

Table of contents

TOPIC 1- SYSTEM FUNDAMENTALS	1
SYSTEM LIFE CYCLE	1
1.1 SYSTEMS IN ORGANIZATIONS	3
Planning and system installation	3
1.1.1 The context for which a new system is planned.....	3
1.1.2 The need for change management	4
1.1.3 Compatibility issues resulting from situations including legacy systems or business mergers	5
1.1.4 Different systems implementation	6
1.1.5 Alternative installation processes	7
1.1.6 Problems that may arise as a part of data migration	10
1.1.7 Various types of testing	11
User focus.....	13
1.1.8 Importance of user documentation	13
1.1.9 Different methods of providing user documentation	14
1.1.10 Different methods of delivering user training	16
System backup.....	18
1.1.11 Causes of data loss.	18
1.1.12 Consequences of data loss in a specified situation.....	19
1.1.13 Range of methods that can be used to prevent data loss.....	19
Software deployment.....	21
1.1.14 Strategies for managing releases and updates	21
1.2 SYSTEM DESIGN BASICS	23
Components of a computer system	23
1.2.1 Hardware, software, peripheral, network, human resources	23
1.2.1 The roles that a computer can take in a networked world	24
1.2.3 Social and ethical issues associated with a networked world	25
System design and analysis	29
1.2.4 Relevant stakeholders when planning a new system	29
1.2.5 Methods of obtaining requirements from stakeholders.	30
1.2.6 Appropriate techniques for gathering the information needed to arrive at a workable solution ..	32
1.2.7 Suitable representations to illustrate system requirements	34
1.2.8 Purpose of prototypes to demonstrate the proposed system to the client	39
1.2.9 Importance of iteration during the design process.....	40
1.2.10 Possible consequences of failing to involve the end-user in the design process.	41
1.2.11 Social and ethical issues associated with the introduction of new IT systems	41
Human interaction with the system	43
1.2.12 Usability.....	43
1.2.13 Usability problems with commonly used digital devices.....	44
1.2.14 Methods that can be used to improve the accessibility of systems.....	46
1.2.15 Range of usability problems that can occur in a system	49
1.2.16 Moral, ethical, social, economic and environmental implications of the interaction between humans and machines.....	53
CHAPTER REFERENCES	58
TOPIC 2 COMPUTER ORGANIZATION	61
COMPUTER ARCHITECTURE	61
2.1.1 The central processing unit (CPU) and its elements.....	61
2.1.2 RAM and ROM	65

2.1.3 The cache memory	66
2.1.4 The machine instruction cycle	68
SECONDARY MEMORY	69
2.1.5 Persistent storage and secondary memory	69
OPERATING AND APPLICATION SYSTEMS	74
2.1.6 Functions of an operating system	74
2.1.7 Software application.....	77
2.1.8 Common features of applications	83
COMPUTER ORGANIZATION – BINARY REPRESENTATION	86
2.1.9 Bit, byte, binary, decimal and hexadecimal.....	86
2.1.10 Data representation	96
SIMPLE LOGIC GATES.....	102
2.1.11 Boolean operators	102
2.1.12 Truth tables using Boolean Operators.....	109
2.1 (Additional Section) Simplification and Boolean expressions.....	116
2.1.13 Logic and Venn diagrams using logic gates.....	119
CHAPTER REFERENCES	123
TOPIC 3. NETWORKS.....	124
INTRODUCTION AND SOME DEFINITIONS	124
NETWORK FUNDAMENTALS.....	126
3.1.1 Different types of networks.....	126
3.1.2 Importance of standards in the construction of networks.....	140
3.1.3 Networks, communication and layers	141
3.1.4 Technologies required to provide a VPN	145
3.1.5 Use of a VPN	148
DATA TRANSMISSION.....	150
3.1.6 Protocol and data packet.....	150
3.1.7 Necessity of protocols	150
3.1.8 Speed of data transmission across a network.....	151
3.1.9 Compression of data.....	152
3.1.10 Characteristics of different transmission media.....	153
3.1.11 Packet switching.....	154
3.1 (Additional Section) Network topologies.....	155
WIRELESS NETWORKING	156
3.1.12 Advantages and disadvantages of wireless networks	156
3.1.13 Hardware and software components of a wireless network.....	158
3.1.14 Characteristics of wireless networks	161
3.1.15 Different methods of network security	165
3.1.16 Advantages and disadvantages of each method of network security	168
CHAPTER REFERENCES	171
TOPIC 4. COMPUTATIONAL THINKING	174
TOOLS USED.....	174
THE CONCEPT OF A PROBLEM	174
ALGORITHM.....	177
FLOWCHARTS.....	179
PSEUDOCODE.....	180
AN ALGORITHM AND ITS EXPRESSIONS	180
THE PROGRAMMING CONCEPT.....	182

