JARAMOGI OGINGA ODINGA UNIVERSITY OF SCIENCE AND TECHNOLOGY

UNIVERSITY EXAMINATIONS 2015/2016

SCHOOL OF INFORMATICS AND INNOVATIVE SYSTEMS

2ND YEAR 1ST SEMESTER EXAMINATIONS FOR THE DEGREE OF BACHELOR OF EDUCATION (ARTS & SCIENCE), BUSINESS ADMINISTRATION WITH IT

MAIN – RESIT

COURSE CODE: **SCS 203**

COURSE TITLE: **INFORMATION SYSTEM ANALYSIS**

EXAM VENUE: **STREAM: BBA, Bed Arts & Sci.**

DATE : 04/05/ 2016 **EXAM SESSION: 2.00 – 4.00 PM**

TIME: 2.00 HOURS

INSTRUCTIONS:

1. **Answer ALL question 1 (Compulsory) and ANY other two questions**
2. **Candidates are advised not to write on the question paper**
3. **Candidates must hand in their answer booklets to the invigilator while in the examination room.**

QUESTION ONE [30 MARKS]

a) Define term information systems (IS)

b) Discuss different types of feasibility analysis.

c) Explain the technology, people, and organizational components of an information system.

d) List and explain the different phases in the systems development life cycle.

e) Why is it important to use systems analysis and design methodologies when building a system?

f) Describe the activities performed by the project manager during project initiation.

(**2+6+6+6+6+4 Marks**)

QUESTION TWO [20 MARKS]

a) What is contained in a Baseline Project Plan and are the content and format of all baseline plans the same?

b) Describe the structured walkthrough process and state the roles that need to be performed during a walkthrough?

c) Describe systems analysis and the major activities that occur during this phase of the systems development life cycle.

(8+8+4 Marks)

QUESTION THREE [20 MARKS]

a) What are some useful character traits for an analyst involved in requirements determination?

b) What are the general guidelines for designing questionnaires?

c) Compare collecting information by interview and by questionnaire. Describe a hypothetical situation in which each of these methods would be an effective way to collect information system requirements.

(4+6+10 Marks)

QUESTION FOUR [20 MARKS]

a) Describe how prototyping can be used during requirements determination and outline if it is better or worse than traditional methods?

b) What is a data flow diagram and why do systems analysts use them?

c) Explain the convention for naming different levels of data flow diagrams.

d) How can data flow diagrams be used as analysis tools?

(5+5+5+5 Marks)

QUESTION FIVE [20 MARKS]

a) Describe the process of designing interfaces and dialogues and what deliverables are produced from this process?

b) Differentiate between system implementation and system maintenance

c) Discuss Four characteristics of a good requirement specification document

(8+6+6 Marks)

MARKING SCHEME

QUESTION ONE

a) Define term information systems (IS)

A system is an organized relationship among functional units or components which work in harmony to achieve a common goal.

b) Discuss different types of feasibility analysis.

Types of Feasibility Analysis

i) Operational Feasibility

A measure of how well the solution of problems will work in the organization. (Is the problem worth solving?) It is also a measure of how people feel about the system project.

ii) Technical Feasibility

A measure of the practicality of a specific technical solution and the availability of technical resources and expertise.

iii) Schedule Feasibility

A measure of how reasonable the projected timetable is.

iv) Economic Feasibility

A measure of the cost effectiveness of a project or solution. It takes into account costs and benefits. Thus it is often called Cost-Benefit Analysis.

c) Explain the technology, people, and organizational components of an information system.

The answer should include: Data, Software, Hardware, Telecommunications and People

d) List and explain the different phases in the systems development life cycle.

