



## Introduction to Object-Oriented Analysis and Design

Lecture : 3a  
Date : 4-11-2005

Yannis Tzitzikas  
University of Crete, Fall 2005

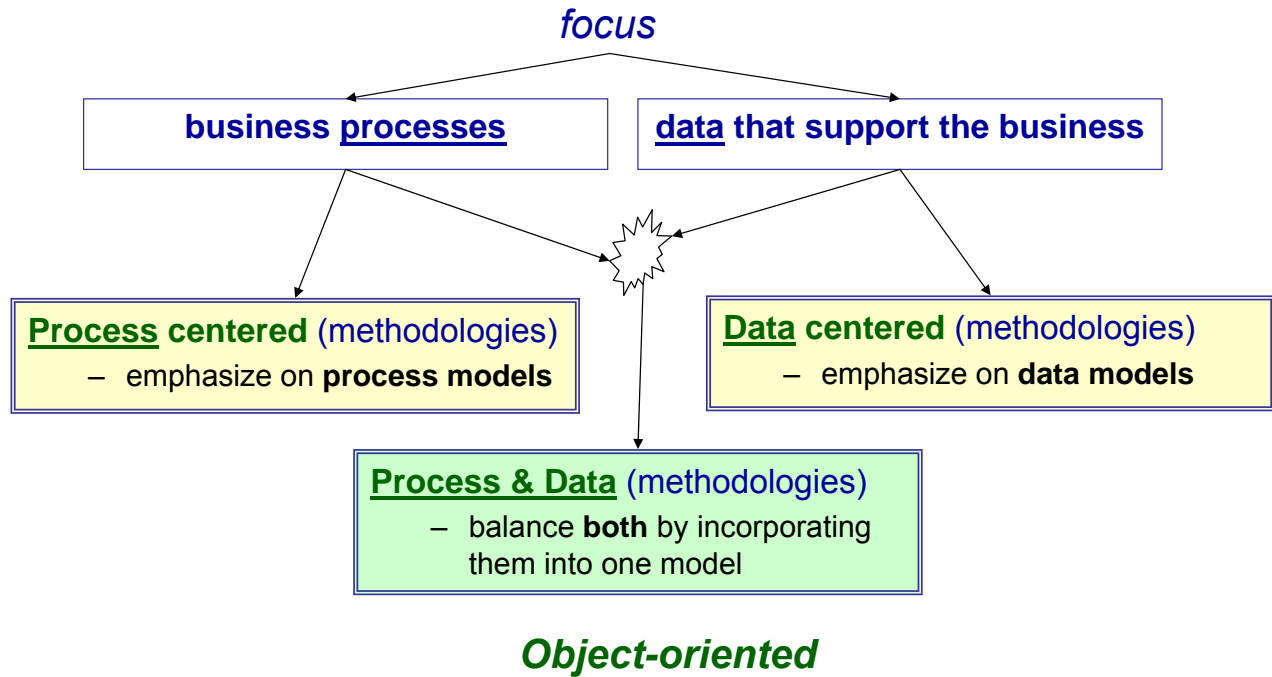


## Outline

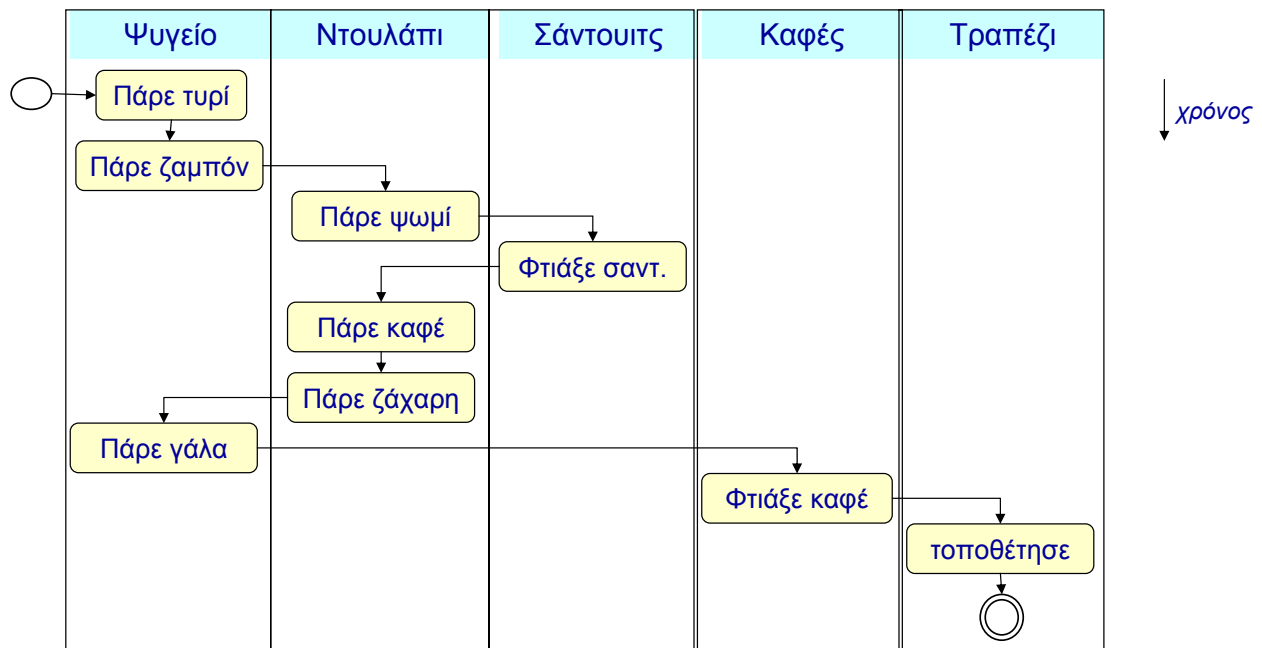
- Categorizing methodologies w.r.t their focus
  - process centered
  - data centered
  - object-oriented
- Basic Characteristics of OO Systems
- Object-Oriented Analysis and Design
- Object-Oriented Analysis and Design and Software Development Methodologies



