Chapter 8
Approaches to System Development

Dr. Supakit Nootyaskool
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang

Chapter outlines

- The system development life cycle (SDLC)
- The support phase
- Methodologies, Models, Tools, and Techniques
- Two approaches to software construction and modeling
- Agile development

Objectives

- Compare
  - The underlying assumption and uses of a predictive and an adaptive system development life cycle
- Describe
  - The key activities and tasks of information system support
  - The two overall approaches used for software construction and modeling
  - The key features of Agile development
- Explain
  - The system development methodology—the SDLC, tools, and techniques

Opening case: Development approaches

- Kim, Mary, and Bob
  - Interview visit CIS on their campus
- Kim
  - Impressed Ajax corp.
  - Data modeling
  - Ajax
    - Developed "IM"
    - SDLC, iteration, business event, DFD, ERD,
- Mary
  - Programmer who known, OMG, UML, UP,
  - Use OOP on Java, C#, .NET
- Bob
  - Interview pinnacle corp.
  - Agile development, Agile modeling technique
- They
  - Work on many different way of key concept and techniques that they learned in school.

8.1 SDLC concept

- Many approaches to developing system
- Categorize development approaches:

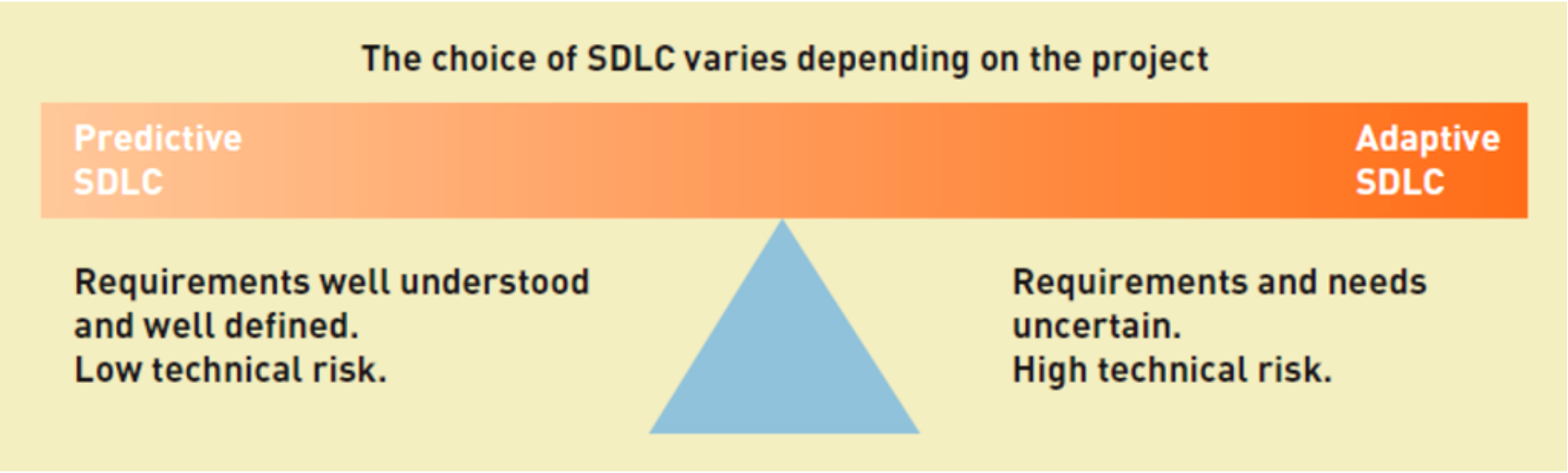
Predictive approach

- The project can be planned in advance and that the new information system can be developed according to the plan.

Adaptive approach

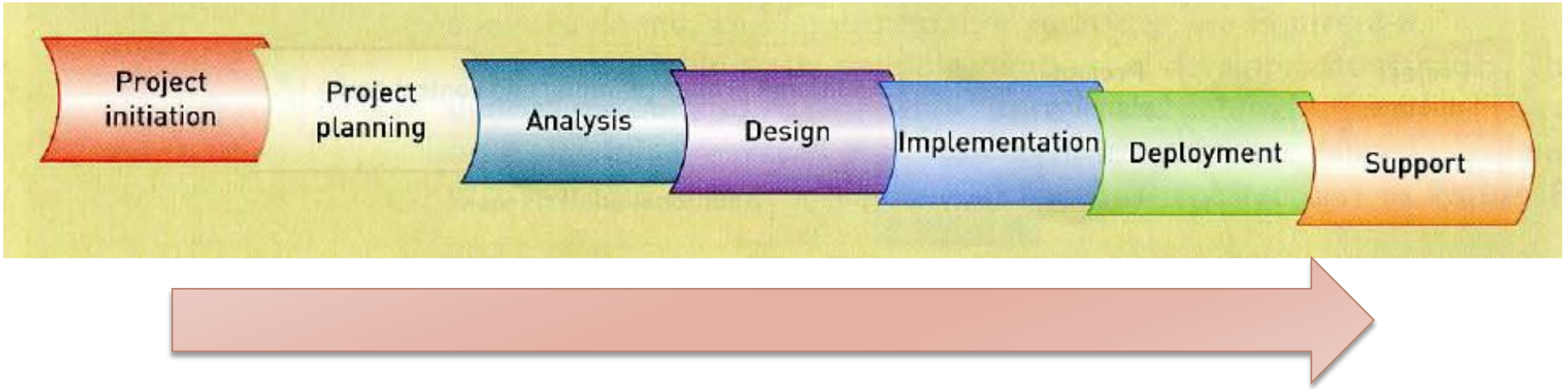
- The project must be more flexible and adapt to changing needs as the project progresses.