Phases in system development life cycle

i. **Problem definition – it involves identification of a problem and selection as appropriate from a number of other problems as it serves a critical function in the organization.**
ii. **Feasibility study – The feasibility study is used to determine if the project should get the go-ahead. If the project is to proceed, the feasibility study will produce a project plan and budget estimates for the future stages of development.**
iii. **Detailed analysis - Analysis gathers the requirements for the system. This stage includes a detailed study of the business needs of the organization. Options for changing the business process may be considered.**
iv. **System Design - Design focuses on high level design like, what programs are needed and how are they going to interact, low-level design (how the individual programs are going to work), interface design (what are the interfaces going to look like) and data design (what data will be required). During these phases, the software's overall structure is defined.**

v. **System Implementation - In this phase the designs are translated into code. Computer programs are written using a conventional programming language or an application generator. Programming tools like Compilers, Interpreters, and Debuggers are used to generate the code. Different high level programming languages like C, C++, Pascal, and Java are used for coding. With respect to the type of application, the right programming language is chosen.**

vi. **System Testing - In this phase the system is tested. Normally programs are written as a series of individual modules, this subject to separate and detailed test. The system is then tested as a whole. The separate modules are brought together and tested as a complete system. The system is tested to ensure that interfaces between modules work (integration testing), the system works on the intended platform and with the expected volume of data (volume testing) and that the system does what the user requires (acceptance/beta testing).**

vii. **System maintenance - System Maintenance – This is normally done to enable the software to accommodate the changing trends in an organization so as to suite different user needs. It is also done to remove some of the mistakes that might have occurred during the implementation phase.**

e) Why is it important to use systems analysis and design methodologies when building a system?

Methodology: A *standard process* followed in *an* organization to conduct all the *steps* necessary to analyze, design, implement, and maintain information systems. It involves a collection of models, tools, and techniques

Models are;

a) **Representation of an important aspect of real world, but not same as real thing**

b) **Abstraction used to separate out aspect**

c) **Diagrams and charts**

Techniques are;

a) **Collection of guidelines that help analyst complete system development activity or task**

b) **Can be step-by-step instructions or just general advice**

Tools are;

a) **Software support that helps create models or other required project components**

b) Range from simple drawing programs (e.g., Visio, PowerPoint) to complex CASE tools

f) Describe the activities performed by the project manager during project initiation.

The activities include:

a) Establish project initiation team

b) Establish relationship with customer

c) Establish project initiation plan

d) Establish management procedures

e) Establish project management environment and workbook

(2+6+6+6+6+4 Marks)

QUESTION TWO

a) What is contained in a Baseline Project Plan and are the content and format of all baseline plans the same?

Assures that customer and development group have a complete understanding of the proposed system and requirements

Provides sponsoring organization with a clear idea of scope, benefits and duration of project

Four Sections

- **Introduction**
- **System Description**
- **Feasibility Assessment**
- **Management Issues**

Baseline Project Plan (BPP): internal document

- **Scope**
- **Benefits**
- **Costs**
- **Risks**

- **Resources**

Statement of Work (SOW): Outlines objectives and constraints of the project to the customer

- **Describes deliverables**
- **Outlines work needed to be performed**

b) Describe the structured walkthrough process and state the roles that need to be performed during a walkthrough?

Walkthrough

1. **Peer group review**
2. **Participants**
    i. **Coordinator**
    ii. **Presenter**
    iii. **User**
    iv. **Secretary**
    v. **Standards Bearer**
    vi. **Maintenance Oracle**
3. **Activities**
    i. **Walkthrough review form**
    ii. **Individuals polled**
    iii. **Walkthrough action list**
4. **Advantages**
    i. **Assures that review occurs during project**

c) Describe systems analysis and the major activities that occur during this phase of the systems development life cycle.

The Oxford Dictionary defines analysis as follows: separation of a substance into parts for study and interpretation; detailed examination.

In the case of systems analysis, the 'substance' is the business system under investigation, and the parts are the various subsystems that work together to support the business.

System Analysis is the process of understanding and specifying in detail what the information system should do.

This deals with analysis of sets of interacting entities prior to their automation as computer systems, and the interactions within those systems.

The analysis phase answers the questions of who will use the system, what the system will do, and where and when it will be used.

(8+8+4 Marks)

QUESTION THREE

a) What are some useful character traits for an analyst involved in requirements determination?