4.1 GENERAL PRINCIPLES	183
Thinking procedurally	183
4.1.1 Procedure appropriate to solving a problem	183
4.1.2 Order of activities and required outcome	183
4.1.3 The role of sub-procedures in solving a problem	184
Thinking logically	185
4.1.4 Decision-making in a specified situation	185
4.1.5 Decisions required for the solution to a specified problem.....	185
4.1.6 Iteration associated with a given decision in a specified problem	186
4.1.7 Decisions and conditional statements	188
4.1.8 Logical rules for real-world.....	190
Thinking ahead	190
4.1.9 Inputs and outputs required in a solution	190
4.1.10 Pre-planning in a suggested problem and solution	191
4.1.11 Need for pre-conditions	192
4.1.12 Pre- and post-conditions	193
4.1.13 Exceptions that need to be considered.....	193
Thinking concurrently.....	195
4.1.14 Parts of a solution that could be implemented concurrently	195
4.1.15 Concurrent processing and problem solution.....	195
4.1.16 Decision to use concurrent processing in solving a problem.....	196
Thinking abstractly	196
4.1.17 Examples of abstraction	196
4.1.18 Abstraction and computational solutions for a specified situation	197
4.1.19 Abstraction from a specified situation	199
4.1.20 Real-world and abstraction.....	200
4.2 CONNECTING COMPUTATIONAL THINKING AND PROGRAM DESIGN	201
4.2.1 Searching, sorting and other algorithms on arrays	201
4.2.2 Standard operations of collections	218
4.2.3 Algorithm to solve a specific problem.....	219
4.2.4 Analyse an algorithm presented as a flow chart	220
4.2.5 Analyse an algorithm presented as pseudocode.....	223
4.2.6 Construct pseudocode to represent an algorithm	226
4.2.7 Suggest suitable algorithms to solve a specific problem.....	231
4.2.8 Deduce the efficiency of an algorithm in the context of its use	232
4.2.9 Determine number of iterations for given input data.....	238
4.3 INTRODUCTION TO PROGRAMMING	241
Nature of programming languages	241
4.3.1 State the fundamental operations of a computer	241
4.3.2 Distinguish between fundamental and compound operations of a computer.....	241
4.3.3 Explain the essential features of a computer language	242
4.3.4 Explain the need for higher level languages	243
4.3.5 Outline the need for a translation process from a higher-level language to machine-executable code	244
Use of programming languages.....	244
4.3.6 Variable, constant, operator, object	245
4.3.7 Define various operators.....	245
4.3.8 Analyse the use of variables, constants and operators in algorithms.....	248
4.3.9 Develop algorithms using loops, branching	251
4.3.10 Describe the characteristics and applications of a collection	256
4.3.11 Develop algorithms using the access methods of a collection.....	256
4.3.12 Discuss the importance of sub-programmes and collections within programmed solutions.	258
4.3.13 Construct algorithms using pre-defined sub-programmes, one-dimensional arrays and/or collections.....	263

CHAPTER REFERENCES	290
TOPIC 5 OBJECT ORIENTED PROGRAMMING.....	291
D.1 OBJECTS AS A PROGRAMMING CONCEPT	291
D.1.1 The general nature of an object	291
D.1.2 Distinguishing between object and instantiation.....	292
D.1.3 & D.1.4 UML diagrams	294
D.1.5 Process of decomposition	300
D.1.6 Relationships between objects	302
D.1.7 Need to reduce dependencies between objects.....	305
D.1.8 Constructing related objects	307
D.1.9 Data types	309
D.1.10 Data items passed as parameters	310
D.2 FEATURES OF OOP	312
D.2.1 Encapsulation	312
D.2.2 Inheritance	312
D.2.3 Polymorphism	313
D.2.4 Advantages of encapsulation	314
D.2.5 Advantages of inheritance.....	315
D.2.6 Advantages of polymorphism	316
D.2.7 Advantages of libraries.....	316
D.2.8 Disadvantages of OOP	317
D.2.9 Use of programming teams.....	317
D.2.10 Advantages of modularity in program development	318
D.3 PROGRAM DEVELOPMENT	319
D.3.1 Class, identifier and variables.....	319
D.3.2 Method, accessor, mutator, constructor, signature and return value.....	320
D.3.3 Private, protected, public, extends and static.....	321
D.3.4 Uses of the primitive data types and the string class.....	322
D.3.5 Code examples for D.3.1 - D.3.4.....	326
D.3.6 Code example for selection statements.....	330
D.3.7 Code examples for repetition statements.....	333
D.3.8 Code examples of arrays	339
D.3.9 Features of programming languages that enable internationalization.....	347
D.3.10 Ethical and moral obligations of programmers.....	347
CHAPTER REFERENCES.....	349
APPENDIX A – INPUT/OUTPUT CLASS	350
APPENDIX A REFERENCES.....	353
APPENDIX B – BOOKSTORE PROGRAM	354
APPENDIX B REFERENCES.....	377
APPENDIX C – TEAM BUILDER SYSTEM	378
APPENDIX C REFERENCES.....	383
INDEX OF TERMS	384

