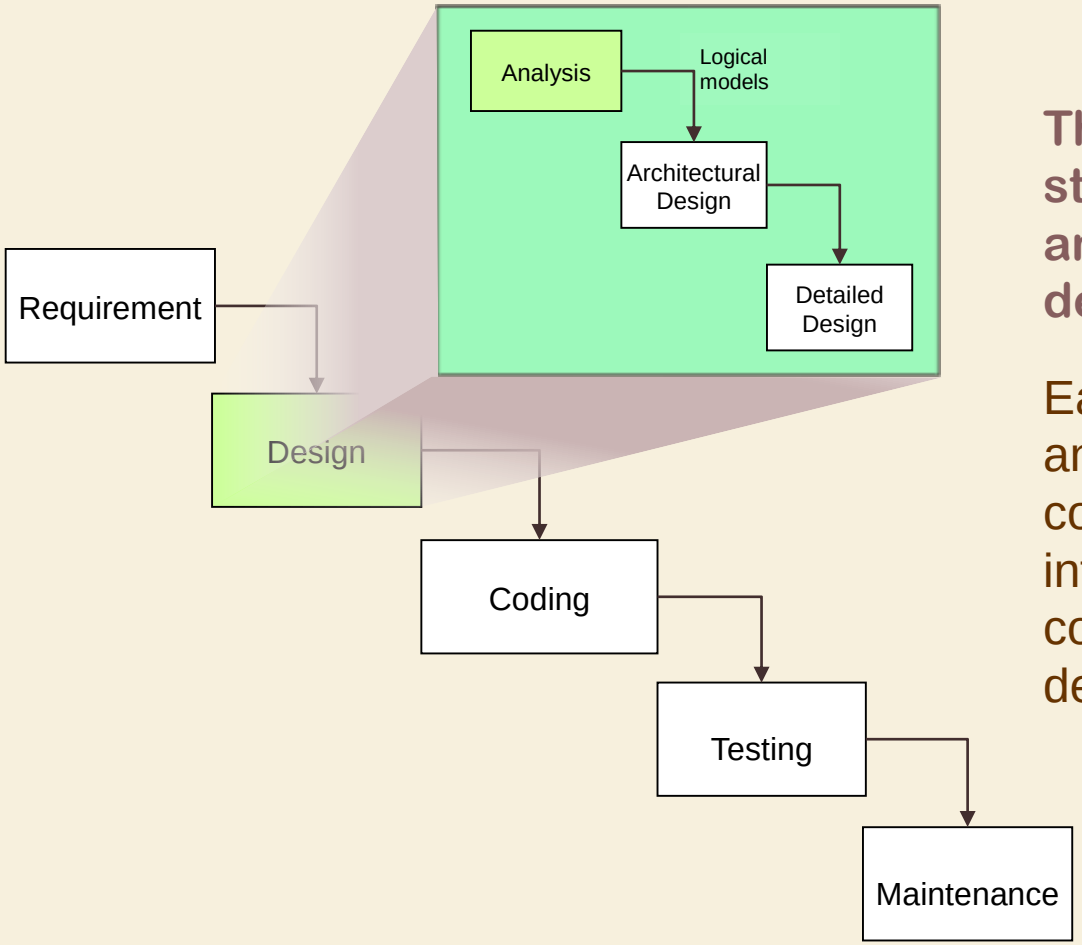
UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Module T03b
Software Engineering

Intro to Analysis & Design

Compiled from multiples sources by Mun'im Zabidi (munim@utm.my)

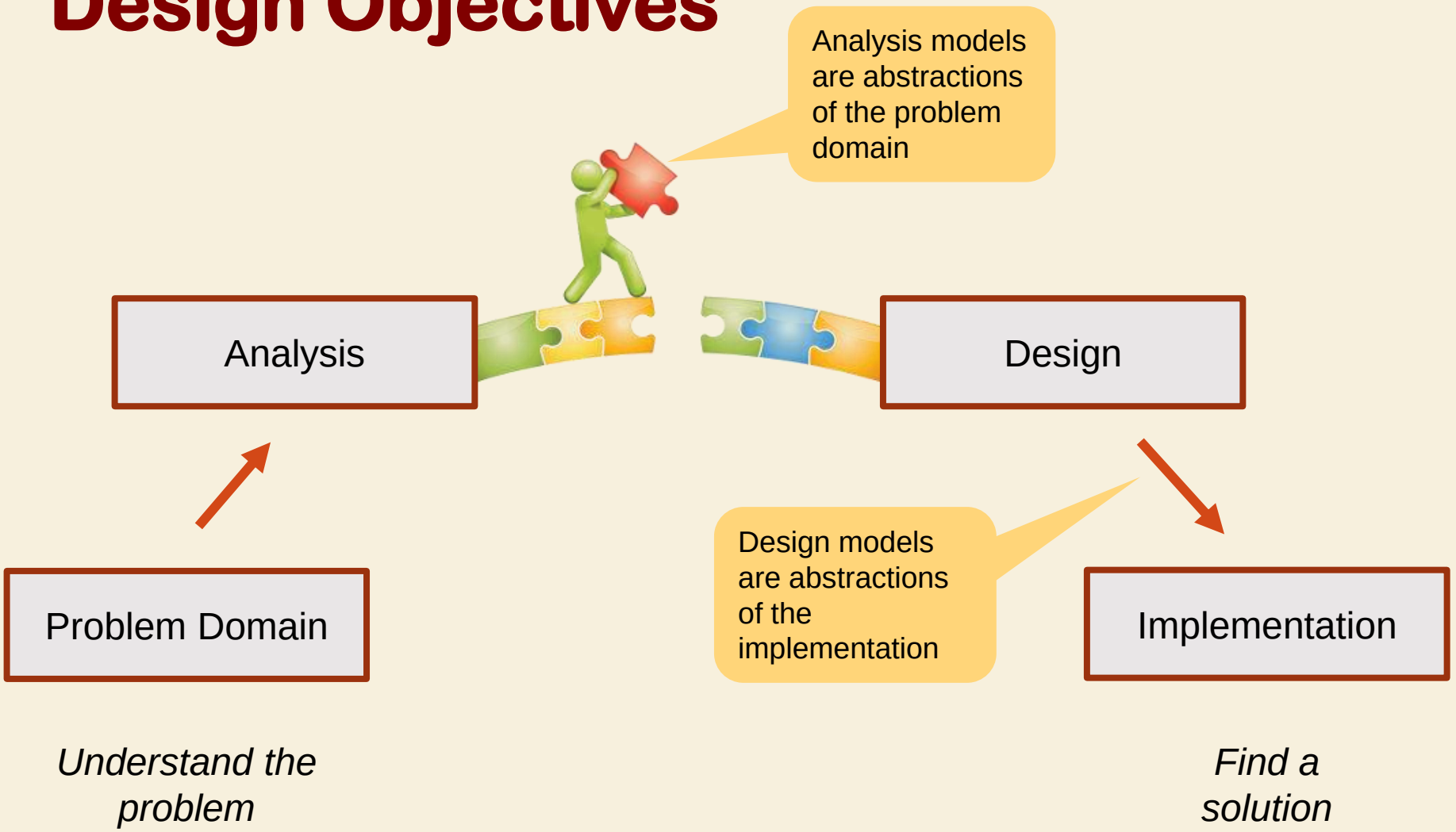
Where Are We?