Predictive VS Adaptive approach to SDLC



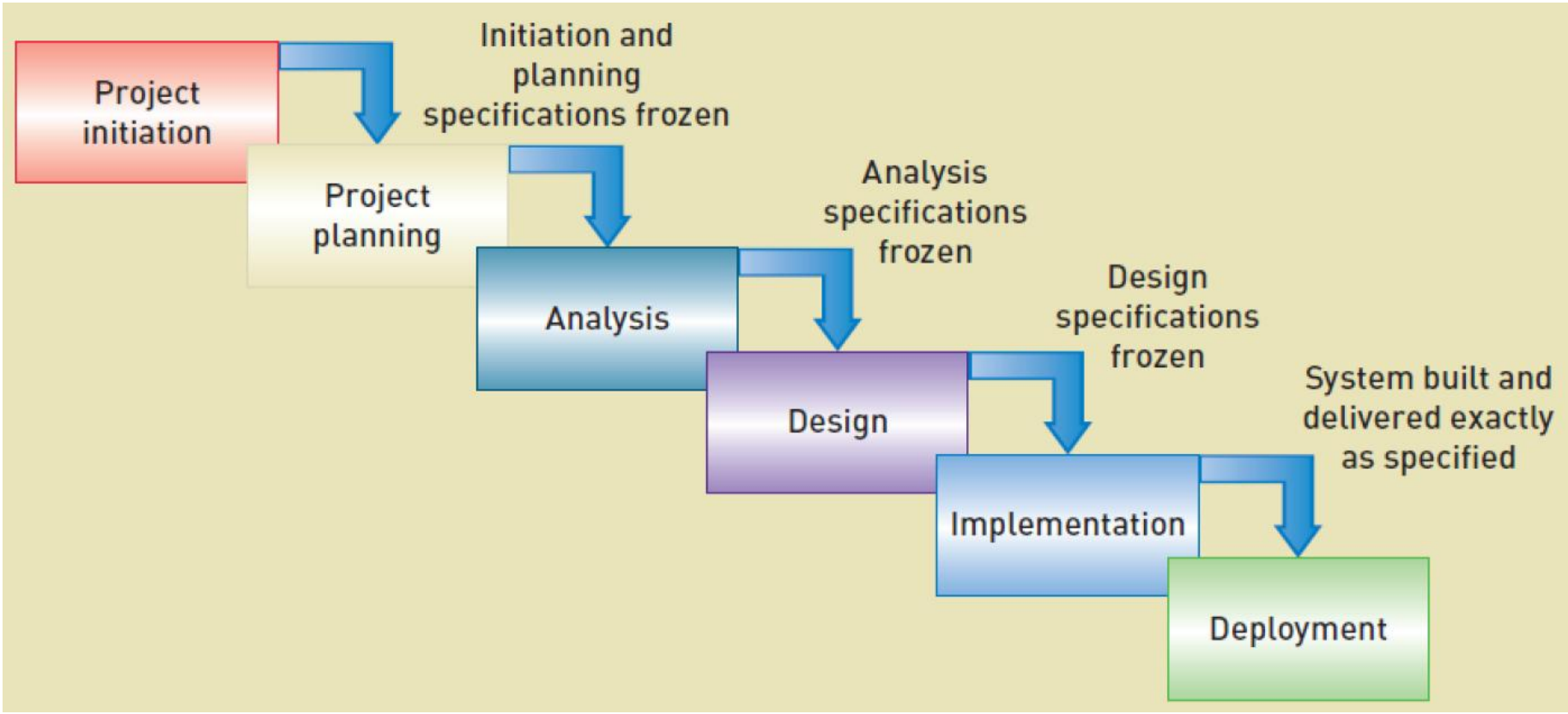
Traditional Predictive approach to the SDLC:

- Phases of the system development project



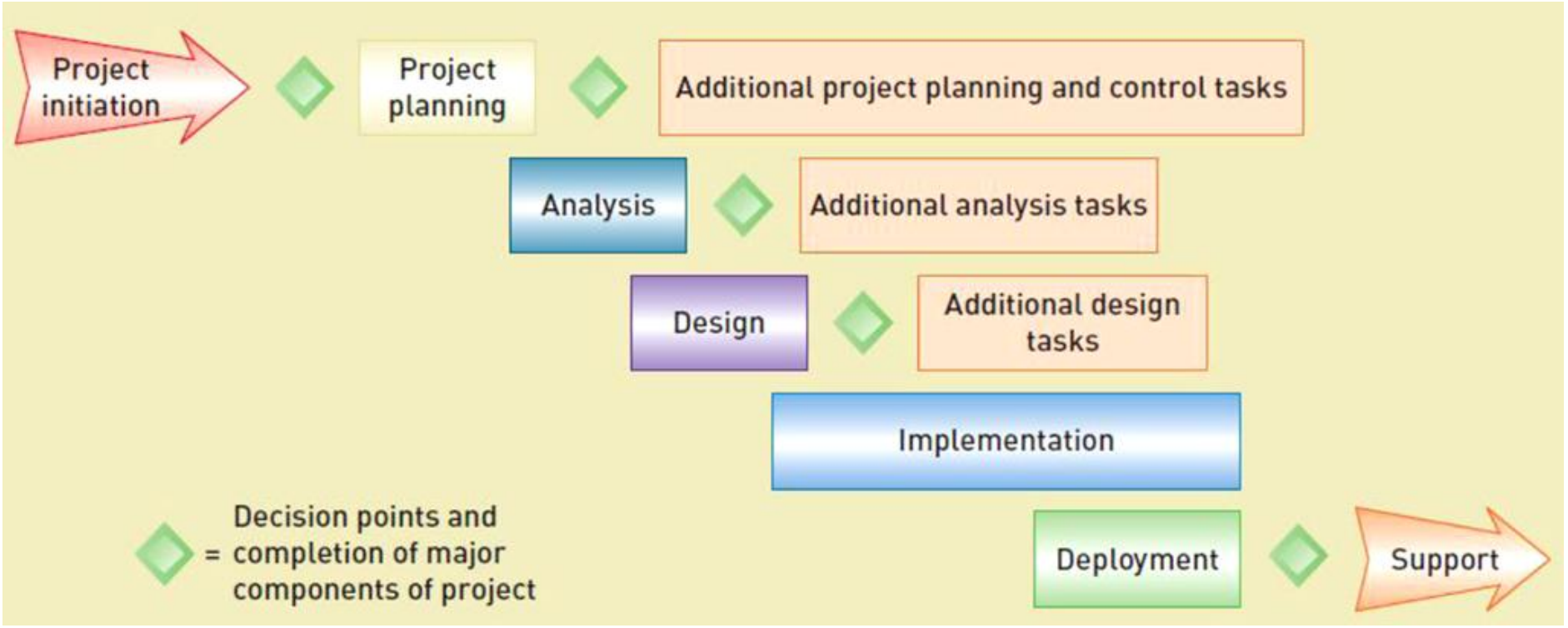
Traditional Predictive approach to the SDLC:

- Waterfall model: the phase can be completed sequentially with no overlap (NO GOING BACK).



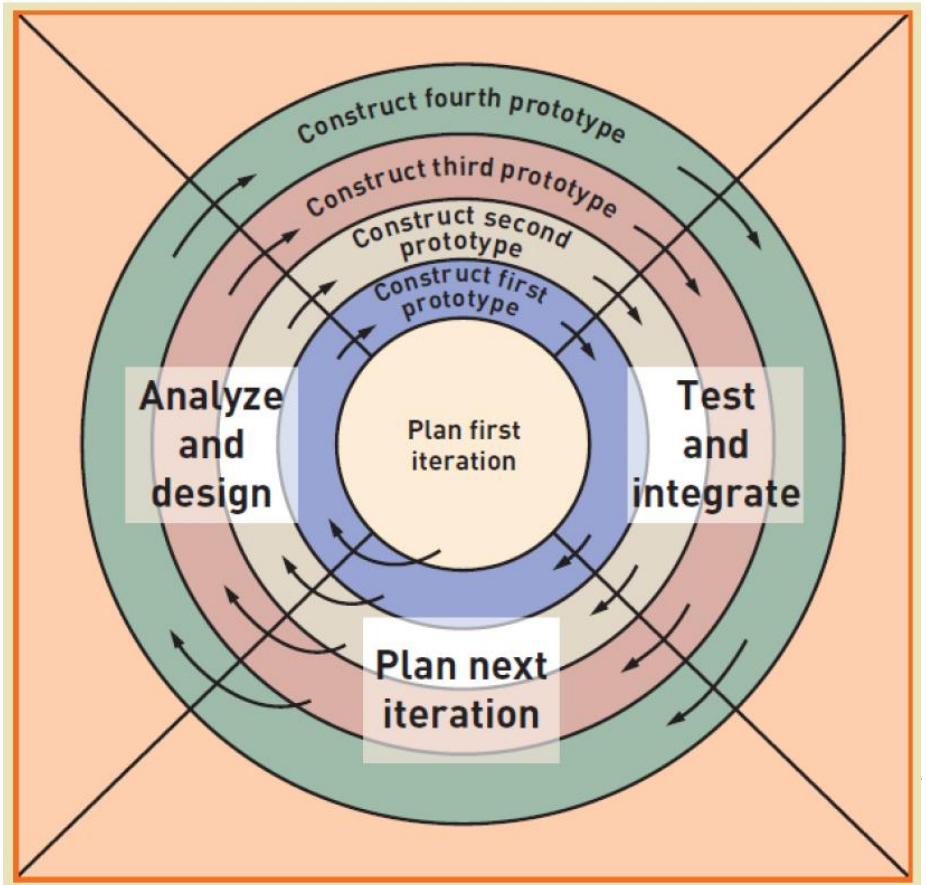
Traditional Predictive approach to the SDLC:

- The Modified Waterfall model: project must be overlap

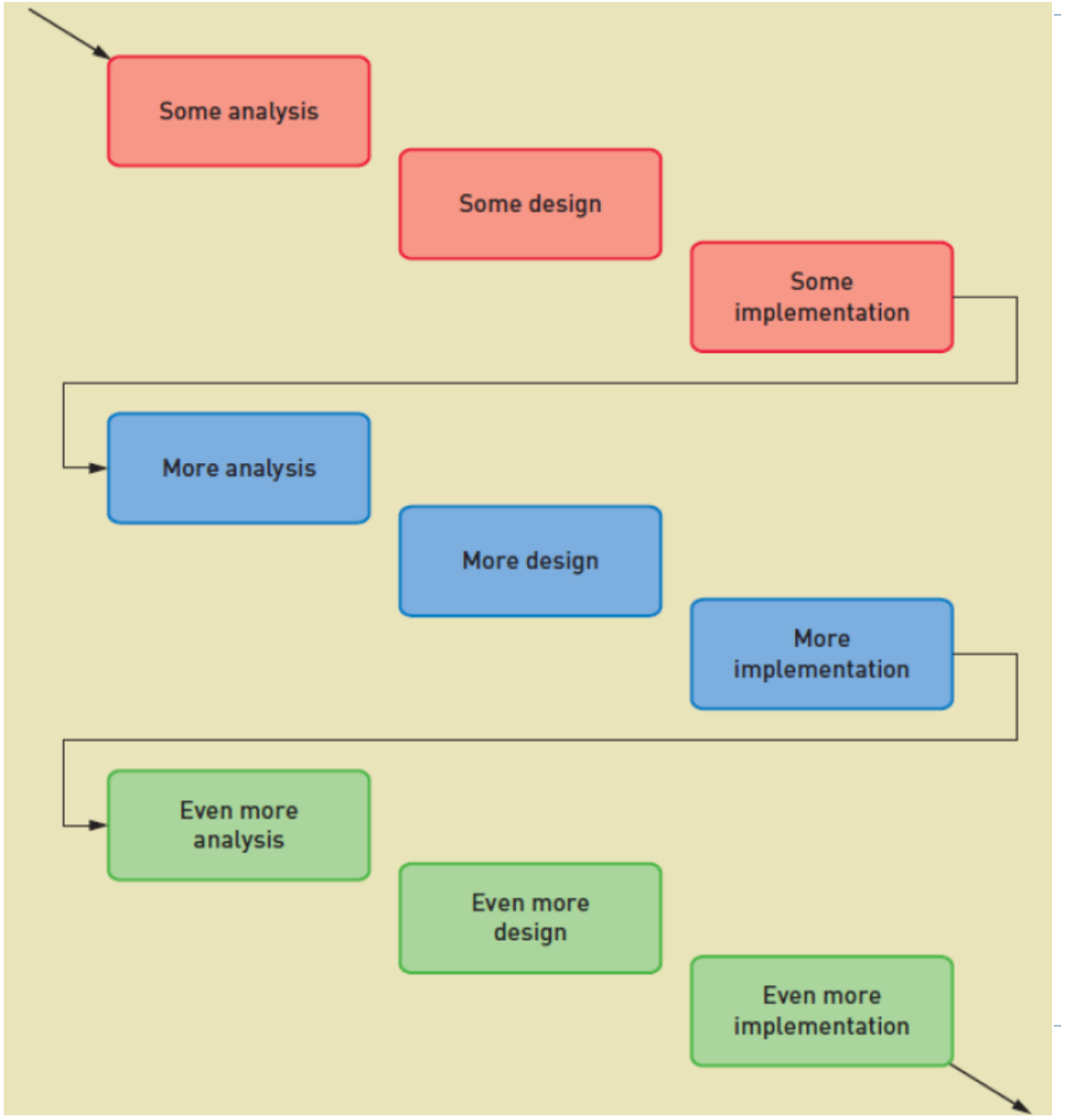


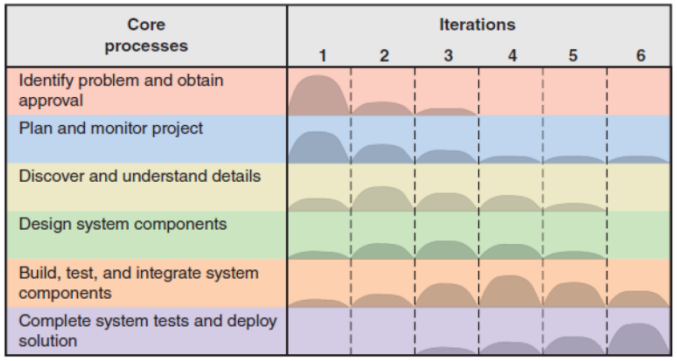
Newer Adaptive approaches to the SDLC

- "Iteration" or "Mini-project" consists with design, build, and test
- Spiral model: It contains many iteration cycles developing repeatedly.



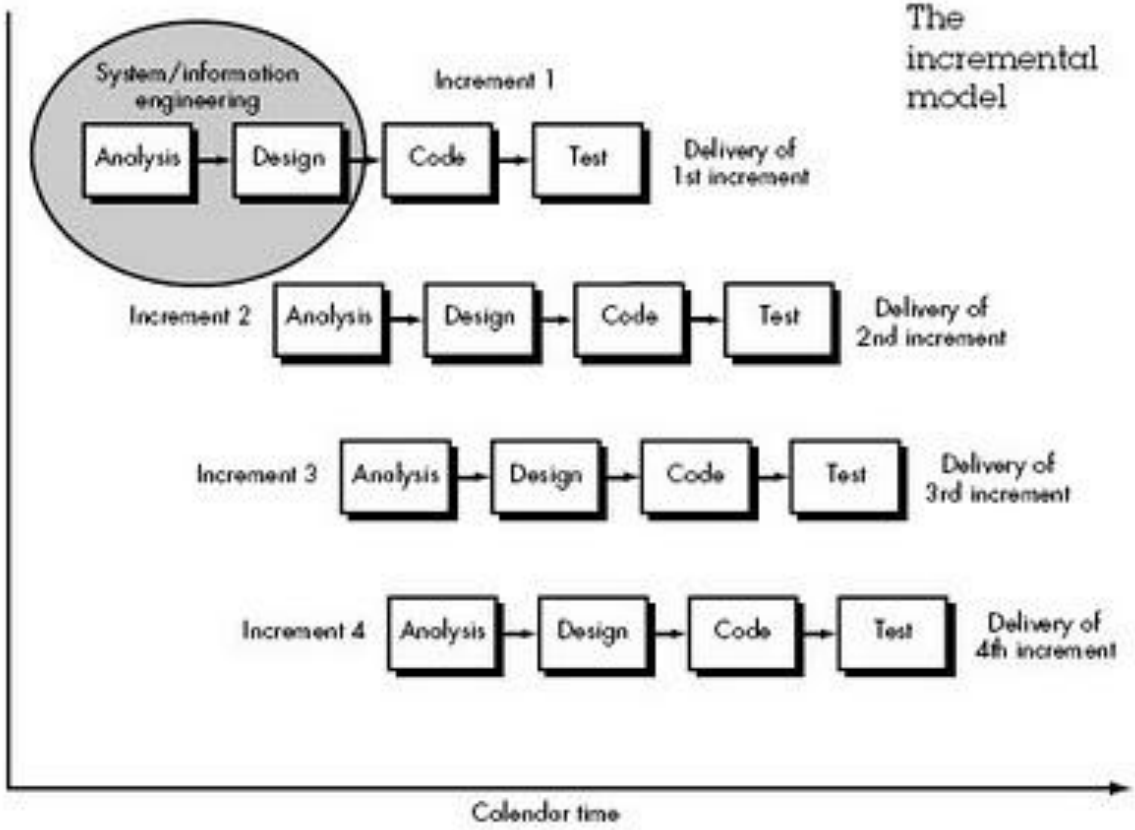
“Iteration model”





Newer Adaptive approaches to the SDLC

- Incremental development is an iterative life cycle.
  - The basic idea is that the system is build in small increments.