TOPIC 1 – SYSTEM FUNDAMENTALS

© IBO
2012

Topic 1- system fundamentals¹

System life cycle

System life cycle refers to the stages through which the development of a new system passes through. Figure 1.1 presents a system life cycle specifically for software, whereas Figure 1.2 presents a more general system life cycle. Although most systems begin with the analysis stage and continue with the design, implementation, operation and maintenance, sometimes it might be necessary to backtrack and return to an earlier stage.²



Figure 1.1: Software life cycle

¹ International Baccalaureate Organization. (2012). IBDP Computer Science Guide.

² International Baccalaureate Organization. (2004). IBDP Computer Science Guide.

Although the software and system life cycles are not directly presented in the new IB Computer Science curriculum, they are briefly mentioned in the first section of this chapter so as to inspire some fruitful discussions between teachers and students.

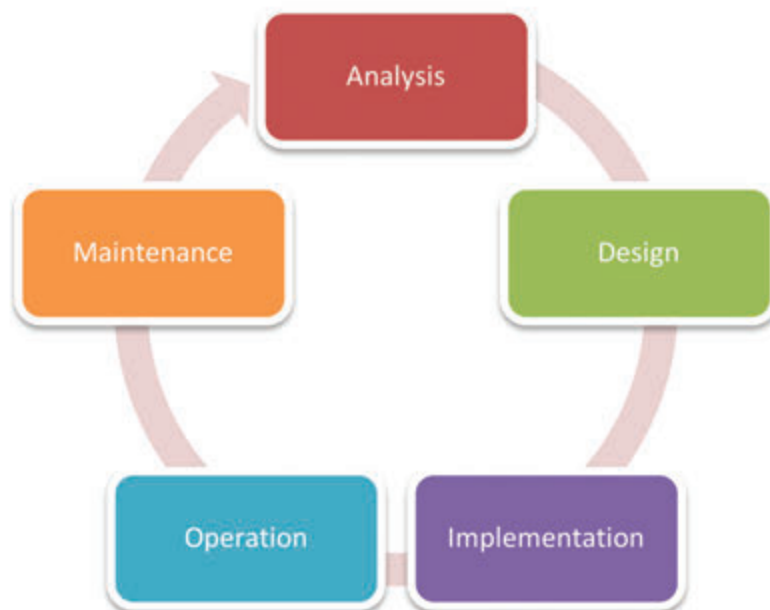


Figure 1.2: System life cycle



Figure 1.3: Some stages of the Software life cycle

The first Topic of the new computer science guide¹ and the section of the new computer science guide¹ that explain the internal assessment requirements directly refer only to four phases of the software life cycle as depicted in Figure 1.3.

- lack of attention to the design of tasks and jobs, allocation of information system tasks, and the overall usability of the system.

At this stage a feasibility report or feasibility study should be conducted. The feasibility study evaluates and analyzes a project and its potential, based on various technical, economical, legal, operational and scheduling criteria. It is used to decide whether the proposed project should be pursued.³

T	Technical feasibility: Is the existing technology sufficient to implement the proposed system?
E	Economic feasibility: Is the proposed system cost effective?
L	Legal feasibility: Are there any conflicts between the proposed system and any regulations/laws?
O	Operational feasibility: Are the existing organizational practices and procedures sufficient to support the maintenance and operation of the new system?
S	Schedule feasibility: How long will we wait?

Example 1.1:

Question: A bookstore uses door-to-door salespersons to collect various orders from various customers. The orders are taken to the company's offices and are input by a secretary. The bookstore has decided to change this department and will ask the salespersons to input the orders at home, using their personal computers. Discuss the various effects of these changes.

Answer: The bookstore will no longer need the secretary or any physical space to accommodate the relevant department. Utility bills will be reduced and probably a smaller office will be enough for the company. The secretary will have to find a new job, or another position in the company. Salespersons will have to acquire some computer skills and they will need to have a PC with an internet connection. New computer software and hardware will also have to be obtained by the company, which will also need to find a secure way (network) to allow the salespersons to connect to the server. Staff working from home may ask for extra pay.