The requirements are studied & clarified, and overall solution determined.

Each major subsystem is analyzed, and components & interaction between components are determined.

Analysis Objectives versus Design Objectives



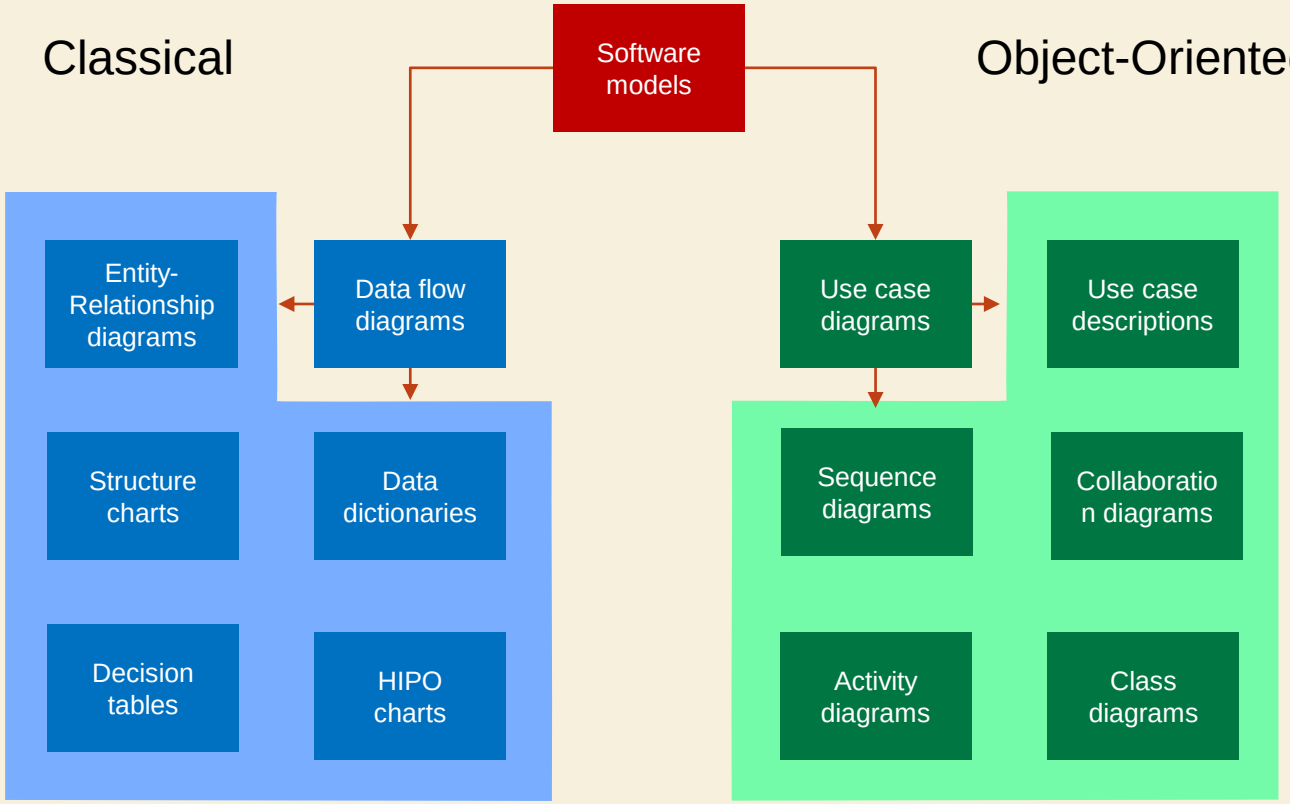
Analysis vs Design

Analysis	Design
Constructing a model of the problem domain	*Constructing a model of the solution domain*
• What is the problem? • What can be improved? • What does the user want? • What are the services and constraints of the system?	• How will the system be designed to meet user requirements? • How will the user interface look like? • How will the database look like?

Two Approaches to System Development

- Classical approach
  - Also called structured system development or traditional approach
  - Structured analysis and design technique (SADT)
  - Includes information engineering (IE)
- Object-oriented approach
  - Also called OOA, OOD, and OOP
  - Views information system as collection of interacting objects that work together to accomplish tasks