# The focus of a methodology

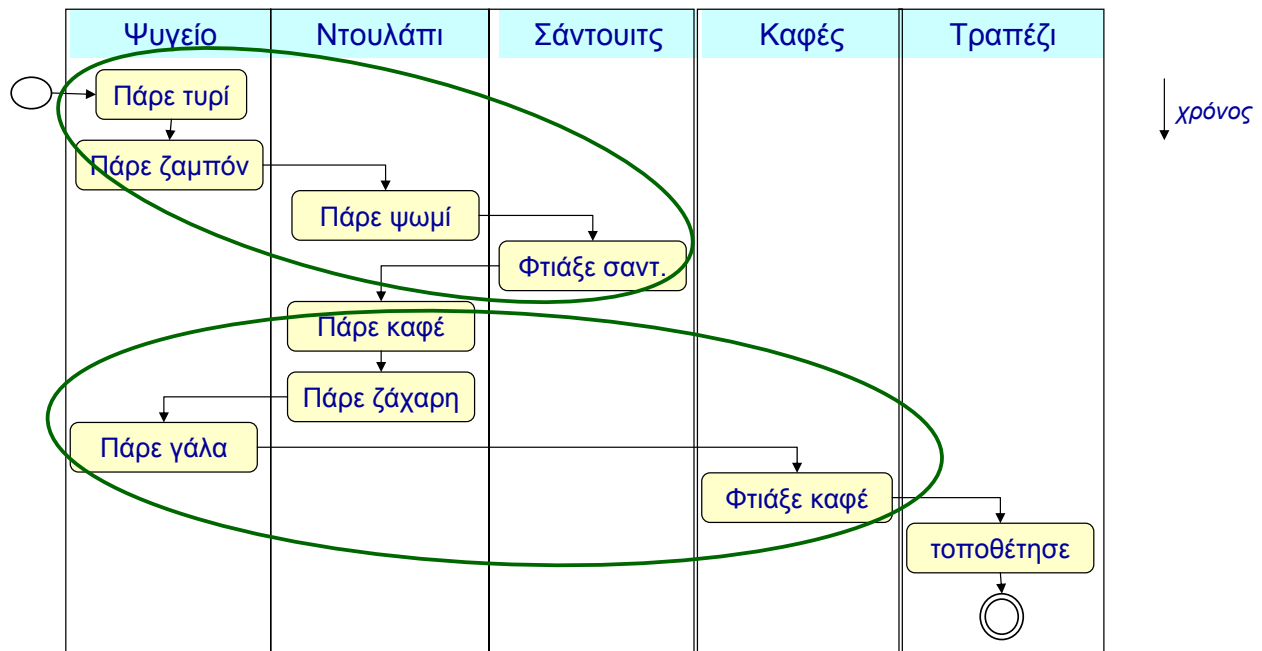


# Παράδειγμα: Ετοιμασία πρωινού





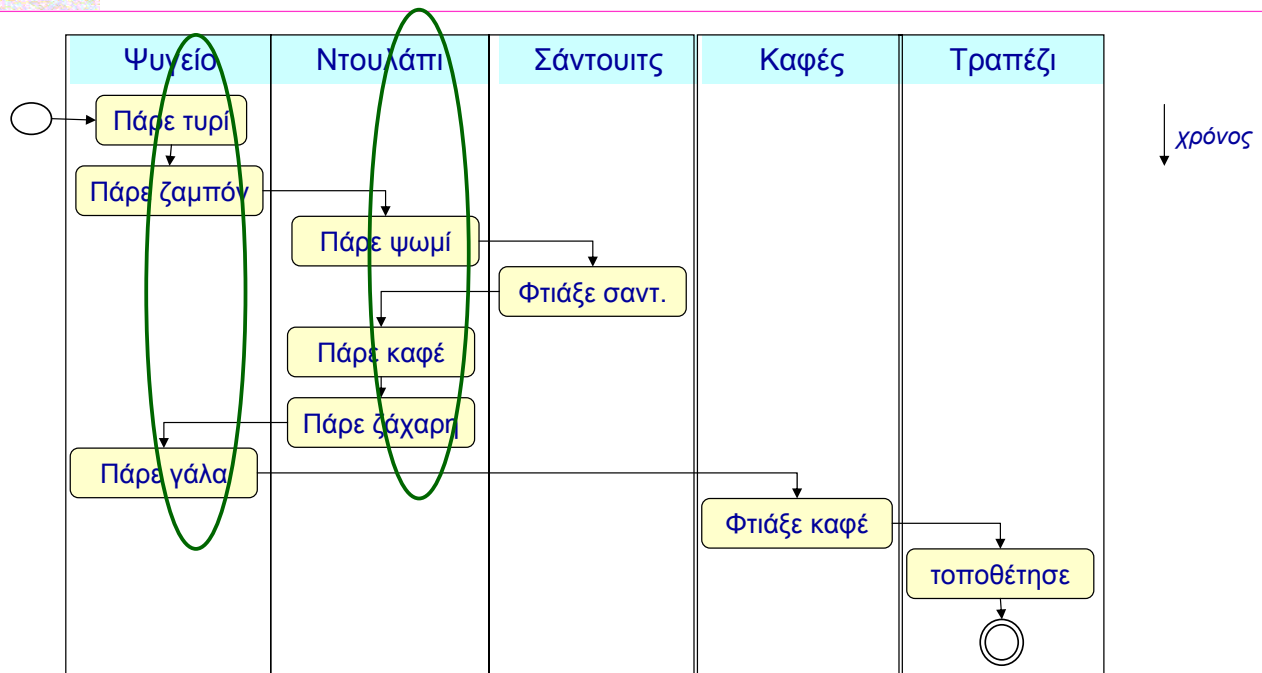
## The emphasis of a Process-centered methodology