1.1.2 The need for change management

Exit skills. Students should be able to ¹ :
Identify factors that are involved. Justify the need for change management. Investigate and outline success factors.

³ Feasibility study. (2014, November 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 18:03, November 23, 2014, from http://en.wikipedia.org/w/index.php?title=Feasibility_study&oldid=634775631

Change management involves various parameters and is a process of shifting individuals, teams, departments and organizations from the present state to a desired state. Successful change management guarantees that all stakeholders accept and embrace changes in their working environments. The goal is to maximize benefits and minimize the negative impacts of change on individuals.

For example, in a small business, if the operating system changes, the employees need to get proper training before they are able to use it in their everyday working schedule. In a school environment a new printer that is able to automatically print both sides could reduce costs but someone would have to inform teachers and students about this new functionality. Some people often feel threatened by a new completely computerized system, because they are afraid that they might lose their jobs. Unfortunately, sometimes their fears come true.

1.1.3 Compatibility issues resulting from situations including legacy systems or business mergers

Exit skills. Students should be able to ¹ :
Describe the importance of compatibility. Identify the way legacy systems interact with modern systems. Suggest strategies for successful integration during business mergers. Identify the international dimension of software compatibility.

The term *legacy system* refers to an old technology, hardware, computer system, or application program. Some systems that belong to this category may still play an important role in an organization. Such a system may still be in use because its data cannot be converted to newer formats, or its applications cannot be upgraded. Keeping a legacy system in operation involves various maintenance challenges. Even high technology agencies such as NASA use legacy systems because the system still provides for the users' needs, even though newer or more efficient technologies are available⁴. It is clear that the exchange of data between legacy and new systems is a factor that the administration needs to pay special attention to.

Business merger is the combining of two or more business entities. The main reason companies merge is to reduce costs. During this process all departments of both companies need to ensure that all subsystems are compatible.

⁴ (n.d.). Retrieved November 23, 2014, from https://www.fbo.gov/index?s=opportunity&mode=form&id=e2cd8e7c507a2bbd3614ede86beb5666&tab=core&_cview=0

Four Strategies for Integration⁵

1. Keep both information systems, and develop them to have the same functionality (high maintenance cost)
2. Replace both information systems with a new one (increased initial cost).
3. Select the best information systems from each company and combine them (it is very difficult for the employees to work with information systems from another company).
4. Select one company's information systems and drop the other companies' (policy problems).

Nowadays information technologies offer enormous potential for the world economy and society. Most organizations interact with individuals and other organizations that are located in different countries. *Language differences* greatly increase communication problems, even if individuals have some knowledge of the others' mother language. Language is not only a form of communication but also a way of thinking and defining the world.

Software incompatibility is a situation where different software entities or systems cannot operate satisfactorily, cooperatively or independently, on the same computer, or on different computers linked by a local or wide area computer network.

1.1.4 Different systems implementation

Exit skills. Students should be able to:

Define SaaS (Software-as-a-Service).

Define on premise software.

Relate and analyze the difference between SaaS (Software-as-a-Service) and on premise software solutions.



Image 1.2: SaaS

Business software can operate installed on a client's infrastructure and premises or hosted on dedicated servers that belong to a company that provides such services. A locally hosted system is the most appropriate solution for larger and complex systems. A remotely hosted system is the most

⁵ Legacy System Integration. (n.d.). Retrieved November 23, 2014, from <http://www.coleyconsulting.co.uk/merge.htm>

appropriate solution where there is no necessary hardware equipment in place or in cases where the administration wishes to outsource responsibilities for maintenance, support, backups, security, etc.

SaaS (Software-as-a-Service) or “on-demand software” is a contemporary delivery method that allows software and data to be hosted and managed centrally on a remote datacenter. Users pay to access the services provided on a subscription basis. SaaS solutions reside on the cloud and need a web browser and a broadband Internet connection to be accessed.⁶



Image 1.3: SaaS use versus on premise installation

SaaS is less expensive because it has a low initial cost and requires few investments in installation, maintenance and upgrading. Companies have to pay only for the SaaS subscription, which is cheaper in the short-to-medium term. SaaS provides a scalable solution, since a company has only to adjust its monthly SaaS subscription as required. In most cases only a web browser and a broadband internet connection are required to access SaaS applications. A wide range of desktop, portable and mobile devices also support SaaS solutions. SaaS requires few IT personnel and all software updates take place far away from

company’s premises. SaaS is considered a safe solution because applications and data reside in the cloud service of the provider's datacenter. However, there is a possibility of data loss, if a SaaS provider goes out of business. The performance of a web browser-based application that is hosted in a distant datacenter that is accessed via an Internet connection is low when compared to software running on a local machine or over the company’s local area network. SaaS integration with other SaaS solutions or software installed locally is always difficult.