Classical vs OO Modeling Tools



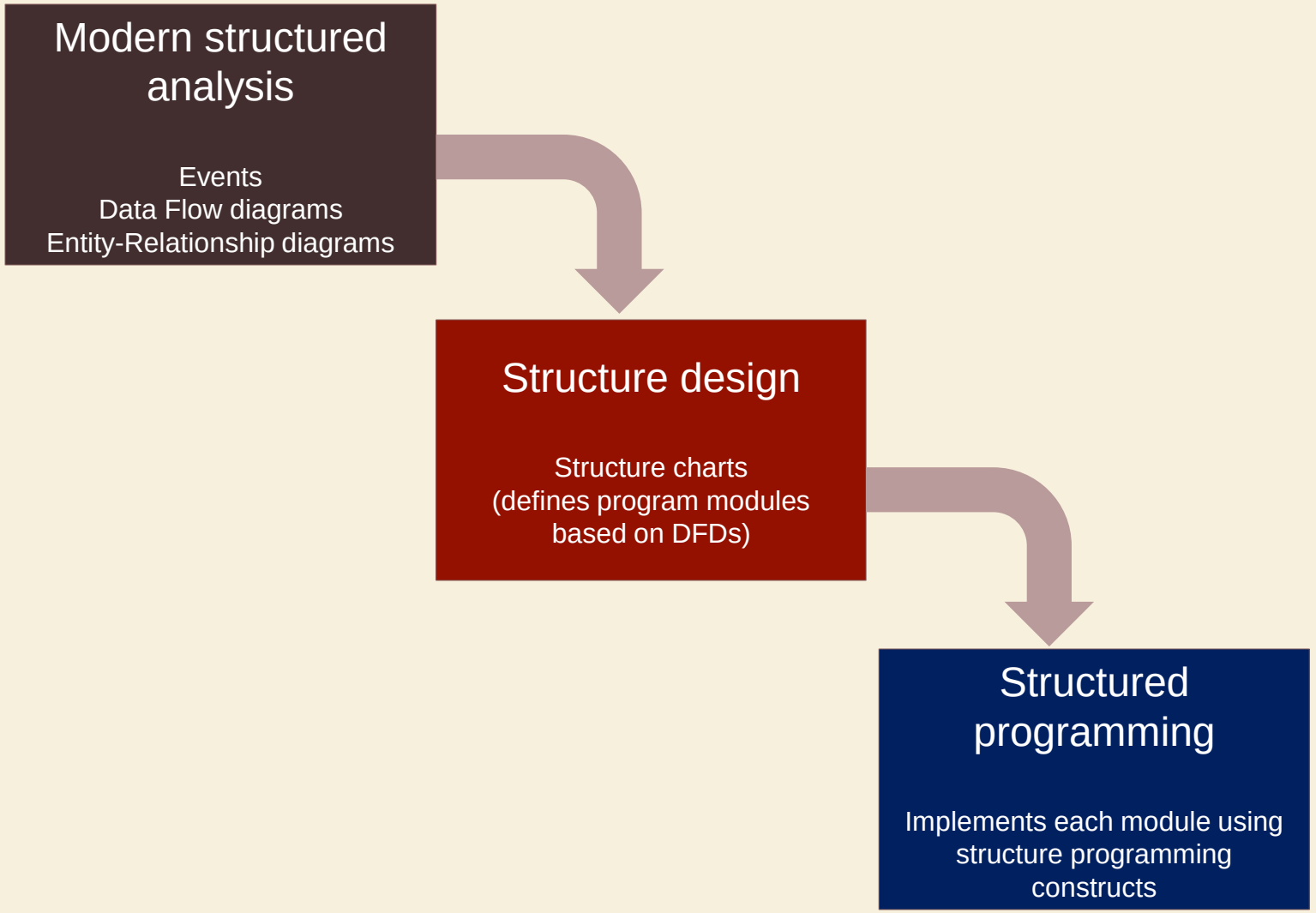
Focus : process modeling.

Focus : data modeling.

Classical vs OO

	Classical/SSAD	Object-Oriented
Methodology	Waterfall	Agile
Focus	Process	Objects
Risk	High	Low
Reuse	Low	High
Maturity	Mature & widespread	Emerging
Suitability	Well-defined projects with stable requirements	Risky projects with changing requirements
Tools	Data Flow Diagrams (DFD), Entity-Relationship Diagram (ERD)	UML (Use Cases, Activity Diagrams)

Traditional Approach



Modeling

- Models are mechanisms for communication
- Each model shows only a small amount of the totality of information about the artefact; too much information in a model makes it difficult to comprehend
- A model is an **abstraction**, which allows people to concentrate on the essentials of a (complex) problem by keeping out non-essential details.

Models: Logical and Physical

Model –

a pictorial representation of reality.

Just as a picture is worth a thousand words, most models are pictorial representations of reality.

Logical model –

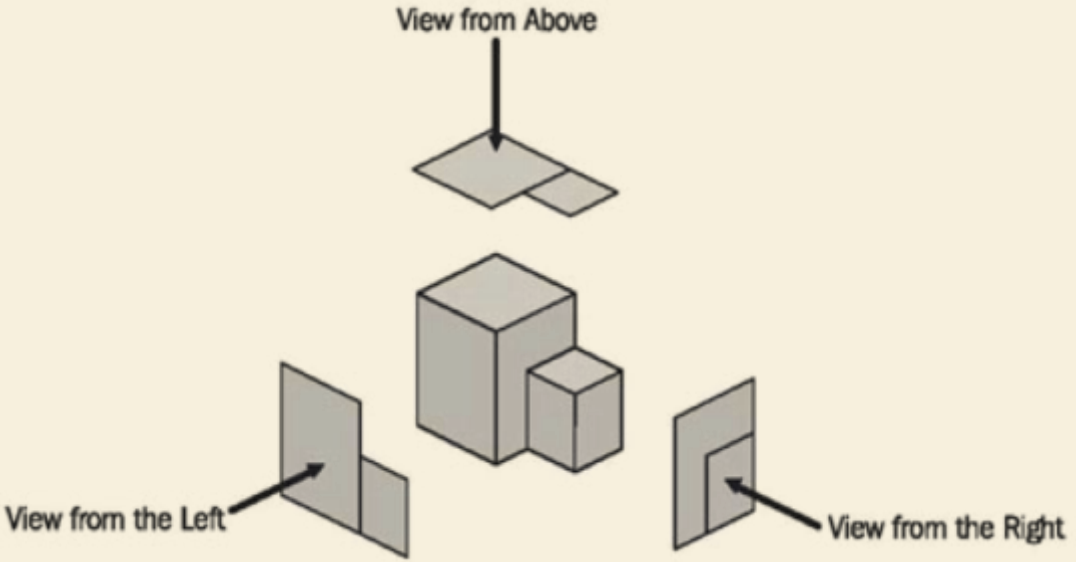
a nontechnical pictorial representation that depicts what a system is or does. Synonyms or essential model, conceptual model, and business model.

Physical model –

a technical pictorial representation that depicts what a system is or does and how the system is implemented. Synonyms are implementation model and technical model.