Ορισμός διαδικασιών: **ετοιμασία Σαντουιτς, ετοιμασία Καφέ**



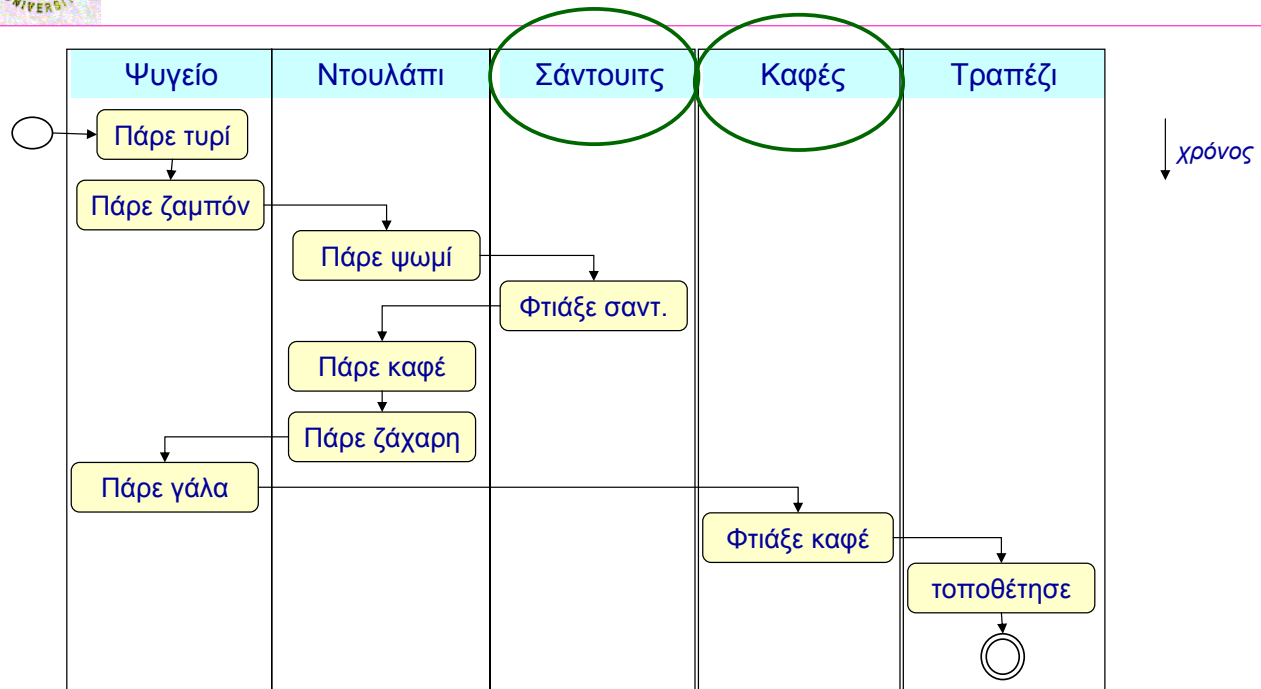
## The emphasis of a Data-centered methodology



Ορισμός περιεχομένων: **Ψυγείου, Ντουλαπιού**



## The emphasis of an Object-oriented methodology



Ορισμός κυρίαρχων εννοιών: **Σάντουιτς, Καφές**  
και κατόπιν μελέτη των σχετιζόμενων δεδομένων και διαδικασιών



## Basic Characteristics of OO systems

Focus on capturing  
the structure and the behaviour  
in little modules that encompass  
both data and process.  
These modules are known as objects.



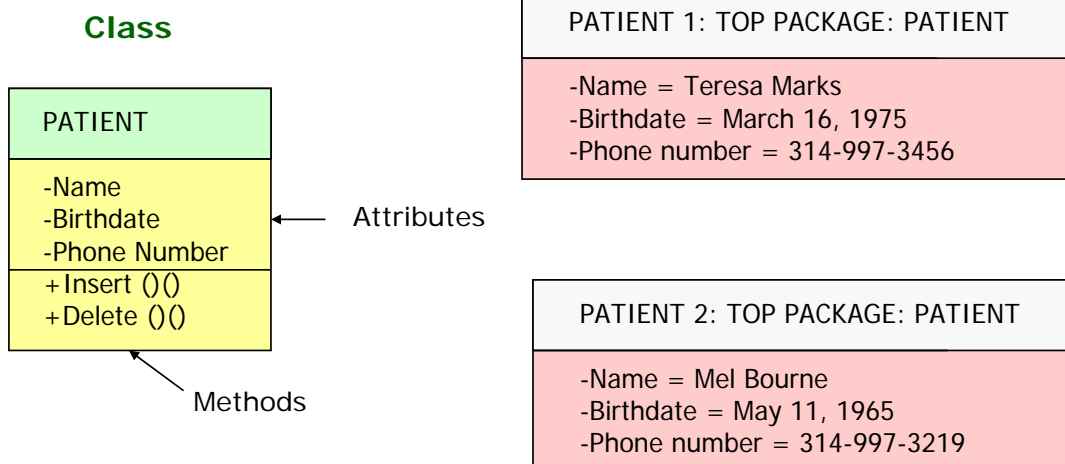
# Basic Characteristics of OO Systems

- **Classes** -- template to define objects
- **Instances** -- specific examples of class members
- **Objects** -- building block of the system
- **Attributes** -- describe data aspects of the object
- **Methods** -- the processes the object can perform
- **Messages** -- instructions sent to or received from other objects