Newer Adaptive approaches to the SDLC

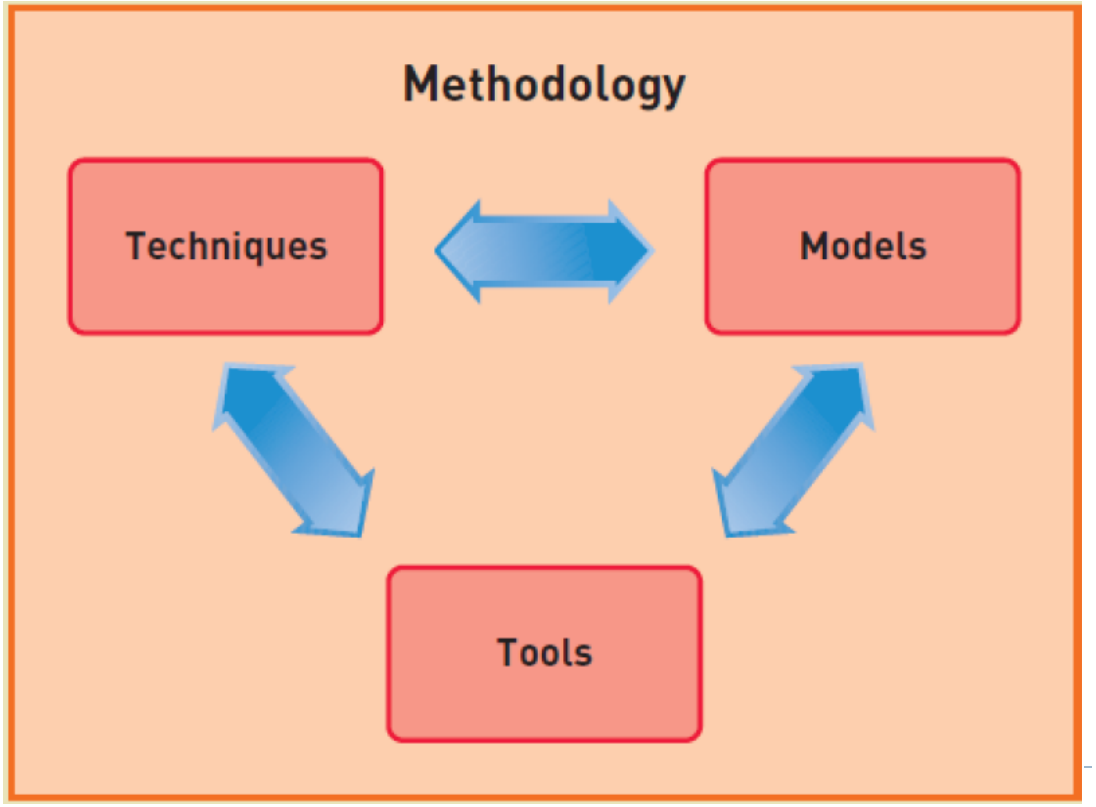
- Walking skeleton provides a complete font-to-back implementation of the new system
  - A walking skeleton is different from a spike. A spike is "the smallest implementation that demonstrates plausible technical success." The spike typically takes between a few hours and a few days to complete, and is thrown away afterward, since it was built with nonproduction coding habits.
  - A walking skeleton, on the other hand, is permanent code, built with production coding habits, regression tests, and is intended to grow with the system. Once the system is up and running, it will stay up and running for the rest of the project, despite the Incremental Re-architecture that is quite likely to occur.