Different Views

- Each modeling technique provides a different view of a system and a complete understanding of a system cannot be obtained without a number of different views
- Eg. A building has:
  - architecture view
  - electrical view
  - plumbing view

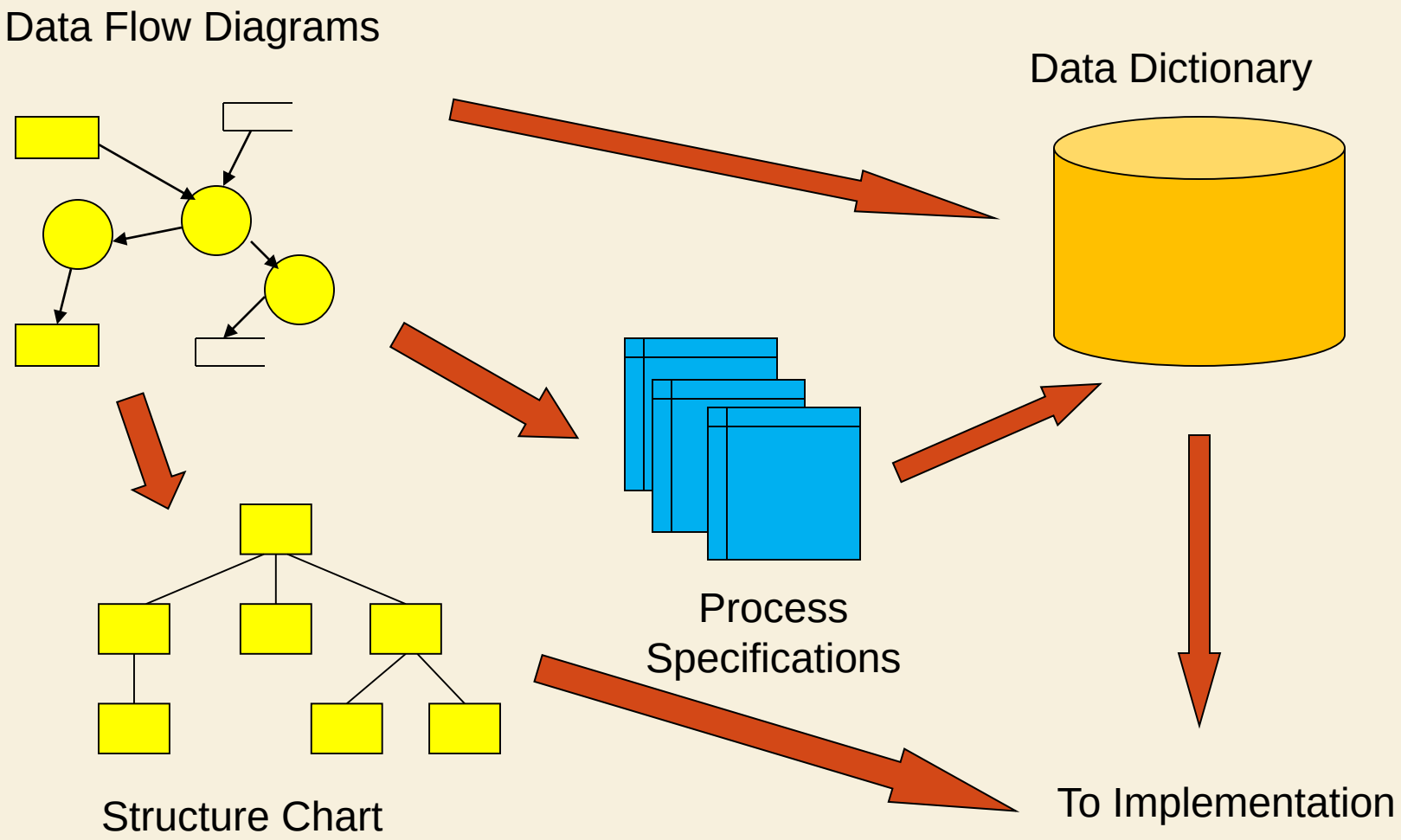


SSADM

SSADM

- Structured systems analysis and design method (SSADM), originally released as methodology, is a systems approach to the analysis and design of information systems
- widely-used computer application development method in the UK
- British Standard BS7738.
- SSADM is a waterfall method
- a pinnacle of the rigorous document-led approach to system design, and contrasts with more contemporary agile methods such as DSDM or Scrum