1.1.5 Alternative installation processes

Exit skills. Students should be able to ¹ :
<p>Explain the major installation processes Suggest with reasons different installation processes Explain the pros and cons of each installation process</p>

The installation of a new system is a situation that most enterprises, organizations and individuals will face one or more times. During this process the old system will be retired and the new system will take its place.

⁶ Software as a service. (2014, November 17). In *Wikipedia, The Free Encyclopedia*. Retrieved 18:05, November 23, 2014, from http://en.wikipedia.org/w/index.php?title=Software_as_a_service&oldid=634189323

Example 1.2:

Question: A medical company is very satisfied from the technical performance of its current information technology infrastructure. Is there any reason to change it?

Answer: Maybe, the current system may be too expensive to run and maintain. So a new system may decrease the total expenses of the company.

Example 1.3:

Question: What is meant by the term “operational feasibility” found in a feasibility report?

Answer: A feasibility report should examine the “operational feasibility” of the proposed system. This part should examine whether the existing organizational practices and procedures are sufficient to support the maintenance and operation of the new system.

Example 1.4:

Question: What is meant by the term “merger”?

Answer: The combining of two or more corporations or business enterprises into a single corporation.

Example 1.5:

Question: What is meant by the term “software incompatibility”?

Answer: It is a situation where different software entities or systems cannot operate satisfactorily, cooperatively or independently, on the same computer, or on different computers linked by a local or wide area computer network.

Example 1.6:

Question: State five advantages of “SaaS”.

Answer: Lower initial cost, easy to upgrade, ease of access from anywhere, easy to predict the cost of initial implementation (subscription), the application is ready to use.

Example 1.7:

Question: State five possible disadvantages of “SaaS”.

Answer: Dependence on a third party, security and confidentiality, dependence on Internet connection, risk of data loss, not as powerful as on premise solutions.

One critical decision when moving from an old system to a new one is the choice of *implementation (conversion, changeover) method*. Changeover is the process of putting the new system online and retiring the old one. The reason for an organization to choose one implementation method in favour of another is often a trade-off between costs and risk. It should be mentioned that in most cases there might be resistance by employees or customers to change and planning should try to minimize the negative effects.

The types of changeovers are:

- *Parallel*. The main characteristic of Parallel Changeover is that both systems work in parallel for a short period of time. This method is very popular because of the limited risk. Outputs of both systems can be compared to ensure that the new system is functioning properly. If the new system fails, the company can revert or return to the old system. When the company is satisfied with the output of the new system, the old system can be terminated. Running two different systems simultaneously means extra costs and workload because it requires that the two systems run parallel for a certain period of time. The company has to maintain two different systems and this results in various organizational tasks. This method is not efficient if the old and the new systems have completely different processing tasks, functions, inputs, or outputs.
- *Big Bang or Direct (immediate)*. This changeover is very risky since the company plugs in the new system and unplugs the old one at the same time. Once the administration has decided to use this method and has prepared all the necessary procedures, the changeover begins. Obviously there are dangers associated with this method if the new system does not function as expected. This method is preferred when the system is not critical. With this approach, all users need to be trained appropriately before the switch takes place, in order to use the new information system efficiently.

Example 1.8:

Question: A nuclear station is equipped with a highly automated control system. This system should be replaced by a new system. Suggest a suitable implementation method. Justify your answer.

Answer: The correct answer is parallel changeover because the main characteristic of this conversion method is the limited risk. The cost of operation and maintenance of both systems will be higher but this is of limited importance in this particular situation. Direct method of implementation is not appropriate because of the high risk of failure and the absence of a trusted backup system. The scenario described in the question does not imply the existence of multiple sites so the pilot method is not considered as an alternative.

- *Pilot*. The pilot method is mostly used in large organizations that have multiple sites. The new system is introduced in one of the sites and extended to other sites over time. The risk of this method is low and the pilot sites can serve as models for the rest of the company. The first group that adopts the new system is called the pilot site or the pilot group. After the system proves successful at the pilot site, it is implemented into the rest of the company using a changeover method (usually direct).
- *Phased*. With the phased conversion method, a company converts one module of the system at a time, meaning that different parts of the system are converted at

different times. The training period is extended and the adoption of the new system takes longer, since each phase must be implemented separately. The installation of the new system is done per module, per department etc.

