ESSENTIAL PRACTICES

OBJECT-ORIENTED SOFTWARE ENGINEERING

- This training course provides software engineers with a comprehensive understanding of Object-Oriented technology. The range of techniques, UML 2.0 notation, principles and current trends in Object-Oriented software engineering are covered. The technical explanation of OOSE is combined with examples, exercises and 'in practice' guidelines.
- This course is a formal training session designed for qualified software engineers.
- The course is a total of three days duration and can be conducted in multiples of half-day sessions.

COURSE OUTLINE

Object-Oriented Analysis:

- Process/Workflow Modelling
- Use Case Analysis
- Scenarios
- Object Identification Techniques

OO Design with UML 2.0:

- Static Design
  - Class Diagram
  - Object Diagram
  - Package Diagram
  - Composite Diagram
- Dynamic Design
  - Sequence Diagram
  - Communication Diagram
  - Interaction Overview Diagram
  - Timing Diagram
  - Activity Diagram
  - State Machine Diagram
- Physical Design
  - Component Diagram
  - Deployment Diagram

Practical Sessions:

Hands-on workshop exercises are undertaken for these topics to gain a more practical appreciation

Design Principles:

- Open/Closed Principle
- Liskov Substitution Principle
- Principle of Dependency Inversion
- A Class Captures One Key Abstraction
- Generalise up the Inheritance Hierarchy
- Interface-Segregation Principle

Design Patterns:

- Benefits of Patterns
- Creational Patterns
- Structural Patterns
- Behavioural Patterns

Software Architecture:

- Architectural Requirements
- Architectural Design Principles
- Pattern Driven
- Logical & Physical Architecture
- Frameworks

OO Implementation & Testing:

- OO Languages
- Testing Throughout the Development Lifecycles
- OO Test Drivers & Test Frameworks

STRATEGIC SYSTEMS (WA) Pty Ltd
TRAINING SERVICES