

Subject:

Object-oriented analysis and design

Chapter 1: Introduction to OOAD

Lê Văn Vinh, PhD

Department of Software Engineering

Faculty of Information Technology

HCMC University of Technology and Education

Contents

I.

Software development

II.

Basic concepts of OOAD

III.

OOAD Approadches

Software development

❖ **All software development requires the following things to occur:**

■ Requirements

- Requirements capture/determination: During this stage, the needs of the customers are determined, including a case-by-case analysis of the functionality required.
- Requirements analysis: During this stage the functionality is described in terms of interactions between the user and the system. Relationships between related functional requirements may also be determined at this stage.

Software development

❖ **All software development requires the following things to occur:**

■ Design

- System design: The overall system structure is determined, including a description of modules (subsystems) and sub-modules and their interaction
- Class-level design: The classes are designed which are to make up the modules, etc.
- User interface design: The appearance and operation of the user interface is determined
- Data design: The structure of the database model is determined

Software development

❖ **All software development requires the following things to occur:**

■ Construction

- Implementation: At this stage, the design is converted into an implementation using a (normally object-oriented) programming language
- Testing: At this stage, the implementation is tested for coding and integration errors, as well as conformity to the requirements and design

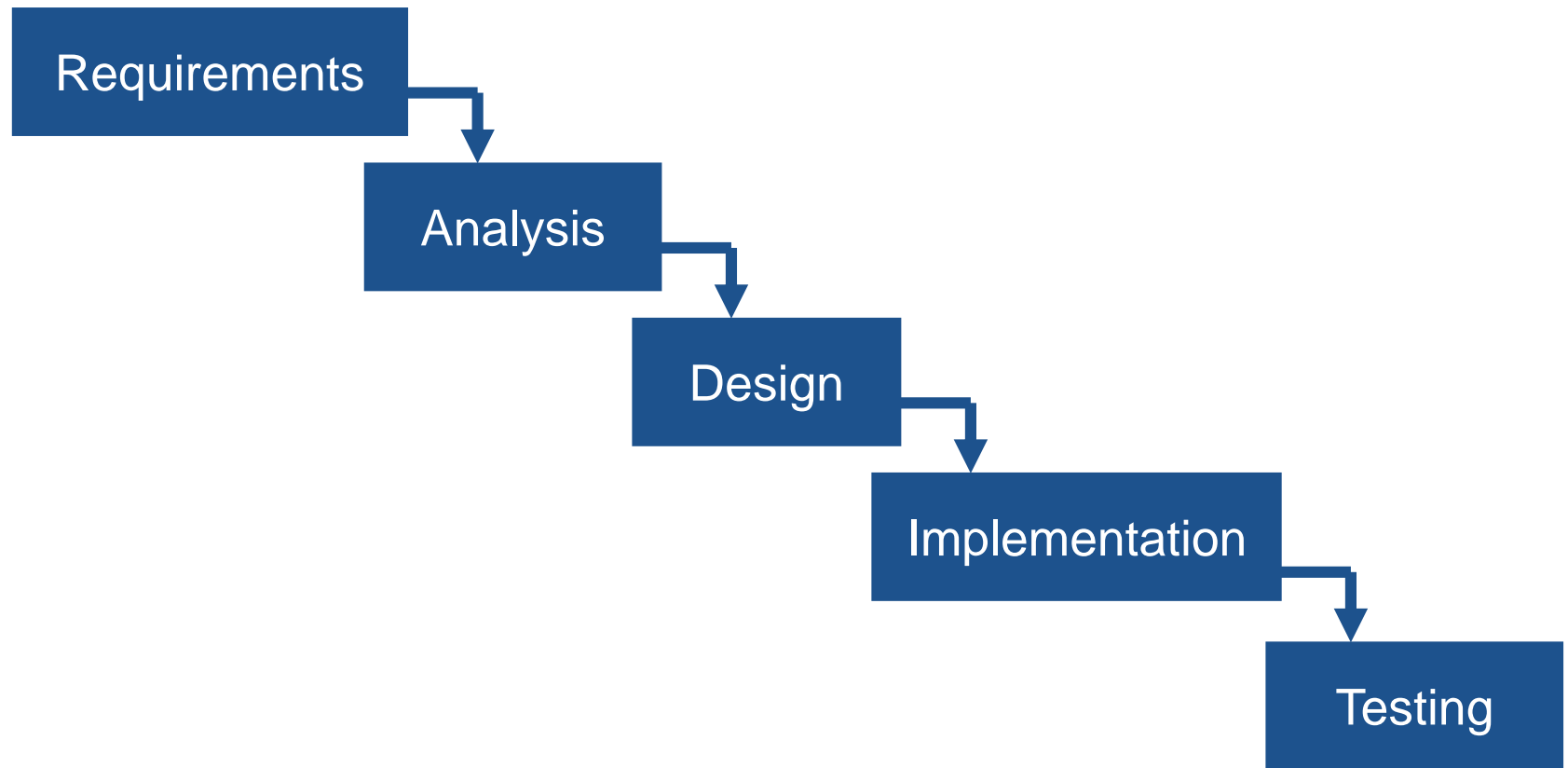
Software development

❖ **The Unified Process**

- The Unified Process (UP) is a methodology proposed by the “three amigos” (Booch, Jacobson, Rumbaugh)
 - These three are also the co-developers of the Unified Modeling Language (UML)
 - The Unified Process is also sometimes called the Unified Software Development Process (USDP), as it is in the textbook (and also RUP – Rational Unified Process)
- The UP introduces a way to develop software where the phases are not distinct, but instead have significant overlap