The analyst should posses some of the following traits;

Analytical

- **Understanding of organizations**
- **Problem-solving skills**
- **System thinking : Ability to see organizations and information systems as systems**

Technical

- **Understanding of potential and limitations of technology**

Managerial

- **Ability to manage projects, resources, risk and change**

Interpersonal

- **Effective written and oral communication skills**

b) What are the general guidelines for designing questionnaires?

- **Interviewing and Listening**
- **Gather facts, opinions and speculations**
- **Observe body language and emotions**

- **Guidelines**
  - **Plan**
    - **Checklist**
    - **Appointment**
  - **Be neutral**
  - **Listen**
  - **Seek a diverse view**
- **Interview Questions**
  - **Open-Ended**
    - **No pre-specified answers**
  - **Close-Ended**
    - **Respondent is asked to choose from a set of specified responses**

c) Compare collecting information by interview and by questionnaire. Describe a hypothetical situation in which each of these methods would be an effective way to collect information system requirements.

Characteristics	Interviews	Questionnaires
Information richness	High(many channels)	Low to moderate
Time required	Can be extensive	Low to moderate
Expense	Can be high	moderate
Chance for following up and probing	Good: Probing and clarification questions can be asked by either interviewer and interviewee	Limited: Probing and follow up done after original data collection
Confidential	Interviewee is known to interviewer	Respondent can be unknown
Involvement of subject	Interviewee is involved and committed	Respondent is passive, no clear commitment
Potential audience	Limited numbers, but complete responses from those interviewed	Can be quite large, but lack of response from some can bias results

(4+6+10 Marks)

QUESTION FOUR

a) Describe how prototyping can be used during requirements determination and outline if it is better or worse than traditional methods?

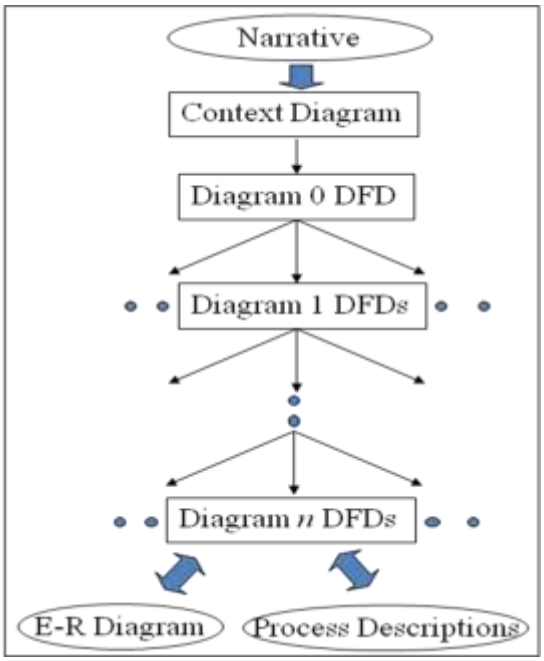
Prototyping:

- **The rapid development and testing of working models**
- **Used in design phase**
- **Especially useful when end user requirements are hard to define**
- **Can be used for small and large systems**
    - **But if system is large, usually prototype just parts**
- **Develop quickly**
- **Refine until acceptable**

b) What is a data flow diagram and why do systems analysts use them?

Data Flow Diagrams (DFDs) are the link between initial business analysis. DFDs allow the analyst to determine: *What* does the organization do? *How* does the organization do it?

c) Explain the convention for naming different levels of data flow diagrams.



- **Create a *narrative*: description of system**
- **Create a *Context Diagram* that contains a single process ("the system") and all entities which share data with the system**

- ***Explode* the "parent" context diagram to produce a Diagram 0 ("child") DFD**
- **Create Diagram 1, 2, …, *n* DFDs that represent "explosions" of Diagram 0, 1, …, *n*-1 DFDs until a diagram has only "primitive" processes**
- **Create *process descriptions* to be implemented by application programs: queries, macros, reports, programming languages**

d) How can data flow diagrams be used as analysis tools?

Gap Analysis

- **The process of discovering discrepancies between two or more sets of data flow diagrams or discrepancies within a single DFD**

Inefficiencies in a system can often be identified through DFDs

(5+5+5+5 Marks)

QUESTION FIVE

i. Describe the process of designing interfaces and dialogues and what deliverables are produced from this process?