Documentation In SSAD



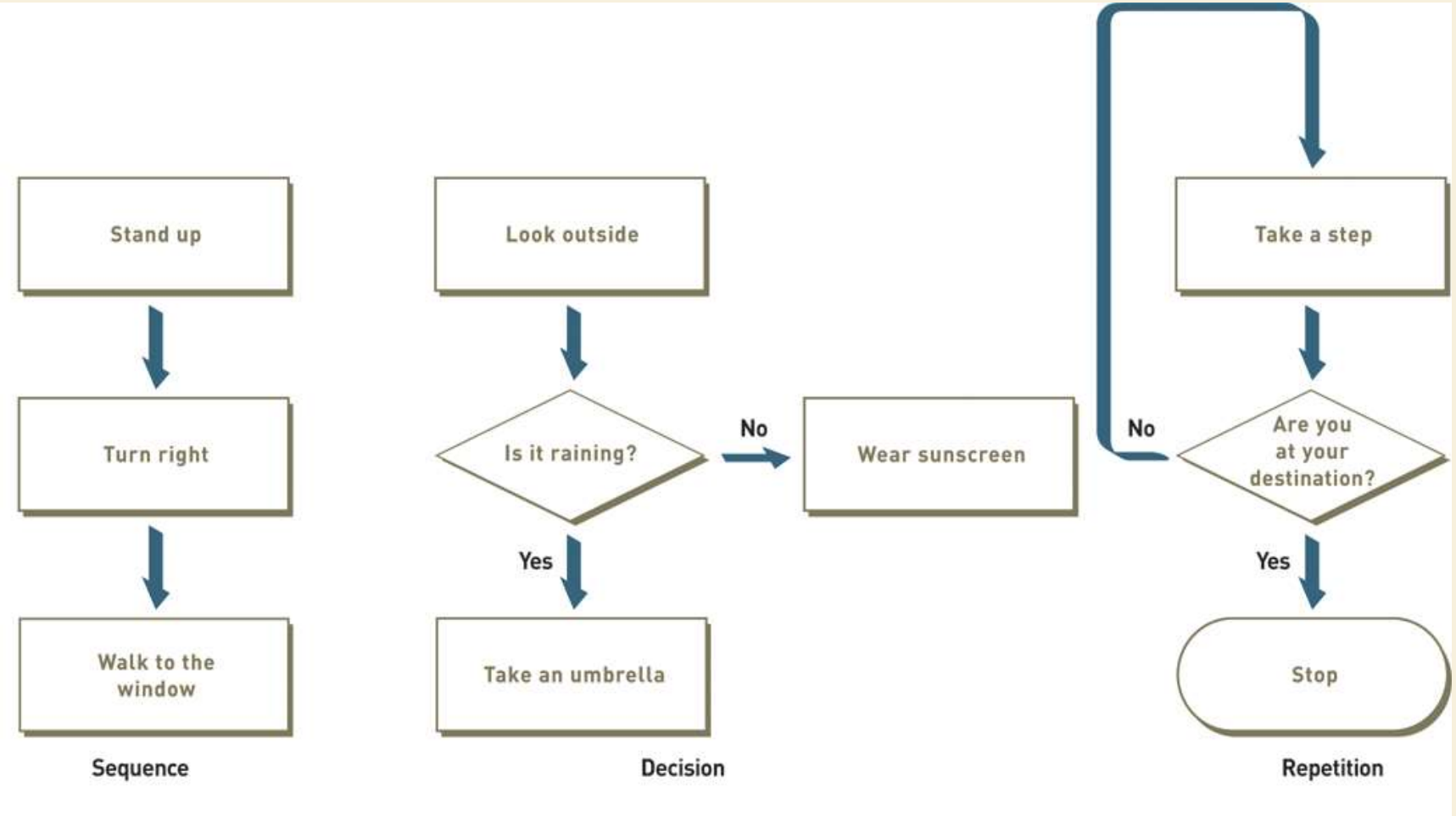
Structured Analysis

- Define what system needs to do (processing requirements)
- Define data system needs to store and use (data requirements)
- Define inputs and outputs
- Define how functions work together to accomplish tasks
- Data flow diagrams (DFD) and entity relationship diagrams (ERD) show results of structured analysis

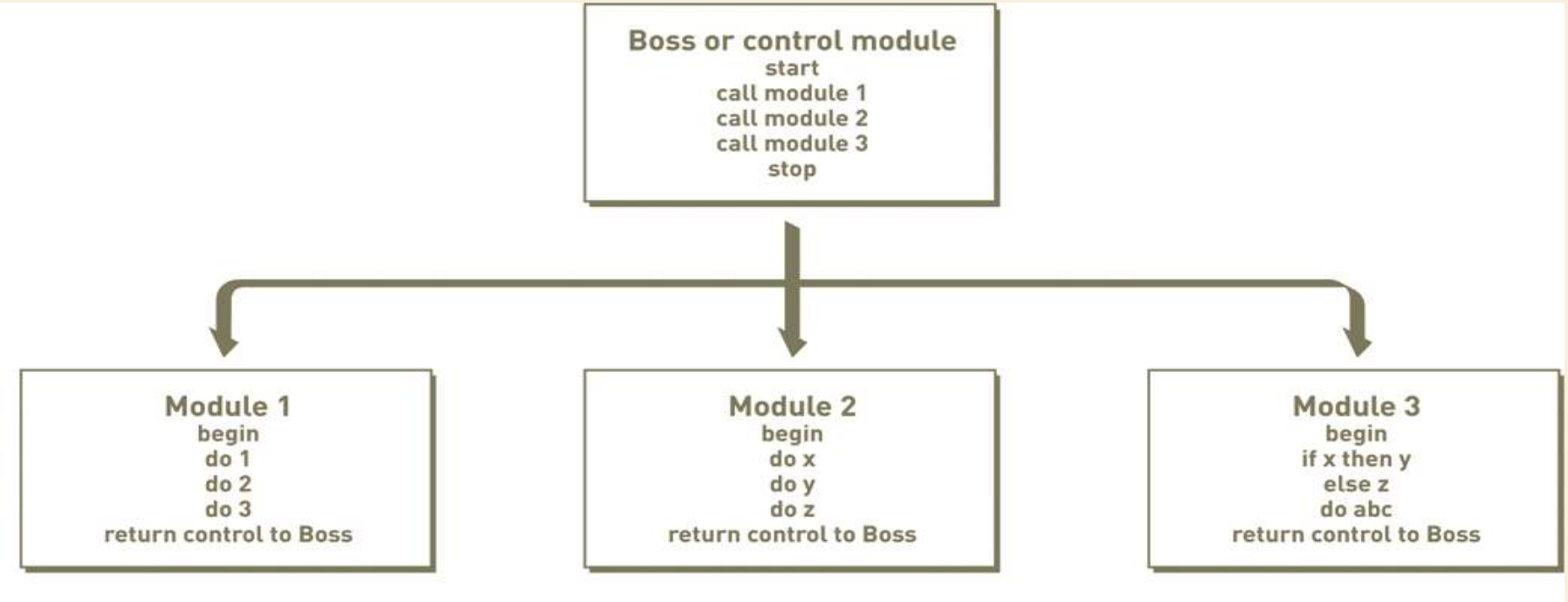
Structured Design

- Technique developed to provide design guidelines
  - What set of programs should be
  - What program should accomplish
  - How programs should be organized into a hierarchy
- Modules are shown with structure chart
- Main principle of program modules
  - Loosely coupled – module is independent of other modules
  - Highly cohesive – module has one clear task

Three Structured Programming Constructs



Top-Down or Modular Programming



OO Approach

Object-Oriented Approach

- Views information system as collection of interacting objects that work together to accomplish tasks
  - Objects – things in computer system that can respond to messages
  - Conceptually, no processes, programs, data entities, or files are defined – just objects
- Data, Processes and Interface focuses integrated into a single focus in objects