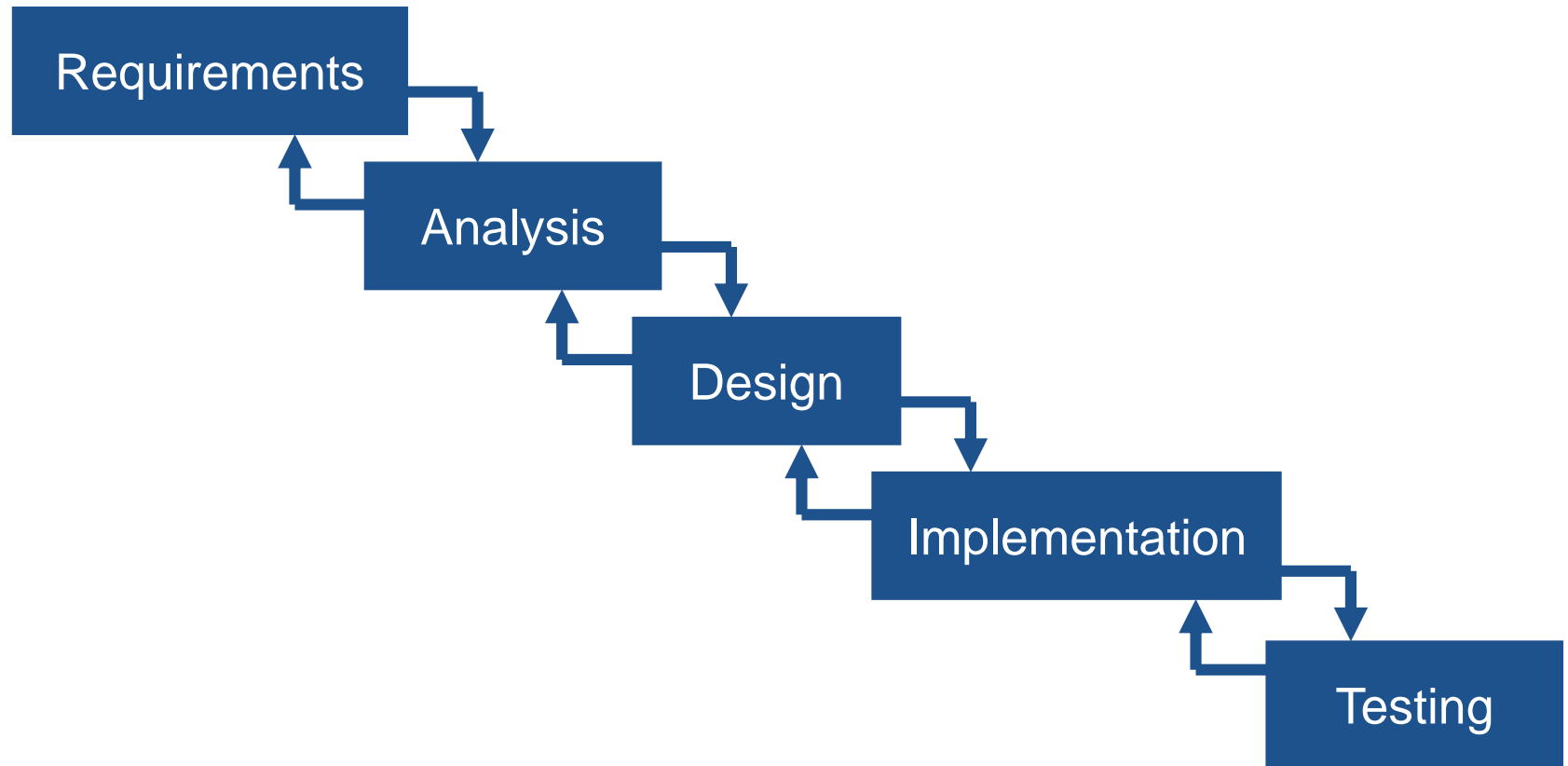
Software development

❖ The waterfall model



Software development

❖ The iterative waterfall model



Basic concepts

❖ **Object-oriented Technology:**

- Object-oriented Analysis (OOA)
- Object-oriented Design (OOD)
- Object-oriented Programming (OOP)

Contents

I.

Software development

II.

Basic concepts of OOAD

III.

OOAD Approaches

Basic concepts (cont.)

❖ **Object-oriented Technology enables reuse:**

- Libraries (i.e through APIs)
- Data structures and Algorithms (low level)
- Components: connected modules
- Architectures: designs (i.e structured components)

Basic concepts (cont.)

- **Analysis** emphasizes an **investigation** of the problems and requirements, rather than a solution.
 - For example, if a new computerized library information system is desired, how will it be used?
- **Design** emphasizes a **conceptual solution** that fulfills the requirements, rather than its implementation.