8.2 The support phase

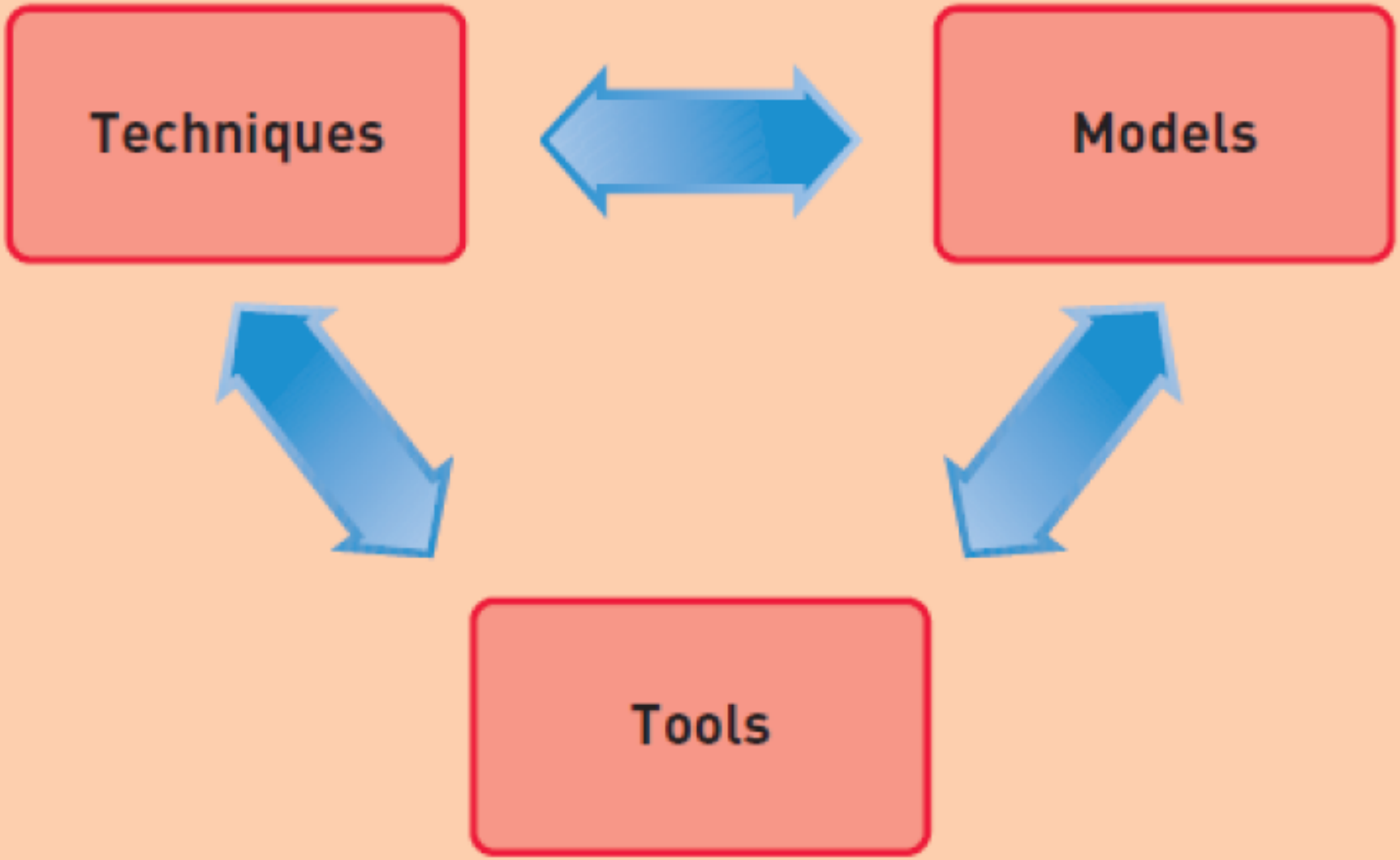
- Support activities: the objective is to maintain and enhance the system after it is installed and in use.
  - Maintaining the system
    - Fix problems and errors
    - Make minor adjustments
    - Update patches
  - Enhancing the system
    - Add desired functionality
    - Add or change functionality to comply with regulations or legislation
  - Supporting user
    - Outgoing user training
    - Help desk:

8.3 Methodologies

- System development methodology provides guideline for every fact of the system development life cycle.
- Methodology contains:



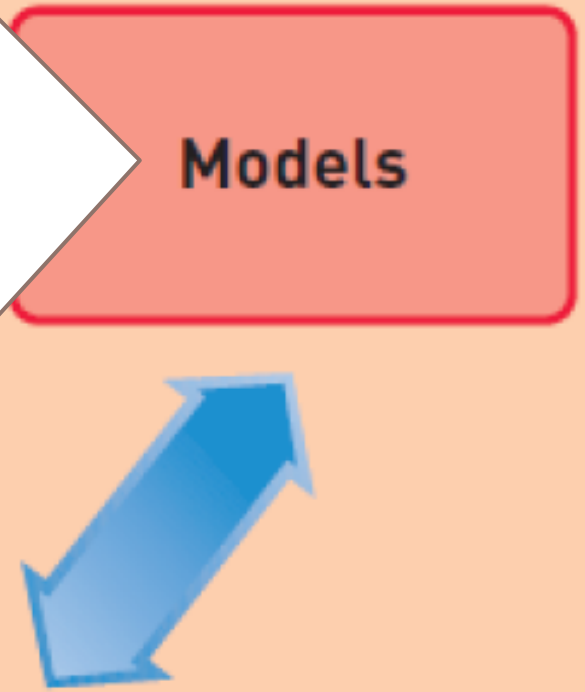
Methodology



- Make to understand a complex concept
- Use for communication project team

- Some models of system components
  - Flowchart
  - Data flow diagram (DFD)
  - Entity-relationship diagram (ERD)
  - Structure chart
  - Use case diagram
  - Class diagram
  - Sequence diagram

- Some models used to manage the development process
  - Gantt chart
  - Organization hierarchy chart
  - Financial analysis models



Techniques

• A collection of guidelines that help an analyst complete activity or task.

• Learning techniques
  - Strategic planning techniques
  - Project management techniques
  - Use interviewing techniques
  - Data-modeling techniques
  - Relational database design techniques
  - Structured programming technique
  - Software-testing techniques
  - Domain modeling techniques
  - Use case modeling techniques
  - Object-oriented programming techniques
  - Architectural design techniques
  - User-interface design techniques