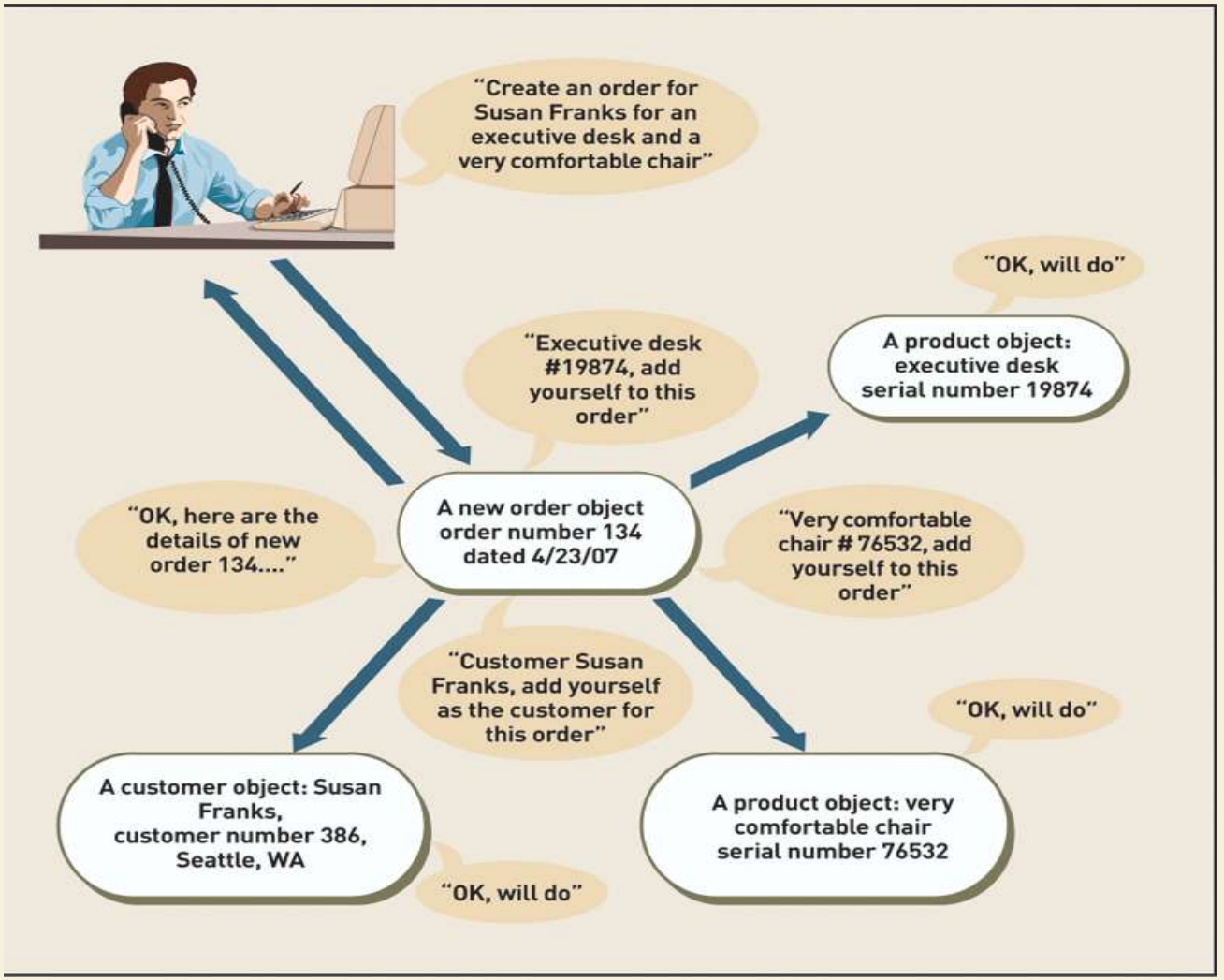
- OO languages: Java, C++, C# .NET, VB .NET, Python

System Concepts for Object Modeling

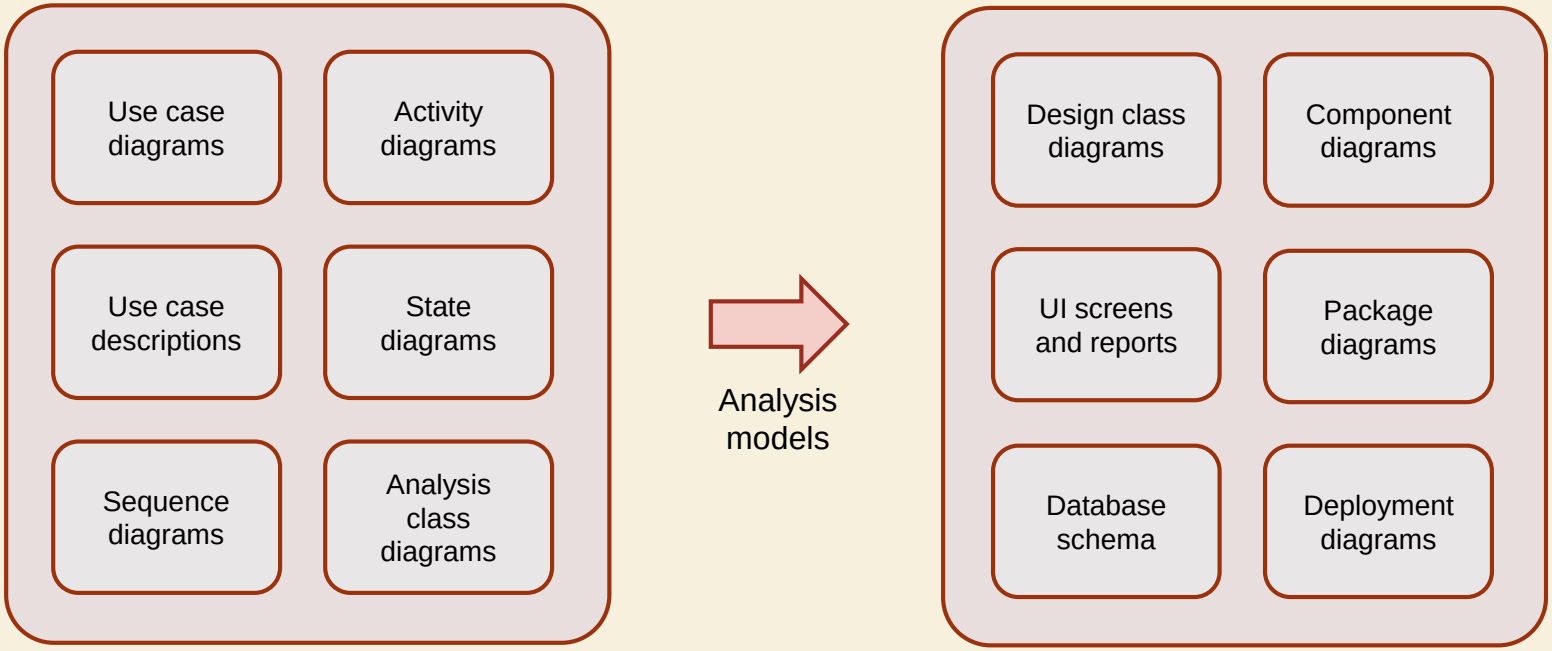
- Object: something that is or is capable of being seen, touched, or otherwise sensed and about which users store data and associate behavior

Something	Data	Instance	Behavior
• Types of Objects: • person, places, thing or events. • Employee, customer, vendor or students -- person objects • Warehouse, regional office, building and room -- place objects • Product, vehicle, equipment -- thing objects • Order, payment, invoice, registration -- event objects	• Attributes: data that represents characteristics of interest about an object. • Customer Number, First and Last Name, Address, Home phone, Type of Customer, Credit Limit, Credit, Account Balance, Account Status.	Object Instance) values of the attributes that describe a specific person, place, thing, or event. E.g: 123456, Lonnie, Mently, 2626, Darwin Drive, West Lafayette, Indianana, 47096 New attributes: picture, sound, video	(: those things that an object can do correspond to functions that act on the objects data. Objects behavior: method, operation, service e.g: door, open, lock, shut, unlock. Phone object: answer, dial, hang up.