1.1.6 Problems that may arise as a part of data migration

Exit skills. Students should be able to¹:

Define data migration.
Identify the importance of incompatibility and incompleteness.
Provide a balanced analysis of different data migration scenarios.

Data migration refers to the transfer of data between different formats, storage types and computer systems. It usually takes place in an automatic manner so as to achieve efficient use of human resources. Data migration happens when an organization changes, upgrades or merges its information systems (for example, due to a merger or takeover).⁷ Many problems may arise when transferring data from one system to another. First of all, there may be incapability of moving the information due to parameters such as incompatibility with the new system or non-recognizable data structures. Also, data may be lost or not transferred due to an incomplete data transfer or errors during the process. In addition, data can also be misinterpreted due to incompatibilities, caused by the different conventions of each country concerning date, time and measurement units.



Image 1.4: Data migration

⁷ Database Glossary - D. (n.d.). Retrieved November 23, 2014, from <http://www.nwdatabase.com/database-glossary-d.htm>

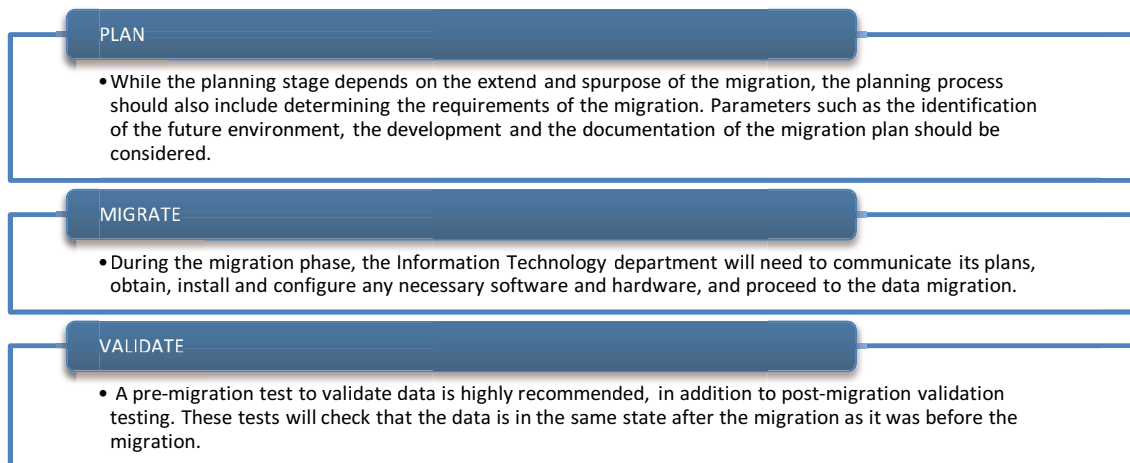


Figure 1.2: Data migration stages⁸

1.1.7 Various types of testing

Exit skills. Students should be able to ¹ :
Identify the importance of testing. Define major testing types. Suggest the best testing for a proposed scenario.

Testing can happen in different ways and in different phases.

Functional testing tests individual commands, text input, menu functions, etc. confirms that they perform and function correctly according to the design specifications. For example, if a choice is made to add a client, does the program go to the "add clients module"?

Data testing is when normal, abnormal and extreme data is put into the system. Suppose that there is a program that accepts a student's percentage in a math exam and gives a "pass" message if the number entered is greater than or equal to sixty; otherwise it prints a "fail" message. This program can be tested with^{9, 10}:

- *Normal Data* such as 76 will be used to check if "pass" and "fail" messages are appropriately provided.
- *Data at the Limits* should also be used, for this particular problem 0, 59, 60, 100 are all examples of normal data at the limits.

Useful Information: Some authors describe as "extreme/boundary data the data at the limits of acceptability/validity" and as "abnormal/erroneous data the data outside the limits of acceptability/validity." cf. David Watson and Helen Williams, 2012, page 79.

⁸ Adopted by: IBM, 2007, Best practices for data migration. <http://www-935.ibm.com/services/us/gts/pdf/bestpracticesfordatamigration-wp-gtw01275-usen-01-121307.pdf>
⁹ IB Computing Home Page. (n.d.). Retrieved November 23, 2014, from <http://www.ib-computing.com>
¹⁰ Watson D., H. Williams, 2012, Cambridge IGCSE Computer Studies Revision Guide, Cambridge University Press.

- *Extreme Data* will be outside the normal limits; -10, 104, 1223 are examples. These data should be rejected during validation testing. The user may not input such data because they're wrong, and it's easy to press a key twice by mistake.
- *Abnormal Data (illegal data)* will be the type of data that we really didn't expect. For this particular program it could be data that looks like a string, a character and not an integer. This data would not usually be entered. A naive user may enter "two", which seems unlikely but he/she could also hit the spacebar and enter "4 5", instead of "45", for example.