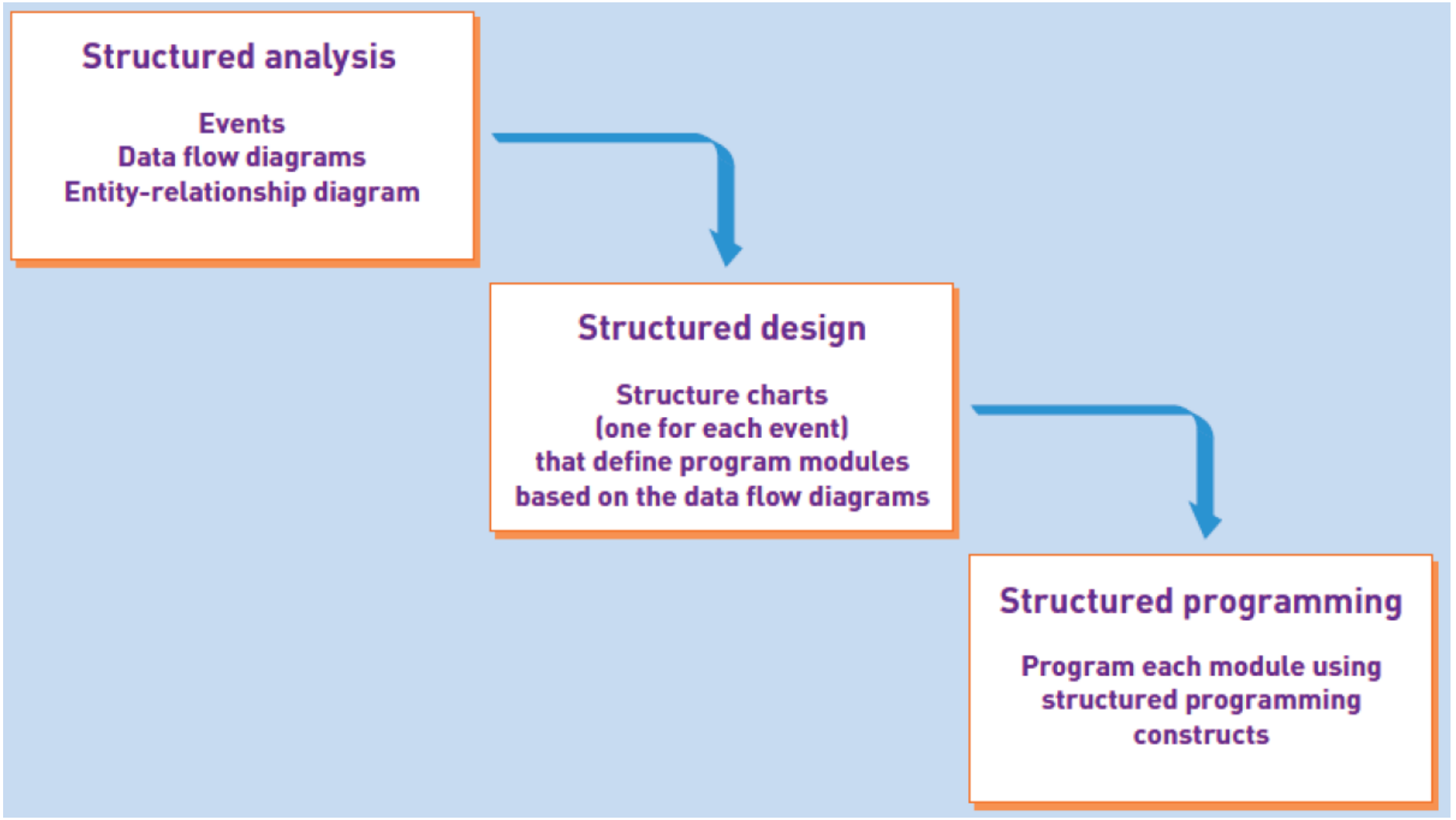
• Software applications that assists developers in creating models or other components required for a project.

- Project management application
- Drawing or graphic application
- Word processor or text editor
- Visual modeling tool
- Integrated development environment (IDE)
- Database management application
- Reverse-engineering tool
- Code generate tool

8.4 Two approaches to software construction and modeling

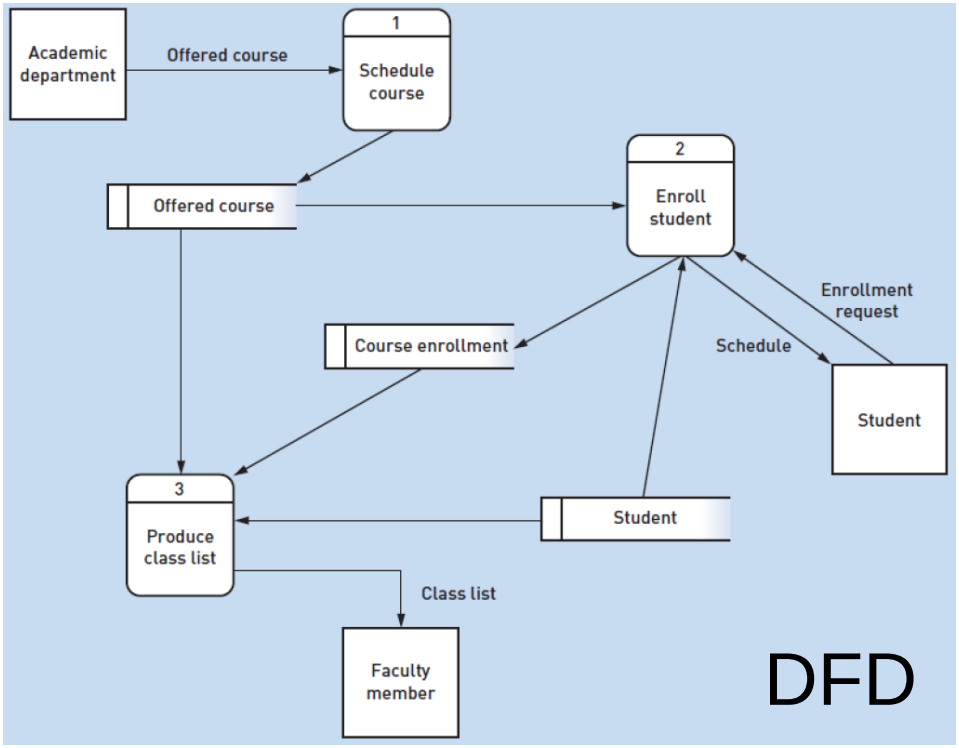
Structure approach	• Structure programming • Structured design • Structured chart • Structured analysis
Object-Oriented approach	• Object-oriented analysis (OOA) • Object-oriented design (OOD) • Object-oriented programming (OOP)

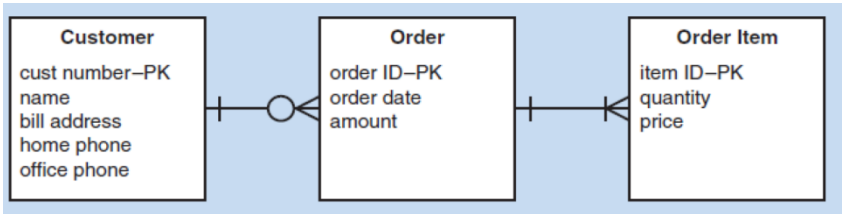
Structure Approach



Structure Approach

- Structured analysis : The technique determines a requirement by analyzing each module..

