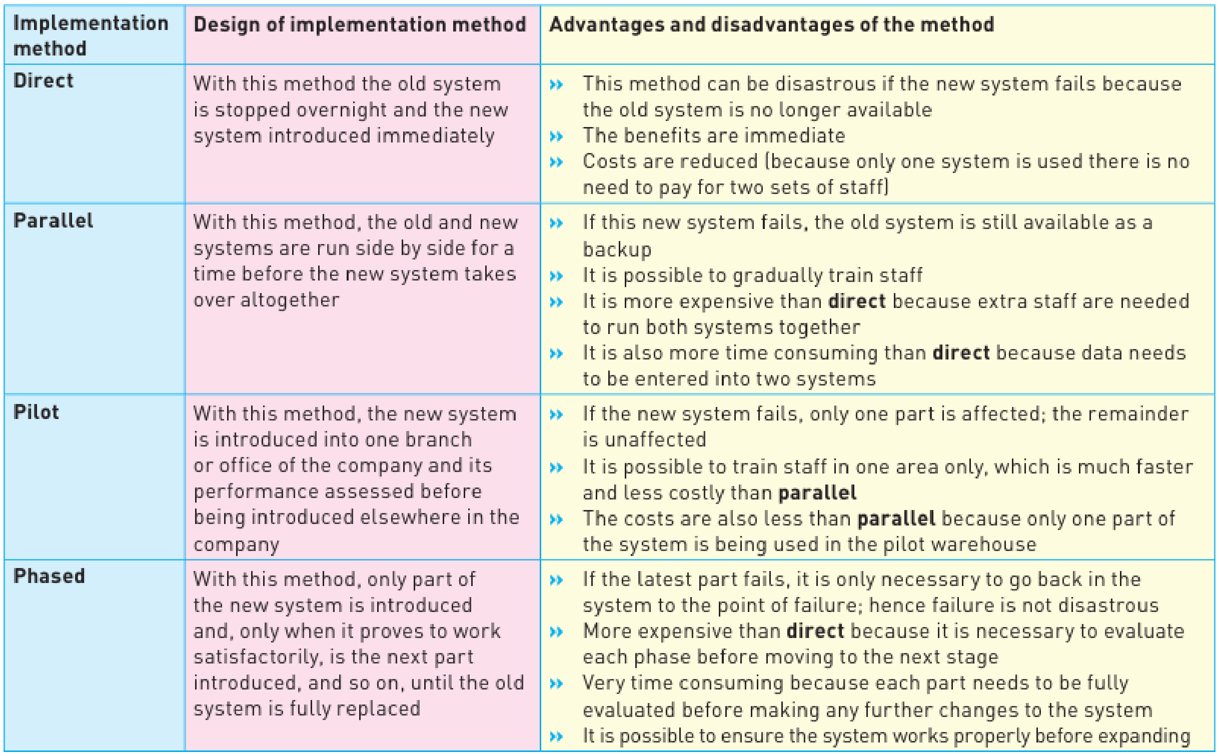
7.4.1 System implementation

Once the system is fully tested, the next stage is to fully implement it. Some of the stages in this process are shown in Figure 7.12.


There is a need to train staff on the new system
Transfer paper files or electronic files to the new system
Scan in the documents
Key in the data
Download files to new database
Changeover to the new system
Direct changeover
Parallel running
Pilot implementation/running
Phased implementation

▲ **Figure 7.12** The implementation stage

▼ **Table 7.7** Methods used in changeover (part of implementation)



Implementation method	Design of implementation method	Advantages and disadvantages of the method
Direct	With this method the old system is stopped overnight and the new system introduced immediately	» This method can be disastrous if the new system fails because the old system is no longer available » The benefits are immediate » Costs are reduced (because only one system is used there is no need to pay for two sets of staff)
Parallel	With this method, the old and new systems are run side by side for a time before the new system takes over altogether	» If this new system fails, the old system is still available as a backup » It is possible to gradually train staff » It is more expensive than **direct** because extra staff are needed to run both systems together » It is also more time consuming than **direct** because data needs to be entered into two systems
Pilot	With this method, the new system is introduced into one branch or office of the company and its performance assessed before being introduced elsewhere in the company	» If the new system fails, only one part is affected; the remainder is unaffected » It is possible to train staff in one area only, which is much faster and less costly than **parallel** » The costs are also less than **parallel** because only one part of the system is being used in the pilot warehouse
Phased	With this method, only part of the new system is introduced and, only when it proves to work satisfactorily, is the next part introduced, and so on, until the old system is fully replaced	» If the latest part fails, it is only necessary to go back in the system to the point of failure; hence failure is not disastrous » More expensive than **direct** because it is necessary to evaluate each phase before moving to the next stage » Very time consuming because each part needs to be fully evaluated before making any further changes to the system » It is possible to ensure the system works properly before expanding

We will now consider **changeover** to the new system in more depth. As indicated in Figure 7.12, there are four common methods used for changing over from the old system to the new system.

7.5 Documentation

7.5.1 Documentation

Once the new system is fully developed, a considerable amount of documentation also needs to be produced for:

1. people who may need to modify or develop the system further at some later stage
2. the end-user.

Technical documentation

Technical documentation is designed to help programmers/analysts to make improvements to the system or repair/maintain the system. This can consist of any of the following:

» Program listing/coding
» Programming language used
» Program flowcharts/algorithms
» System flowcharts
» Purpose of the system/program/software
» Limitations of the system
» Input formats
» Hardware requirements
» Software requirements
» Minimum memory requirements
» known ‘bugs’ in the system

» List of variables used (and their meaning/description)
» File structures
» Sample runs (with results and actual test data used)
» Output formats
» Validation rules
» Meaning of error messages.