This is a list of goals that a developer should try and have in their software design. Some goals contradict other ones, but that is where you have to decide what is best for your program.

Minimal Complexity

The main goal in any program should be to minimize complexity. As a developer most of your time will be maintaining or upgrading existing code. If it is a complete mess, then your life is going to be that much harder. Try and avoid those solutions where you use a one line complex solution to replace 20 lines of easy to read code. In a year, when you come back to that code, it will take you that much longer to figure out what you did.

Ease of Maintenance

This is making your code easy to update. Find where your code is most likely going to change, and make it easy to update. The easier you make it, the easier your life is going to be down the road. Think of it as a little insurance for your code.

Loose Coupling

So What is loose coupling? It is when one portion of code is not dependant on another to run properly. It is bundling code into nice little self reliant packages that don't rely on any outside code. How do you do this? Make good use of abstraction and information hiding.

Extensibility

This means that you design your program so that you can add or remove elements from your program without disturbing the underlying structure of the program. A good example would be a plug-in.

Reusability

Write code that will be able to be used in unrelated projects. Save yourself some time. Again information hiding is your best bet for making this happens.

High Fan-in

This refers to having a large number of classes that use a given class. This implies that you are making good use of utility classes. For example you might have a bunch of classes that use the Math class to do calculations.

Low to medium Fan-out

This refers to having a low to medium amount of classes used by any given class. If you had a class that includes 7 or more classes this is considered high fan out. So try and keep the number of classes you include down to a minimum. Having a high fan-out suggests that the design may be too complex.

Portability

Simply put, design a system that can be moved to another environment. This isn't always a requirement of a program, but it should be considered. It might make your life easier if you find out your program does have to work on different platforms.

Leanness

Leanness means making the design with no extra parts. Everything that is within the design has to be there. This is generally a goal if you have speed and efficiency in mind. A good example of where this might come in handy is creating a program that has to run on a system with limited resources (cell phone, older computers)

Standard Techniques

Try and standardize your code. If each developer puts in their own flavor of code you will end up with an unwieldy mess. Try to layout common approaches for developers to follow, and it will give your code a sense of familiarity for all developers working on it.

ii. Differentiate between system implementation and system maintenance

System implementation refers to the actual transfer of the program or system design into the code. An appropriate programming language is chosen and used to write the program.

System Maintenance – This is normally done to enable the software to accommodate the changing trends in an organization so as to suite different user needs. It is also done to remove some of the mistakes that might have occurred during the implementation phase.

iii. Discuss Four characteristics of a good requirement specification document

Unambiguous

- **Every requirement has only one interpretation.**
- **Each characteristic of the final product is described using a single unique term.**
- **A glossary should be used when a term used in a particular context could have multiple meanings.**

Complete

A complete SRS must possess the following qualities:

1. **inclusion of all significant requirements,**
2. **definition of the responses of the software to all realizable classes of input,**
3. **conformity to any standard that applies to it,**
4. **full labeling and referencing of all tables and diagrams and the definition of all terms.**

Verifiable

- **Every requirement must be verifiable.**
- **There must exist some finite cost-effective process with which a person or machine can check that the software meets the requirement.**

Consistent

- **No set of individual requirements described in the SRS can be in conflict.**

Modifiable

- **The structure and style of the SRS are such that any necessary changes to the requirements can be made easily, completely and consistently.**
- **Requirements:**
    1. **a coherent and easy-to-use organization (including a table of contents, index and cross-referencing),**

2. **Not be redundant - this can lead to errors.**

Traceable

- **The origin of each requirement must be clear.**
- **The SRS should facilitate the referencing of each requirement in future development or enhancement documentation.**
- **Types:**
    1. **Backward traceability**
        - **Each requirement must explicitly reference its source in previous documents.**
    2. **Forward traceability**
        - **Each requirement must have an unique name or reference number.**

Usable during the operation and maintenance phase

- **The SRS must address the needs of the operation and maintenance phase, including the eventual replacement of the software.**

(8+6+6 Marks)