Structured Analysis and Design

Synopsis:

Changes to systems often begin from changes in needs and requirements. Requirements Engineering is the process of discovering requirements, analyzing requirements for incompleteness, inconsistency, relevance and practicality and negotiating the final requirements for the systems. In this course, we shall discuss a structured approach to understanding and documenting requirements and techniques for translating the requirements into computer-based design specifications. In particular, we shall study into data and process models, process modeling techniques, structured module design, quality assurance checks (Coupling and Cohesion), testing and testing techniques.

Who should attend?

Systems Analysts, Software Developers, Business Analysts, Programmers

Objectives

1. Define requirements
2. Learn about data and process modeling techniques
3. Learn how to analyze problem requirements and document them
4. Learn how to translate requirements into software design
5. Learn about software testing techniques

Prerequisites

Have a strong understanding of how a computer program works

Class Organization

Classroom teaching cum workshop

Duration

3 days

Course Outline:

1. INFORMATION SYSTEMS

Define information systems
Explain the Systems Development Life Cycle

2. INVESTIGATION

Understanding the Existing System
Preliminary Investigation
Feasibility Study
Detailed Investigation (Requirements Definition)
Information Gathering Techniques
Structured Walkthroughs

3. REQUIREMENTS DEFINITION

Explain Three-Step Modelling Process: Abstracting, Analyzing, and Representing
Explain the three Perspectives of an Information System: Process, Data, Behaviour (or Event)

4. PROCESS MODELLING TECHNIQUES

Describe Data Flow Diagram (DFD)
Explain components of a Data Flow Diagram: Process, Data Flow, Data Store, External Entity
Explain Process Descriptions using Structured English, Decision Table and Decision Tree
Describe Data Dictionary
Explain what is defined in a Data Dictionary
Explain data definition in Data Dictionary using Sequence, Selection and Repetition structures
Define Data Elements, Data Types, Data Structures
Explain how to combine, split or merge data flows
Explain common mistakes made

Explain how to level data flow diagrams into Context Diagram, Level 1 Diagram, Level 2 Diagram
Explain how to balance data flow diagrams
Case Study: Practise drawing data flow diagrams and process descriptions

5. **DATA MODELLING TECHNIQUES**
Define Entity and Relationships
Describe Entity Relationship Diagram
Describe how to choose entity names
Define Entity Descriptions, Entity Attributes, Entity Keys
Define Attribute Definitions, Attribute Descriptions
Define Domain Definitions, Relationship Definitions
Define One-to-One Relationships, One-to-Many Relationships
Describe how to resolve One-to-One Relationships and Many-to-Many Relationships
Explain how to remove redundant relationships
Case Study: Practise drawing Entity Relationship Diagrams

6. **PROCESS DESIGN**
Describe process design strategy
Describe Structure Chart as a tool for visualizing process design
Describe Transform Analysis and Transaction Analysis
Case Study: Practise process design

7. **QUALITY ASSURANCE CHECKS**
Define Coupling and Cohesion
Describe the various types of Coupling: Data, Stamp, Control, Common, Content
Describe the various types of Cohesion: Functional, Sequential, Communicational, Procedural, Temporal, Logical, and Coincidental

8. **TESTINGS**
Explain Testing Strategies: Unit Testing, Integration Testing, System Testing
Explain Testing Techniques: Black-Box Testing and White-Box Testing

9. **SUMMARY**
Summarize important points taught in course
Review course objectives and how they are met

Description of course outline uses the following keywords to explain the coverage of the course

Define, Describe, Explain:	Refers to the theoretical explanation of concepts covered
Discuss:	Explain concepts in detail with some level of interaction from the students
Practise:	Practical exercises to let students practise the concepts taught
Summarize, Review:	Re-visiting the concepts covered

Course Leader

Dr. Danny Poo graduated with a BSc (Hons), MSc and PhD in Computer Science from the University of Manchester Institute of Science and Technology (UMIST), England. He is currently a tenured Associate Professor in the Department of Information Systems, National University of Singapore and teaches a course on Object-Oriented Software Engineering at the undergraduate level. Prior to this, he taught a course on “Systems Analysis and Design” to classes that exceed 350 students. He is presently the Vice-Chairman, Steering Committee for the Asia-Pacific Software Engineering Conference and is actively involved in teaching professionals from the computing industry in Object-Oriented Programming. He is the founder and director of Cicada Cube Pte Ltd, an NUS spin-off company specializing in Enterprise-level Search and Retrieval Solutions. A well-known speaker in seminars, he has conducted numerous in-house training for organizations both locally and regionally. Dr. Poo has consulted for local and multi-national companies including: OCBC Bank, Rhodes-Schwarz, Gemplus, Tech Semiconductor, Micron Semiconductor, Monetary Authority of Singapore, Sema Group, Singapore Polytechnic, Ministry of Defence (Singapore), Port of Singapore Authority, and FTMS (Malaysia). **Dr. Poo has co-written three books** “**Object-Oriented Programming and Java**”, Springer-Verlag, 1998; “**Developing Systems Using J2EE**”, Prentice-Hall, 2004 and “**Learn To Program Java**”, Prentice-Hall, 2004. His fourth book on “**Learn To Program Java User Interface**” will be published in January 2005 by Prentice-Hall.