You will learn about:

- THE SOFTWARE DEVELOPMENT PROCESS
- STRUCTURED AND OBJECT-ORIENTED DEVELOPMENT METHODS
- USER REQUIREMENTS DETERMINATION AND SPECIFICATION
- MODELLING OF BUSINESS FUNCTIONS WITH DATA FLOW DIAGRAMS AND WITH USE CASE DIAGRAMS
- MODELLING OF DATA STRUCTURES WITH ENTITY-RELATIONSHIP DIAGRAMS AND WITH OBJECT DIAGRAMS
- PROS AND CONS OF VISUAL MODELLING WITH CASE TOOLS
- METHODS OF DEVELOPING MAINTAINABLE AND SCALABLE SYSTEMS
- DESIGNING RELATIONAL AND OBJECT-RELATIONAL DATABASES
- DESIGNING GRAPHICAL USER INTERFACES
- IMPLEMENTING CLIENT/SERVER APPLICATIONS

Some of possible customisations and extensions of this workshop for in-house courses:

- SYSTEMS ANALYSIS (3 DAYS)
- RELATIONAL DATABASE DESIGN (3 DAYS)
- OBJECT-ORIENTED ANALYSIS AND DESIGN (3 DAYS)
- FUNDAMENTALS OF OBJECT-ORIENTED APPLICATION DEVELOPMENT (1 DAY) - for managers and project leaders
- WHAT CAN (AND WILL?) GO WRONG IN OBJECT-ORIENTED DESIGN AND IMPLEMENTATION (1 DAY) - for programmers and designers
- OBJECT-RELATIONAL DATABASES (1 DAY) - for relational database developers

Australian organisations that benefited from this workshop (or its customised version) include:

- in Sydney:
  Leeds & Northrup, CSIRO Australia Telescope, Foxboro, Australian Independent Media Data, Nortell, Tower Technology, Qantas
- in Brisbane:
  Comalko Smelting, CITEC, Bank of Queensland
- in Canberra:
  Department of Social Security, Department of Defence, Centrelink
About the presenter

- **Dr. Leszek A. Maciaszek** - Associate Professor in Computing, Macquarie University, Sydney.
- More than 25 years of **experience** (14 in Australian context) in teaching, research and development of information systems and database applications.
- **Consultant** to corporations and organisations in Australia, USA, Europe, and Asia (current engagements concentrate on providing leadership in the development of client/server object-oriented database systems).
- **Regular presenter** of industry workshops and seminars in Australia, Asia, and Europe.
- **Author** of over seventy publications on databases, object technology, systems analysis and design, workgroup computing and software engineering.
- The **initiator** and the **leading developer** of the CASE workbench IDDK (Intelligent Database Design Kit).
- Member of **Program Committees** and **speaker** at some of the most reputable international conferences in databases, software engineering, and information systems development.
- **Referee** of papers and articles submitted to major international journals, such as ACM Transactions on Database Systems, IEEE Software, The VLDB Journal, Object Oriented Systems, Software Practice & Experience.
- **Researcher** in database technology, object technology, and information systems development.
- **Academic teacher** of advanced undergraduate and postgraduate courses, mainly in database design and programming and in systems analysis and software engineering.
**COURSE CONTENTS**

1 **Introduction to software process:**
   - components of information systems: people, processes, data, software, hardware
   - software development lifecycle
   - method vs process
   - structured vs object approach

2 **Strategic planning and business modeling:**
   - mission statement, objectives, goals
   - strategies and policies
   - effectiveness vs efficiency
   - business system planning

3 **Requirements determination:**
   - system scope and system users
   - communication, interpersonal skills, interviewing, presentation skills
   - requirements analysis techniques
   - function and information requirements
   - other requirements (look and feel, usability, performance, maintainability, security, political, legal)
   - joint application development, group dynamics

4 **Requirements specification:**
   - specification techniques
   - template of a specification document
   - Data Flow Diagrams
   - Use Cases

   **Workshop in Process Modelling with Data Flow Diagrams and with Use Cases**
   - Entity-Relationship Diagrams
   - Class Diagrams

   **Workshop in Information Modelling with Entity-Relationship Diagrams and with Class Diagrams**
   - testing and metrics for specification phase

5 **Working with visual modelling tools:**
   - capability maturity model
   - organisation maturity vs visual modelling technology
   - visual modelling as a workgroup computing environment
   - forward and reverse engineering

   **Demonstration of upper-engineering visual modelling (CASE) tools for systems analysis**
6
**Project planning:**
- risk and feasibility analysis
- project management techniques
- guidelines for successful project planning
- techniques of project cost estimation

7
**Advanced analysis:**
- advanced information modeling
- model-view-controller (MVC) approach
- dynamic modeling
- integrating function, information and dynamic models
- analysis for different application domains
- using class libraries, patterns, and application frameworks

8
**Developing maintainable and scalable systems:**
- essence and accidents of software engineering
- system complexity
- software components and re-use
- proper and improper uses of inheritance
- the power of composition (aggregation)
- design patterns

9
**Database design:**
- client/server architectures
- relational vs object databases
- design of database structures
- design of database-wide business rules
- database program design

*Demonstration of lower-engineering CASE tools for database design*

*Workshop in relational and object-relational database design*

10
**Client/Server application design:**
- techniques of user-computer interaction
- principles of good interface design
- interaction techniques, menus, controls and toolbars
- primary and secondary windows
- object collaboration diagrams
- modelling collaboration of interface and database objects

*Workshop in client/server program design with extended Object Collaboration Diagrams (to design the collaboration of GUI and database objects).*
11 Architectural design:
• coupling and cohesion
• component diagrams
• deployment diagrams
• three-tier architectures
• architectures for data warehouses
• testing and metrics for design phase

12 Implementation:
• implementation planning
• software package evaluation and acquisition
• programming team organization
• test types
• walkthroughs and inspections
• performance evaluation
• leading visual modelling (CASE) tools
• leading relational, object-relational, and object databases