# Classes and Objects: Example

## Instantiated Objects of the Class



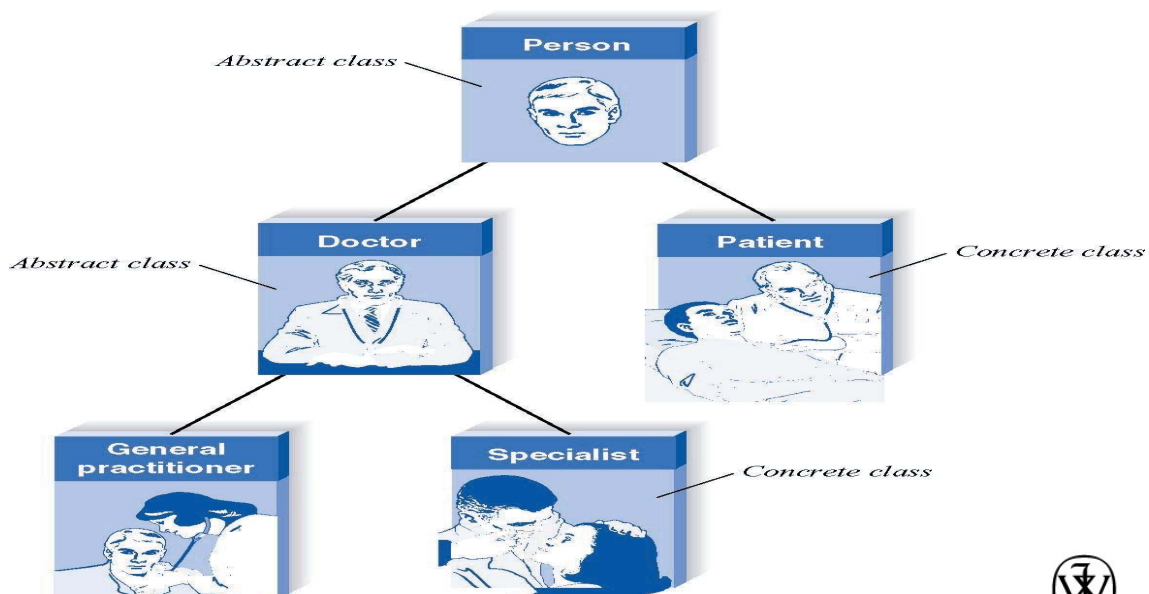


# The Key to Reusability

- **Encapsulation**
  - Combining data and processes in a single object
- **Information hiding**
  - Showing only the information required to use the object
    - What we gain ?

