Applying UML 2.0, OOAD & Agile Practices

Introduction
- Why use models.
- Difference between model and methodology.
- What is the Unified Modeling Language?
- The 3 Amigos and their work.
- Introducing UML 2.0, the notation.
- Identifying business processes.
- Notation, Patterns and Methodology.
- Which Methodology to choose?

Fundamental Concepts
- Building from components.
- Modeling concepts.
- What is an object?
- Containment.
- Messages and methods.
- Object interaction.
- Exercise: testing some basic concepts.

Inception
- The Unified Process.
- Inception artifacts.
- The four phases .
- Planning the inception phase.
- Development of a business vision.
- Overview of Object Oriented Analysis & Design.
- Actors, Use Cases and the System.
- Creation of Use Cases.

Requirements Analysis
- Requirements gathering.
- Tools and techniques for identification and analysis of requirements.
- FURPS guidelines.
- Identifying business objects.
- Use-Case driven Requirements analysis.

Use Case driven Requirements
- What is use case modeling?
- Main & alternative scenarios.
- Goals and stories.
- Use case diagrams.
- Exercise: use case for the course project.
- Use case types & formats.
- Actors and system.
- Exercise: create a detailed use case.
- Applying the EBP guideline.

Other Requirements
- The supplementary specification.
- What goes into the supplementary specification?
- Making a Glossary document.
- Where to begin?
- Deciding on a Go/No-Go for the Project.

Use Case workflow modeling
- Activity diagrams.
- Convenience features.
- Exercise: create activity diagr. for the use case.
- Iteration completed.
- Detailing the next steps.

Elaboration
- Elaboration artifacts.
- Main activities during elaboration.
- Planning the elaboration phase.
- The design model.
- Structuring of a high-level business use-case.
- Describing detailed Use Cases.

Sequence Diagrams
- Sequence diagrams to detail the Use Case.
- The System sequence diagram.
- Emphasis on the time-ordered flow.
- System events.
- Inter-system events usage.
- UML Sequence diagram notation and events.
- Exercise: create a system sequence diagram.

Domain Model
- Definition of the domain model.
- Modeling concepts in a domain.
- Purpose of the domain model.
- Exercise: find concepts for the course project.
- Identifying attributes for the domain model.
- Concepts or attributes?
- Adding associations to the domain model.
- Multiplicity and roles.
- Exercise: adding associations & attributes.
- Specification classes.
- Exercise: add a specification class to domain.

Operation Contracts
- Operation contracts.
- When to use an operation contract?
- The 5 categories for an operation contract.
- Detailing pre-and post conditions.
- Operation contract guidelines.

GRASP Patterns
- What are GRASP patterns?
- The design of behavior.
- Assigning responsibilities to classes and objects.
- Pattern resources.
- Identifying the 5 first patterns.
- The Ying/Yang of modeling.
- GRASP versus GOF patterns.
- Artifact relationships.
- Exercise: create a collaboration diagram.
- Updating the domain model.

Organizing the Domain model
- When to split the domain model.
- Criteria to group conceptual classes together.
- Using packages to organize the domain model.
- Exercise: create service packages.
- Linking the domain model to the collaborations.

From Analysis to Design
- Wrap up of the Analysis activities.
- Communication between team members.
- Sharing work between analyst & developer.
- The way forward: design tasks for the developer.

Applying UML 2.0, OOAD & Agile Practices

Using the Input from the Analysts
- Operation contracts.
- The domain model.
- Assigning responsibilities to classes and objects.
- Meta class pattern.
- Identifying the 5 first patterns.
- The Ying/Yang of modeling.
- Exercise: create collaboration diagram.

Interaction Diagram Specifics
- Detailing object behavior.
- The link between message and method.
- Associations and links.
- Message sequencing.
- Conditional messages.
- Operations translated in collaboration diagrams.
- UML Objects and messages.
- Notation of message structure and iteration.

Object Visibility
- When to establish visibility between objects.
- Attribute and parameter visibility.
- Global and Local visibility.
- Which type of visibility to use when?

Design Classes
- From domain model to class diagram.
- Adding methods to the class diagram.
- Showing temporary visibility.
- From class diagram to code.
- What about method signatures?
- Exercise: create a class diagram.

OCL (Object Constraint Language)
- When to use object constraint language?
- Using inv, context, pre and post.
- Using collections.
- OCL and executable UML.
- Tools that support OCL.

Package Usage
- Case study of an airline reservation system.
- Exercise: create the domain model.
- What are subsystems?
- Grouping classes into subsystems.
- Criteria to group classes into subsystems.

Relationships between Use Cases
- Extending and Including Use Cases.
- Abstract Use Cases.
- Inheritance between Actors and Use Cases.

Fine tuning the Domain Model
- Composition and Aggregation.
- Association classes.
- Qualified associations.
- Inheritance and Specializations.

Additional Patterns and their usage
- Polymorphism and pure fabrication.
- Indirection and other advanced patterns.
- Applying patterns to the domain model.

Construction
- Positioning the current phase.
- Main activities during construction.
- Overview of construction artifacts.
- Planning the construction phase.
- Construction templates.

Coding Phase
- Tips and tricks for creating code from classes.
- Defining classes with collections.
- Order of Implementation.
- Detailing method signatures for the developer.
- Creating methods from collaboration diagrams.

State Diagrams in Construction
- When to create state charts.
- Identification of state and transitions.
- Guard conditions & sub states.
- Exercise: model a digital stopwatch.
- History marker.
- Internal transitions.
- Exercise: model a public phone system.

Layered Architecture
- The layers pattern.
- Classic three-their architecture.
- Connecting to the domain layer.
- Linking to the User interface.
- Using packages to decompose a system.
- Avoiding mutual dependencies.

Transition
- Key ideas.
- Beta testing and Pilot phase
- Getting feedback from the pilot
- Test evaluation and test planning
- Test cases and test procedures
- Minor adjustments based upon feedback

GOF (Gang of Four) Patterns
- When to use Design patterns.
- Some common examples.
- GRASP versus GOF patterns.
- Other patterns.
- Exercise: find the composite pattern.
- The State and Singleton pattern.

Case Study
- Requirements Gathering.
- Creation of Use Cases, high level and detailed.
- Making the Sequence diagrams.
- How to obtain a Domain model.
- Operation contracts.
- Deriving Collaboration diagrams.
- Updating the Class diagrams.
- Elaborating some sample code.

Conclusions
- When is UML really useful?
- How to plan the different phases.
- When does the project end?