 - For example, a description of a database schema and software objects. Ultimately, designs can be implemented.

Basic concepts (cont.)

❖ **Analysis and design have been summarized in the phase:**

- Do the right thing (analysis)
- and do the thing right (design).

Basic concepts (cont.)

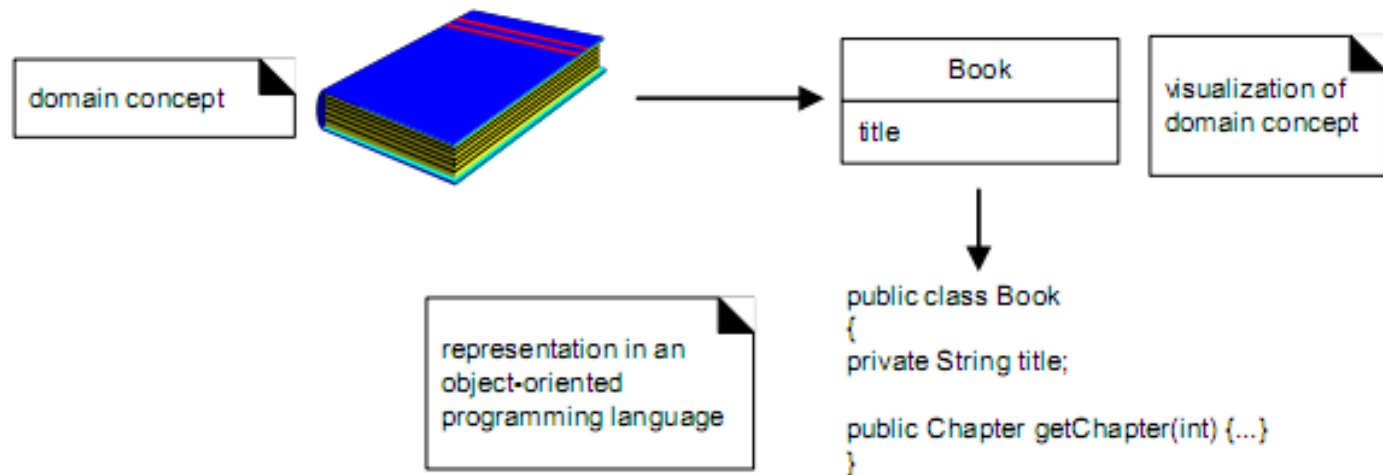
❖ **Object-Oriented Analysis**

- Emphasis on **finding** and **describing** the objects—or concepts—in the problem domain
- For example, in the case of the library information system, some of the concepts include Book, Library, and Reader.

Basic concepts (cont.)

❖ Object-Oriented Design

- Emphasis on defining **software objects** and how they **collaborate** to fulfill the requirements
- For example, in the library system, a **Book** software object may have a **title** attribute and a **getChapter** method.