Alpha testing is done before the software product is made available to the general public. Normally, Alpha testing will be carried out by the company that develops the software in a laboratory type environment and not by the end users in their usual workplaces^{11, 12}.

Beta testing includes comments and suggestions of the users. Unlike Alpha testing, users outside the company are involved in the testing. Their feedback is valuable and can be used to fix defects and errors that were missed, and also contributes in the preparation of support teams that will deal with expected issues. Frequent beta testing results in last minute software changes. In some cases, the Beta version will be made available to the general public. This can provide vital real-world information and feedback^{11, 12}.

Dry-run testing is conducted using pen-and-paper by the programmer. During dry run testing the programmer mentally runs the algorithm. He examines the source code and decides on what the output of a run should be (execution).

During *unit testing*, individual parts of the system are tested separately.

During the *integration testing*, the entire system is tested at the same time to verify that that all components can work together.

User acceptance testing is used to determine if the system satisfies the customer needs and in most cases is conducted in user premises before accepting transfer of ownership. This type of testing is the last stage of the software testing process.

Debugging is a systematic process of finding and correcting the number of bugs (errors) in a computer program.

It is important to mention that there are computer programs that can automatically test other programs. This makes the testing process faster and cheaper.

¹¹ Alpha vs. Beta Testing. (n.d.). Retrieved November 23, 2014, from <http://www.centercode.com/blog/2011/01/alpha-vs-beta-testing/>

¹² (n.d.). Retrieved November 23, 2014, from <http://ezinearticles.com/?Alpha-Testing-and-Beta-Testing&id=433>

Example 1.9:

Question: Several verification and validation checks are performed on data being entered into a database by a bank employee. The database contains an “age” field and a “name” field. Give two examples of invalid data and one example of valid data for the field “age”, and explain how this field could be validated.

Answer: All “age” entries should be checked to see if they are positive numbers and for consistency with a minimum/maximum range (e.g. 18-110). This process includes a type and a range check. Invalid data for the age field could be “Y” and “234” while valid data would be “25”.

Question: How the “name” field could be verified?

Answer: It could be typed twice and the two entries compared. If both entries are the same then the “name” is verified. This process is called double entry and reduces data entry errors.

Validation and verification in relation to data input²

- *Validation* is the process of evaluating whether data input follows appropriate specifications and is within reasonable limits.
- *Verification* is the process of ensuring that the data input is the same as the original source data. A way of ensuring data verification is through double entry.

Verification vs. validation software testing

Verification is the confirmation that a computer product meets identified specifications, while validation is the confirmation that a computer product meets its design function or is appropriate for the intended use.

- Validation: Are we developing the correct system?
- Verification: Are we developing the system correctly?
- Validation: Does the product meet the customer-needs?
- Verification: Does the product comply with a specific regulation or condition?

User focus**1.1.8 Importance of user documentation**

Exit skills. Students should be able to¹:

Identify the importance of proper user documentation.
Describe the way proper documentation affects implementation.

Programs and systems may become increasingly complex as their aim gets more and more and convoluted. Additionally, the user's way of thinking may differ from the developers'; hence the way the product functions may not be clear to the user. All of the above may lead to serious problems during the implementation or operational stage of the Software Life Cycle (SLC). The user might not know how to use the product to its full capacity, and even specialized technicians may not be able to set up the new system properly for the firm they work for. With appropriate external documentation, these issues can be easily countered, since there will be a user manual explaining every component of the product.

A systematic and organized software development procedure ensures that all supporting documents are produced in an orderly and methodical fashion. *It is very important to distinguish between internal documentation and external documentation.*

Internal documentation is the code comprehension features and details provided as part of the source code itself. Proper internal documentation includes:

- Appropriate module headers
- Appropriate comments
- Useful and meaningful variable names
- Useful module, classes, methods and function headers
- Appropriate code indentation
- Appropriate code structuring
- Appropriate use of enumerated types
- Appropriate use of constant identifiers
- Appropriate use of data types defined by the user

It is very difficult and likely impossible for the source code to be read by the final user.

External documentation is typically written as a separate document from the program itself. It is provided through various types of user supporting documents such as a users' guide, software requirements specification document, detailed description of the design and implementation features of the program and test document.

1.1.9 Different methods of providing user documentation

Exit skills. Students should be able to ¹ :
Identify the importance of proper user documentation. Suggest various methods of user documentation. Describe the way propped documentation affect implementation.