Object-Oriented Approach to Systems



Analysis vs. Design Models

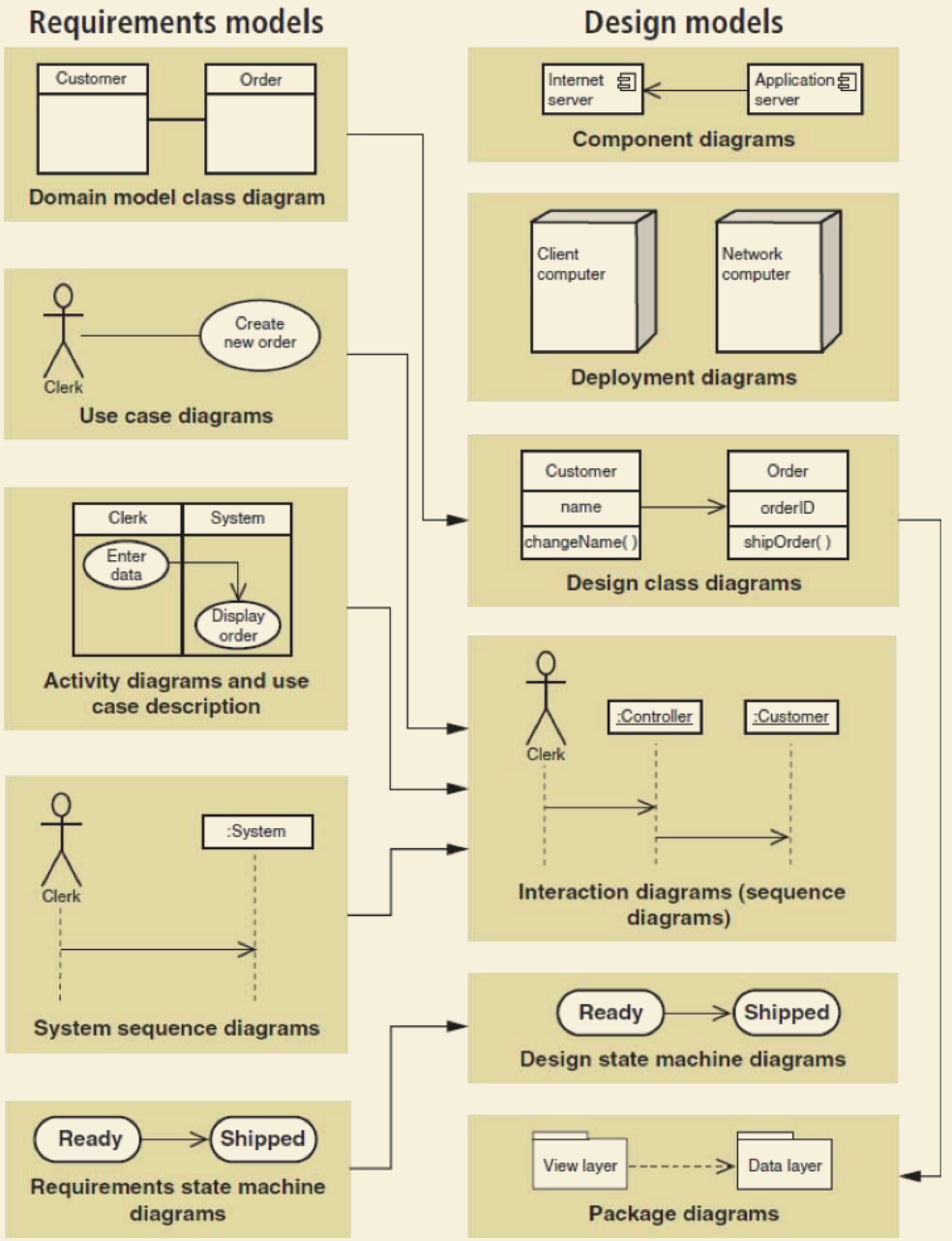


Investigation of the problem and requirements

Description of software solution

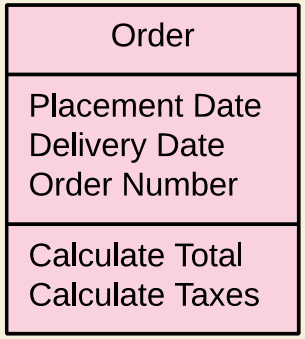
UML Requirements vs. Design Models

Diagrams are enhanced and extended

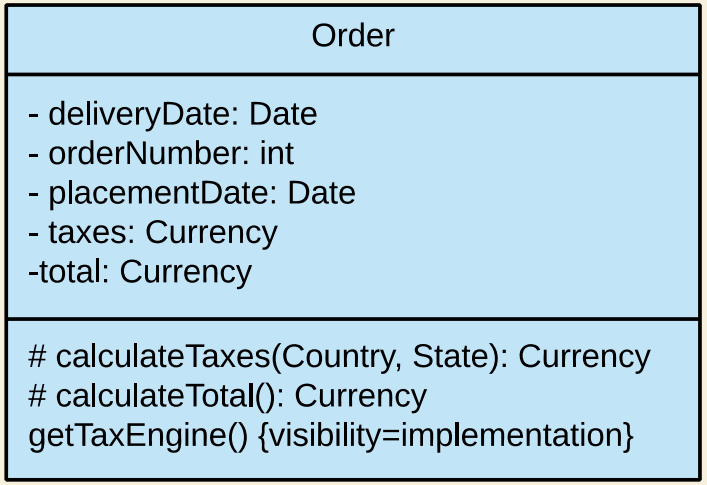


Analysis vs Design Class Icons

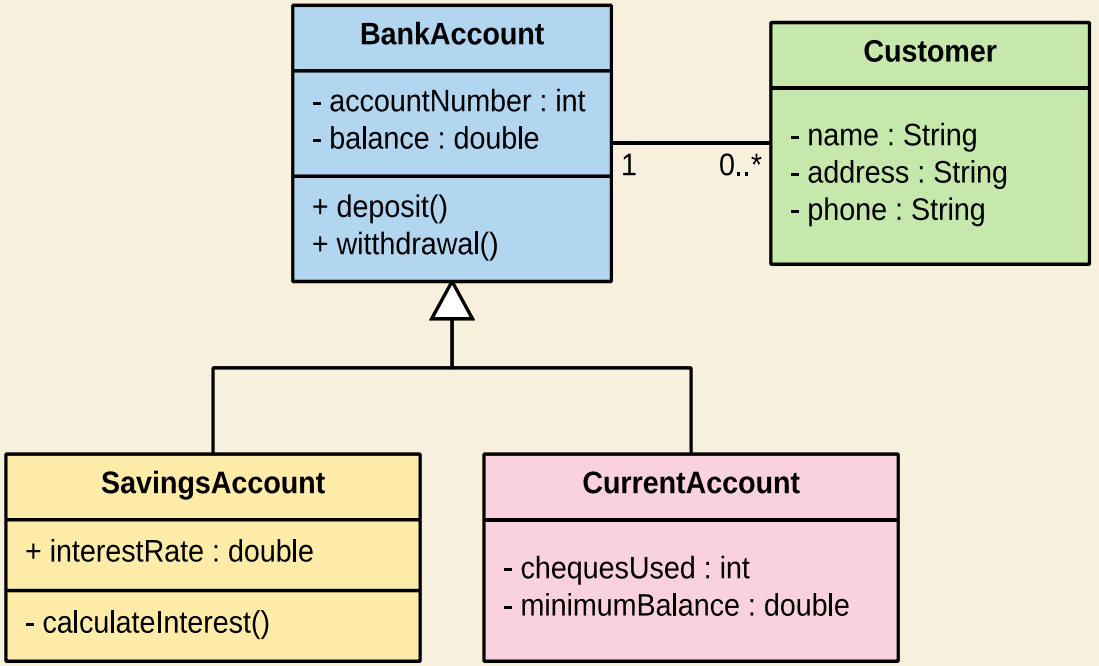
Analysis



Design



Class Diagram Created During OO Analysis



Object-Oriented Approach

- Object-oriented analysis (OOA)
  - Defines types of objects users deal with
  - Shows use cases are required to complete tasks
- Object-oriented design (OOD)
  - Defines object types needed to communicate with people and devices in system
  - Shows how objects interact to complete tasks
  - Refines each type of object for implementation with specific language of environment
- Object-oriented programming (OOP)
  - Writing statements in programming language to define what each type of object does

Basic OO Modeling Steps

- Use Cases
  - Capture requirements
- Domain Model
  - Capture process, key classes
- Design Model
  - Capture details and behaviors of use cases and domain objects
  - Add classes that do the work and define the architecture

Domain Analysis

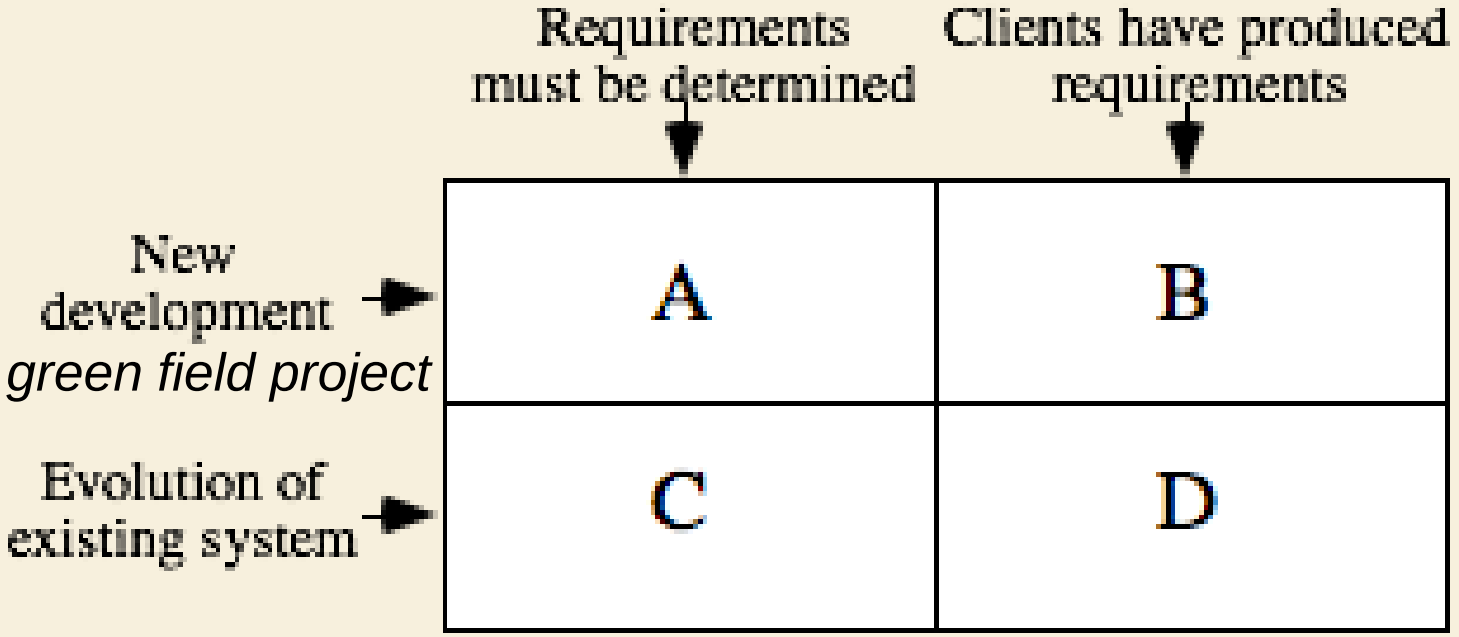
Domain Analysis

- The process by which a software engineer learns about the domain to better understand the problem:
  - The domain is the general field of business or technology in which the clients will use the software
  - A domain expert is a person who has a deep knowledge of the domain
- Benefits of performing domain analysis:
  - Faster development
  - Better system
  - Anticipation of extensions

Domain Analysis document

- A. Introduction
- B. Glossary
- C. General knowledge about the domain
- D. Customers and users
- E. The environment
- F. Tasks and procedures currently performed
- G. Competing software
- H. Similarities to other domains

The Starting Point for Software Projects



Defining the Problem and the Scope

- A problem can be expressed as:
  - A difficulty the users or customers are facing,
  - Or as an opportunity that will result in some benefit such as improved productivity or sales.
- The solution to the problem normally will entail developing software
- A good problem statement is short and succinct

Defining the Scope

- Narrow the scope by defining a more precise problem
  - List all the things you might imagine the system doing
    - Exclude some of these things if too broad
    - Determine high-level goals if too narrow

- Example: A university registration system