User documentation

User documentation is designed to help users to learn how to use the software or system.

7.6 Evaluation

7.6.1 Evaluate a solution

Once a system is up and running it is necessary to do some **evaluation** and carry out any maintenance if necessary.

Some results from the evaluation may lead to two things happening:
» Update of hardware because:
– of feedback from end-users
– new hardware comes on the market, necessitating change
– changes within the company require new devices to be added or updated.

» Update of software because:
– of feedback from end-users
– changes to the company structure or how the company works that may require modifications to the software
– changes in legislation that may require modifications to the software.

Revision questions

1. March/2023/Paper_0417/12/No.3
State three items that should be included in the technical documentation of a computer system.

2. March/2023/Paper_0417/12/No.17(a)
Verification and validation are used to check data.
(a) Identify and describe two methods of verification.

3. March/2023/Paper_0417/12/No.15
Tawara Stores is a small company that does not make much profit. The managing director is deciding whether to implement a new IT system using direct changeover or parallel running.
Describe the advantages and disadvantages of using direct changeover compared with parallel running in this scenario.

4. June/2023/Paper_0417/11/No.7
A systems analyst has been asked to carry out an analysis of a company's current computer system.
(a) State three methods of analysing the current system.

(b) The systems analyst is working with a team to create some new software. The software documentation will be stored on the cloud.
Describe how the team can use the cloud to share the documentation.

(c) Members of the team can use video-conferencing to join business meetings from different locations around the world.
Describe, giving reasons, three disadvantages of using video-conferencing for business meetings rather than meeting in person.

5. Nov/2023/Paper_0417/11/No.10
A teacher has set up a database for a class library. The data about the books in the class library will be entered by the teacher.
(a) The data can be verified by double data entry or visual verification.
Compare double data entry with visual verification. Your answer must include similarities and differences.
(b) When data is entered into a date field it is validated. An example of the data that could be entered is 01/04/2023
State two validation checks that could be used on this data.

6. Nov/2023/Paper_0417/12/No.9
A hospital administrator enters data from medical forms into a database. Validation and verification are both used when entering data to minimise errors.
Explain what is meant by validation and verification and why they are both needed in this scenario.

7. Nov/2023/Paper_0417/12/No.10(a)
The manager of the Tawara Hotel has decided to computerise the hotel's room booking system. The new booking system will allow customers to book rooms online using an app on their smartphones.

The manager has employed a systems analyst who will carry out research into the current system by examining existing documentation and completed questionnaires from staff.

(a) One item of information that the systems analyst will need to identify is problems with the current system.
State three other items of information that the systems analyst will need to identify.

8. Nov/2023/Paper_0417/13/No.1
Peter needs to create a validation check for product numbers. Three examples of product numbers he needs to check are:

977135120127916
976143148163921
845132120651166

Identify two validation checks he could use to check the numbers.

9. Nov/2023/Paper_0417/13/No.10
A school has employed a systems analyst to help update its computerised examination system.
(a) The systems analyst researches the current system to see what will be required in the new system.
State the methods of researching the current system that are described:

(i) talking to the network manager and examinations officer about the current system
(ii) watching the users operating the current system
(iii) forms sent out to all users to complete and return.

(b) After completing the analysis, the new system needs to be designed. One item that needs to be designed is a file structure.
Identify three other items that would need to be designed.

10. March/2022/Paper_12/No.7(b_c)
(a) Identify the most appropriate method of implementation of the new online booking system in this scenario.

(b) Explain, giving reasons, why your answer to part (a) is the most appropriate method of implementation for this scenario.

11. June/2022/Paper_11/No.11
A system can be analysed using different methods.
Discuss the advantages and disadvantages of analysing a system by interviewing staff rather than using questionnaires.

12. June/2022/Paper_12/No.12
Different methods can be used to analyse a system.
Discuss the benefits and drawbacks of analysing a system by observing staff rather than looking at the current systems documentation.

13. June/2022/Paper_12/No.13
solvedpapers.co.uk
A librarian in a college department library is entering data into a student borrowers' file. As the data is entered it needs to be verified.
(a) Describe the two types of verification.

14. June/2022/Paper_12/No.13
A librarian in a college department library is entering data into a student borrowers' file. As the data is entered it needs to be verified.
(a) Describe the two types of verification.
(b) Explain, using examples, why it is necessary to carry out validation as well as verification for this system.

15. June/2022/Paper_13/No.5(a)
The manager of a cinema has purchased a new booking system. There are 26 rows in the cinema with 20 seats on a row. Each seat is identified by a unique ID. The seat ID consists of the row letter followed by the position of the seat on the row. An example of the seat on row F position 12 would be F12.
He is setting up a test plan to test the system.

(a) Write down one example of each of the following types of test data. The answers must be different in each case.
Normal Abnormal

16. Nov/2022/Paper_11/No.4(a)
A teacher is setting up a database which she will use to record her student's exam marks.
(a) The teacher is planning to test the mark range using different types of test data. The range of marks that the teacher can award is 0 to 100.
(i) Write down one example of extreme test data that the teacher can use to check the range.

(ii) Extreme test data is one type of test data. The teacher uses two other pieces of test data.
Write down the type of test data for each of the examples.

17. Nov/2022/Paper_12/No.15
Proofreading is often confused with visual verification.
Describe the differences between proofreading and visual verification.

18. Nov/2022/Paper_13/No.5(b)
The restaurant owner is planning to upgrade the software used to book reservations. He has employed a systems analyst who is analysing the current system. The systems analyst can either send out questionnaires for staff to answer or observe staff working with the current system.
(a) Discuss the advantages and disadvantages of using questionnaires that staff answer rather than observing staff to gather information about the current system.