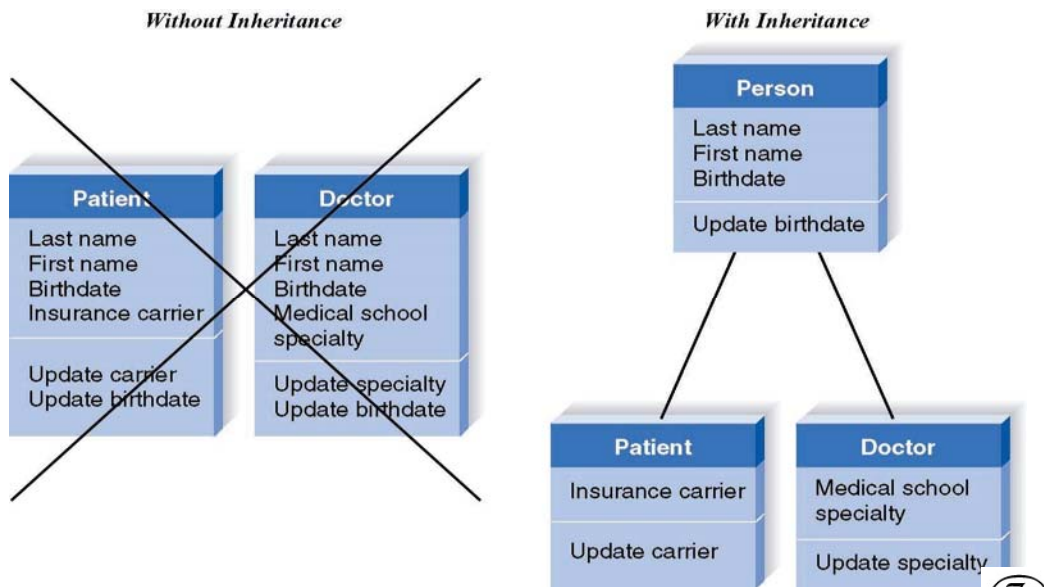


# Class Hierarchy





# Inheritance

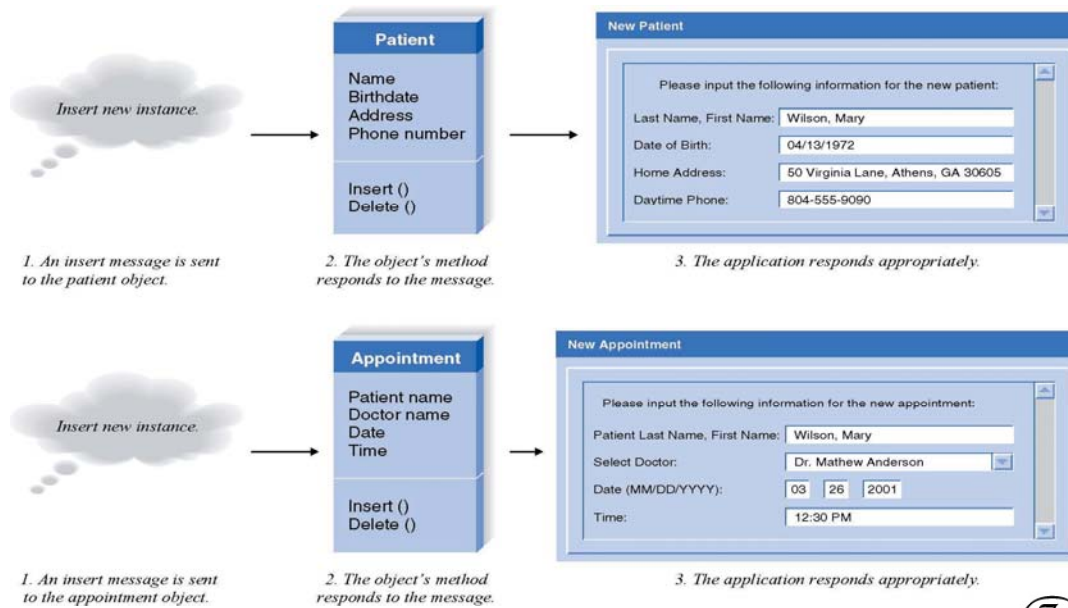


# Polymorphism and dynamic binding

- **Polymorphism**
  - the same message can be interpreted differently by different classes of objects
- **Dynamic binding**
  - dynamic (or late) binding is a technique that delays typing the object until run-time.



# Polymorphism and dynamic binding



# Characteristics of some widely used languages

Feature	C++	Eiffel	Java
strong typing	Optional	yes	yes
static(s)/dynamic(d) typing	S	S	S+D
garbage collection	no	yes	yes
multiple inheritance	yes	yes	no
pure objects	no	yes	no
dynamic loading	no	no	yes
standardized class libraries	no	yes	yes
correctness constructs	no	yes	no





## Other benefits of object-orientation

- Aids reuse
- makes code more modular
- best suited for GUIs and event-driven programming



## Object-Oriented Analysis and Design

***Attempts to balance emphasis  
on data and process  
and uses UML for diagramming***



## Object-Oriented Analysis and Design and Software Development Methodologies

- OO approaches for developing information systems could use any of the traditional methodologies that we saw
  - Waterfall, parallel, phased, prototyping, throwaway prototyping, RAD, XP, ..
- However they are most associated with a phased development RAD methodology



## Object-Oriented Analysis and Design

- According to the creator of ULM, any modern oo approach to developing information systems should be
  - **Use-case Driven**
  - **Architecture Centric**
    - the underlying software architecture drives the specification, construction and documentation of the system
    - Should support at least 3 separated but interrelated architectural views
      - **Functional**
      - **Static**
      - **Dynamic**
  - **Iterative and Incremental**
    - software is not released in one big bag at the end
    - Each iteration of the system brings it close and closer to real user need



## Reading and References

- **Systems Analysis and Design with UML Version 2.0** (2nd edition) by A. Dennis, B. Haley Wixom, D. Tegarden, Wiley, 2005. CHAPTER 2
- **Requirements Analysis and System Design** (2nd edition) by Leszek A. Maciaszek, Addison Wesley, 2005, CHAPTER 1
- **Object-Oriented Systems Analysis and Design Using UML** (2nd edition) by S. Bennett, S. McRobb, R. Farmer, McGraw Hil, 2002, Chapter 4
- Shari Lawrence Pfleeger. Τεχνολογία Λογισμικού: Θεωρία και Πράξη, 1. Κλειδάριθμος, Αθήνα, 2003, Κεφάλαιο 6