Basic concepts (cont.)

❖ What is OOAD?

- Object-Oriented = Objects + Communication + Classification + Inheritance

❖ **Both the world of the client, and the world of the developer are modelled in terms of OBJECTS.**

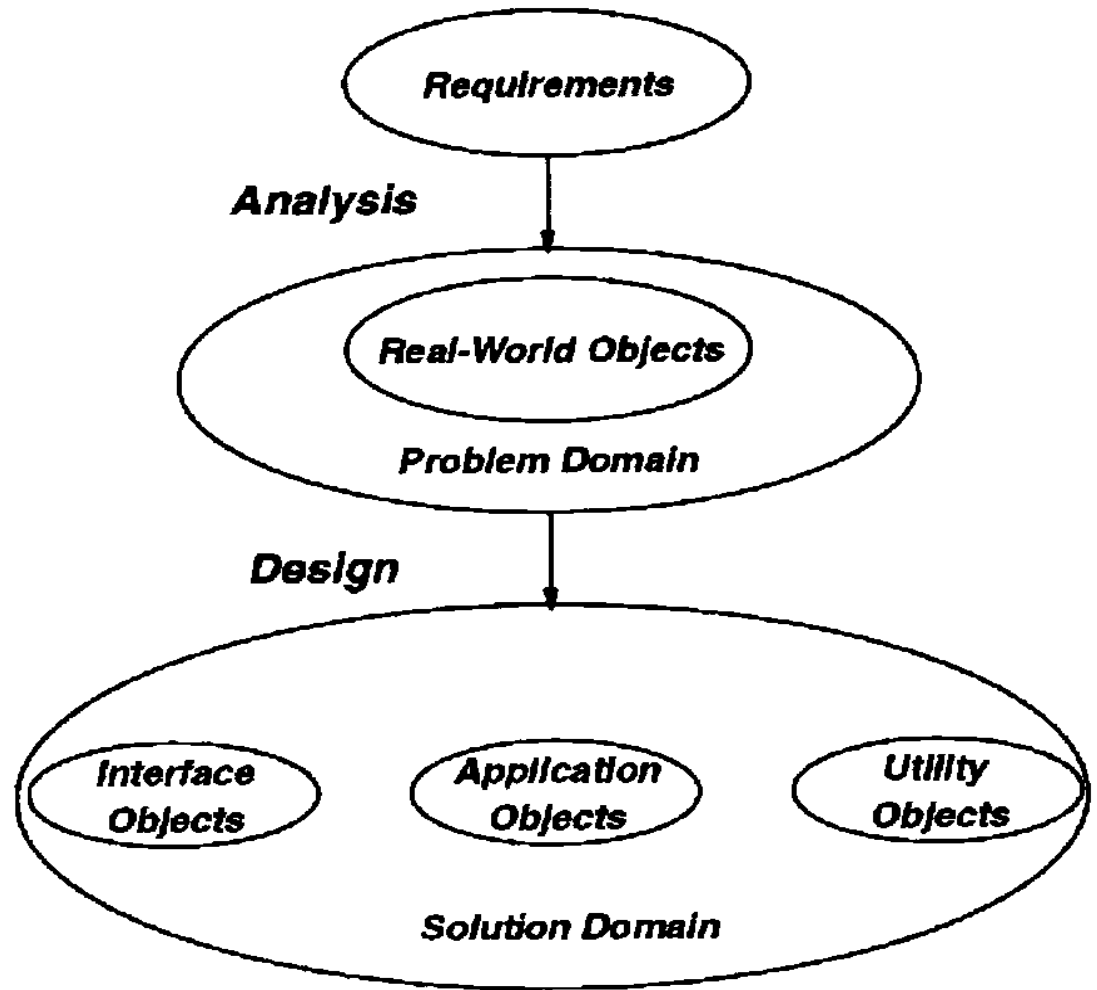
Basic concepts (cont.)

❖ **Object-Orientation Benefits:**

- Problem domain is modelled in terms of real-world objects.
- Abstraction, encapsulation, and generalization help managing complexity and achieving reuse.
- No real distinction between analysis and design terminology & tools.

Basic concepts (cont.)

❖ OOAD Model:



Contents

I.

Software development

II.

Basic concepts of OOAD

III.

