1. Learning Outcomes of Level 5 Software Architecture 5N2772

Learners will be able to:

1. Demonstrate an understanding of general systems theory to the design of information systems in organisations

2. Discuss the impact of organisational structure on the design of information systems

3. Explain the various stages and associated roles, of the traditional software development lifecycle from feasibility/investigation through end-of- maintenance/support

4. Describe the role of modern methodologies in the software development process

5. Demonstrate an understanding of the management and technology issues, including risks and common pitfalls, involved in the design and construction of software systems

6. Use straightforward UML use-case diagrams to depict the interaction between an actor and system

7. Engage a modern methodology to aid the design of a system from initial requirements

8. Apply the main tools and techniques used in the gathering, recording and analysis of information relating to an existing information system

9. Generate Java (or suitable Object- Oriented language) classes from UML class diagram specifications

10. Test the design specification of a new information system based on a provided use-case

11. Explain the development of an information system using a traditional systems development life cycle model

2. Indicative Content

This section provides suggestions for programme content but is not intended to be prescriptive. The programme module can be delivered through classroom based learning activities, group discussions, one-to-one tutorials, field trips, case studies, role play and other suitable activities, as appropriate.

Section 1 : Introduction to Systems Development
1,2,5

Learning outcome 1 - Demonstrate an understanding of general systems theory to the design of information systems in organisations

Facilitate the learner to:

- Discover the history and aims of General Systems Theory
- Define Information Systems and the different types of information systems prevalent in Organisations
- Analyse in detail the history of Information Systems development for a period of time e.g. 1970 to present

Learning outcome 2 - Discuss the impact of organisational structure on the design of information systems

Facilitate the learner to:

- Explore the different types of Organisational Structure e.g. Hierarchical, Flat, Matrix, Networked
- Examine what type of organizational structure tends to be most willing to embrace technological change and sophistication and why.

Learning outcome 5 - Demonstrate an understanding of the management and technology issues, including risks and common pitfalls, involved in the design and construction of software systems

Facilitate the learner to:

- Examine the importance of having business case for starting the project
- Examine the importance of tool for analysing viability and feasibility of construction of a software system. e.g. Cost-Benefit analysis , Contingency plan.
- Outline the importance of setting project goals e.g., Time, Budget
- Examine the importance estimating resources needed
- Discover the impact of poorly defined system requirements
- Poor communication among customers, developers, and users

Section 2: Systems Development Methodologies
3,4,11

Learning outcome 3 - Explain the various stages and associated roles, of the traditional software development lifecycle from feasibility/investigation through end-of-maintenance/support

Facilitate the learner to:

- **Explore the main stages in the Traditional SDLC Namely**
    - Stage 1: Feasibility analysis
    - Stage 2: Requirement analysis and specification
    - Stage 3: Designing the product architecture
    - Stage 4: Building or Developing the Product
    - Stage 5: Testing
    - Stage 6: Deployment and Maintenance
- Discuss the different variations of the SDLC eg. Waterfall, V-Shaped Model, Incremental Model, Spiral Method (SDM)

Learning outcome 4 - Describe the role of modern methodologies in the software development process

Facilitate the learner to:

- Examine the history and evolution of the Agile Software Development methodologies
- Discover and discuss the main principles of the Agile manifesto
- Examine some of the main Agile Software Development methods e.g. Scrum, Kaban Method, Dynamic systems development method (DSDM) ,Extreme Programming (XP),
- Outline the advantages and disadvantages of Agile Development methodologies over more traditional SDLC methodologies e.g. Waterfall approach

Learning outcome 11 - Explain the development of an information system using a traditional systems development life cycle model

Facilitate the learner to:

- Using a traditional SDLC outline the stages in the development of a current information system using this approach e.g. Waterfall SDLC
- For each stage in the methodology explain the main activities that took place

Section 3:Systems Modelling
6,7,8,9,10

Learning outcome 6 - Use straightforward UML use-case diagrams to depict the interaction between an actor and system

Facilitate the learner to:

- Discover what a Use Case is.
- Discover and define the elements that make up a UML use-case diagram e.g
  - Actors
  - Use-Case
  - System boundary
  - Relationships (extends, includes)
- Understand the difference between the different elements and how to elicit them from a business use case
- Create Use-Case diagrams from a set of business rules or requirements.

Learning outcome 7 - Engage a modern methodology to aid the design of a system from initial requirements

Facilitate the learner to:

- Design and develop a piece of Software using an Agile Software Development Methodology
- Document the stages, processes, procedures, meetings and tools used from your chosen modern methodology in Designing you piece of software

Learning outcome 8 - Apply the main tools and techniques used in the gathering, recording and analysis of information relating to an existing information system

Facilitate the learner to:

- **Discover the tools and techniques that are used gather and record information about a current information system e.g.**
  - Use Cases
  - Stakeholder identification
  - Stakeholder Interviews
  - Questionnaires
  - Data- Gathering
  - Observation
  - Document Analysis

- Use a relevant selection of the above tools and techniques to gather information about an existing information system

Learning outcome 9 - Generate Java (or suitable Object-Oriented language) classes from UML class diagram specifications

- Discover what a class diagram is and what the main elements are that make up a class diagram. e. g Attributes, operations, relationships between classes
- Examine the purpose of class diagrams and who in the SDLC uses them.
- Examine the difference between platform dependent and platform independent class diagrams.
- Use a suitable IDE to Forward Engineer a class diagram to create the related class or classes
- Discuss the relationship between class diagrams and code, and classes that are engineered from them.

Learning outcome 10 - Test the design specification of a new information system based on a provided use-case

Facilitate the learner to:

- Examine the importance and benefits of testing
- Derive a test case from a provided use case to include all or some of the following elements
    - ID
    - Name
    - Actor
    - Description
    - Priority
    - Frequency of use
    - Preconditions
    - Typical workflow
    - Exception workflows
    - Post conditions