User documentation can either be in a written or in an online form, so that the user can search the document more easily and quickly. The user documentation should include all the instructions that are mandatory for the system to operate and should contain frequently asked questions, which are always necessary for new users. The document should be well structured and divided into the appropriate categories.



Image 1.5: Different methods of user documentation

Manuals

These can be provided online or offline. One of the advantages of online manuals is the potential use of multimedia features. Manuals can also be provided online as pdf files which one can download and print. The advantage of a printed manual is that it can be read without the use of a PC or an internet connection.

Email support

Online support is an asynchronous type of support and can be provided via an email address. Users contact the support team of the company to resolve any problems with the help

of specialized technicians.

Embedded Assistance or integrated user assistance

Software suites like MS Office have inbuilt help systems (tool tips and dynamic page content within the system itself). One example is when the user hovers the mouse over an icon a small text box appears with valuable help information. This kind of assistance is considered an excellent way to increase the usability of a software application. Embedded user assistance is context-specific, task-specific, and does not require novice users to ask the right question to find the suitable answer.

Frequently Asked Questions

Frequently Asked Questions (FAQ) are listed questions and answers, all supposed to give users answers to a common set of problems and pertaining to a particular topic. The format is commonly used on online forums. The list of questions contains questions that tend to recur.

Live chat sessions

Online support is a type of real time support that is extremely useful for emergency situations. A live chat technician will ask for the description of the occurring problem, and try to present a list of possible solutions. A telephone call, a live chat session or a video session provides a feeling of being supported by a real person, which is preferred by many users.

Online portals or web portals

Online support is provided in many ways, depending on the product or service that is being documented. Online portals can provide updated manuals, support pages and FAQ pages.

Remote desktop connections

Remote Desktop is a function that enables a specialized person to connect to the user's PC across the Internet from virtually any computer. Remote Desktop will actually allow a specialized technician to take control of the user's PC as though he/she was sitting directly in front of it. This solution is ideal if the user is not very experienced. This solution has some security disadvantages because the technician is allowed to have full access over the user's PC.

1.1.10 Different methods of delivering user training

Exit skills. Students should be able to¹:
Identify the importance of proper user training
Suggest various methods of user training
Describe the way propped user training affect implementation

It is impossible to take full advantage of a new advanced IT system without proper training. Moreover even the highest investment in technology requires users that are able and wish to use it. User training is extremely important in almost every case.

All the staff must be familiar with the new system as they will make mistakes if they are not properly trained. Staff can be trained by self-instruction, formal classes or online training. Furthermore, the developers of the new software have to create clear educational material for solving any kind of questions a user might have.

User training can be delivered in a number of different ways depending on a variety of factors, such as the number of the students, the availability of instructors, the size of the business, and the training budget.

Self-Instruction or self-study

Self-instruction allows the user to learn in his/her own time, until he/she achieves mastery of the subject. Printed manuals, books, e-books or other resources such as video tutorials or online exercises can be provided and used whenever the user needs to improve his skills.

A user can benefit a lot through self-study. First of all, there is no tuition fee. Furthermore the user can study whenever he/she wants (no formal class at a fixed time and a fixed place). The disadvantages include lack of guidance or teacher support and the final result depends on the motivation of the user and their ability to learn on their own.



Image 1.6: Different methods of user training

Formal classes

A formal class offers an interactive setting that promotes open and free discussion between students and the teacher (instructor). Having several students learning in the same classroom has the additional advantage of allowing students to exchange ideas with one another. Direct interaction with the expert allows for ideas to be exchanged easily and without any technical communication barriers.

A classroom situation may disadvantage shy members. The classroom can also

obstruct one's ability to learn by allowing other, more self-assured students to dominate the discussion environment.

Remote/distance learning/online training¹³

The main benefit of asynchronous online learning is that it allows participants to take part in high quality courses from anywhere in the world provided they have a PC and Internet connection. This type of Virtual Classroom is accessible 24/7/365. Time efficiency and time management are valuable strengths of distance learning. Students can access their virtual courses, lectures, course materials, and class discussions at any time, day or night. The use of interactive learning environments contributes to self-direction and promotes critical thinking, and thus is highly supported by the literature of adult education and training.

An online educational program requires participants who are able to access the online learning environment. Lack of the required infrastructure will exclude otherwise eligible students from the online course. Students and instructors must possess a minimum level of IT skills in order to function effectively in an online environment. Online asynchronous education places greater responsibility on the student and gives students control over their learning experience, and thus is considered inappropriate for more dependent and immature learners.

¹³ Illinois Online Network: Educational Resources. (n.d.). Retrieved November 23, 2014, from <http://www.ion.uillinois.edu/resources/tutorials/overview/>