OOAD Approaches

OOAD Approaches

- ❖ **Booch**
- ❖ **Coad & Yourdon**
- ❖ **Fusion**
- ❖ **Jacobson Objectory & OOSE**
- ❖ **LBMS SEOO**
- ❖ **Rumbaugh OMT**
- ❖ **Shlaer and Mellor OO Analysis**
- ❖ **UML & RUP**

OOAD Approaches (cont.)

❖ **Booch**

- Grady Booch, *Object-Oriented Design with Applications*, Benjamin/Cummings, 1994.
- Strong in design, although also covers requirements and domain analysis.
- Booch's method is ONE of 3 major contributors to UML.

OOAD Approaches (cont.)

❖ Coad & Yourdon

- Coad & Yourdon, *Object-Oriented Analysis and Object-Oriented Design*, Prentice-Hall, 1990 and 1991.
- Focuses on analysis of business problems, and uses a friendlier notation.

OOAD Approaches (cont.)

❖ Coad & Yourdon

- Consists of 5 stages in analysis (SOSAS):
 - Subjects
 - Objects
 - Structures
 - Attributes
 - Services

OOAD Approaches (cont.)

❖ **Coad & Yourdon**

- Consists of 4 components in design:
 - Problem domain
 - Human interaction
 - Task management
 - Data management

OOAD Approaches (cont.)

❖ **Fusion**

- Developed in 1990 by Derek Coleman of HP.
- Incorporate some major ideas from Booch, Jacobson, Rumbaugh, and others.
- Not widely marketed, thus not popularly used.

OOAD Approaches (cont.)

❖ **Fusion**

- Developed in 1990 by Derek Coleman of HP.
- Incorporate some major ideas from Booch, Jacobson, Rumbaugh, and others.
- Not widely marketed, thus not popularly used.

OOAD Approaches (cont.)

❖ **Jacobson Objectory & OOSE**

- Objectory is proprietary.
- OOSE is a simplified version of Objectory.
- Use Object-oriented concepts, plus the “use case”
- Use cases and actors are used to model the system requirements.
- The SECOND of 3 major contributors to UML.

OOAD Approaches (cont.)

❖ **LBMS SEOO**

- Systems Engineering Object-Oriented.
- Proprietary to LBMS – a U.K company.
- Tightly integrated with PowerBuilder.
- Consists of 4 major components:
 - Work-breakdown structures and techniques
 - An object modeling methodology
 - GUI design techniques
 - Relational database linkages

OOAD Approaches (cont.)

❖ **Rumbaugh OMT**

- James Rumbaugh, *Object-Oriented Modeling and Design*, Prentice-Hall, 1991.
- Analysis consists of 3 models:
 - Object model
 - Dynamic model
 - Functional model
- The **THIRD** of 3 major contributors to UML.

OOAD Approaches (cont.)

❖ **Shlaer and Mellor OO Analysis**

- First came in 1988.
- Shlaer and Mellor, *Object-Oriented Systems Analysis -- Modeling the World in Data and Object Lifecycles: Modeling the World in States*, Prentice-Hall, 1988 and 1992.
- Considered an object-based extension of data modeling.

OOAD Approaches (cont.)

❖ **UML & RUP**

- Unified Modeling Language
- Administred by OMG – Object Management Group.
- First came in 1997 (version 1.1)
- The latest version is UML 2.3 (05/2010)
- Combine “best of the best” from many different methodologies.

OOAD Approaches (cont.)

❖ **UML & RUP**

- UNIFIED Modeling Language
- OMG adopted UML in November 1997 as the standard for object-oriented modeling
- Combines commonly accepted concepts from many OO methods
- Seamless from requirements to deployment
- Applicable to any domain
- Language and platform independent
- Usable with any development process

OOAD Approaches (cont.)

❖ **UML & RUP**

- Unified MODELING Language
- All engineering disciplines have adopted modeling techniques
- Allow to capture the important aspects of a system while omitting the rest
- Help all software systems stakeholders understand what the system will be and what are the possible options available
- Multiple models reveal different perspectives on the system

OOAD Approaches (cont.)

❖ UML & RUP

- Unified Modeling LANGUAGE: used for
 - **Visualizing**: Graphical models with precise semantics
 - **Specifying**: Models are precise, unambiguous and complete to capture all important Analysis, Design, and Implementation decisions.
 - **Constructing**: Models can be directly connected to programming languages, allowing forward and reverse engineering
 - **Documenting**: Diagrams capture all pieces of information collected by development team, allowing to share and communicate the embedded knowledge.