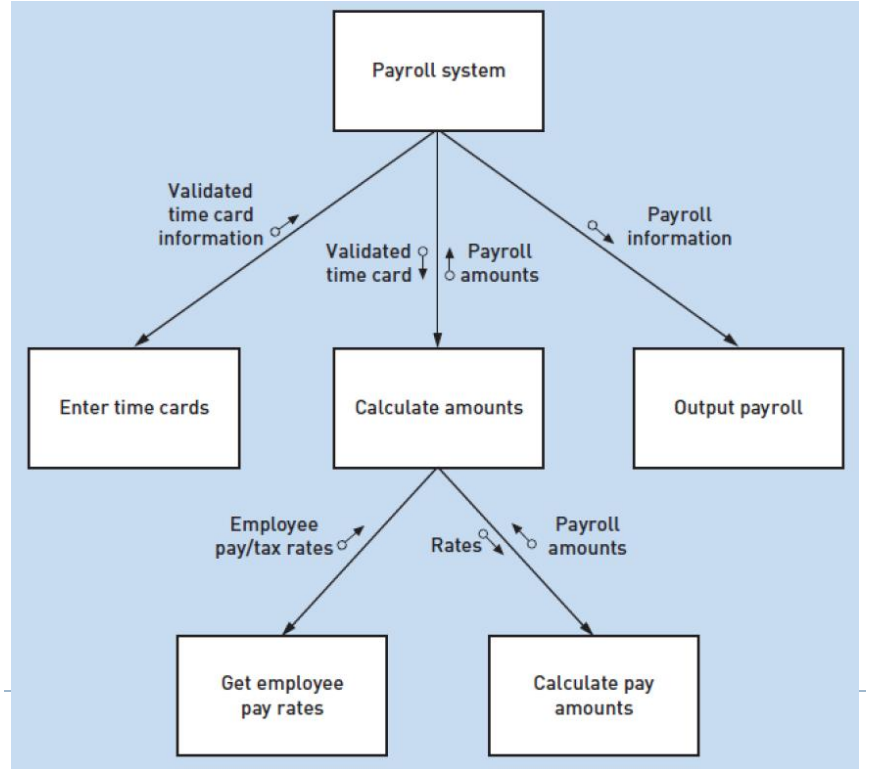




ERD

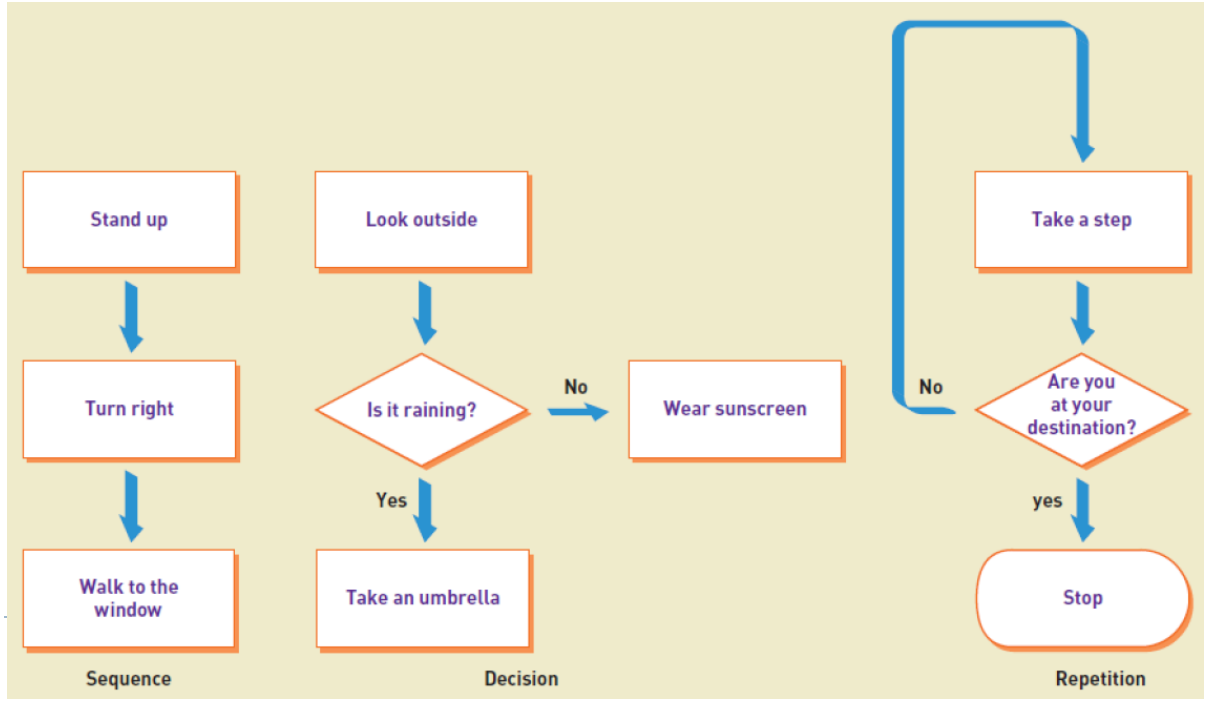
Structure Approach

- Structure design :The organizing program as a set of modules informs the hierarchical structure.
- Structure chart : A graphical diagram shows the hierarchical organization of modules.



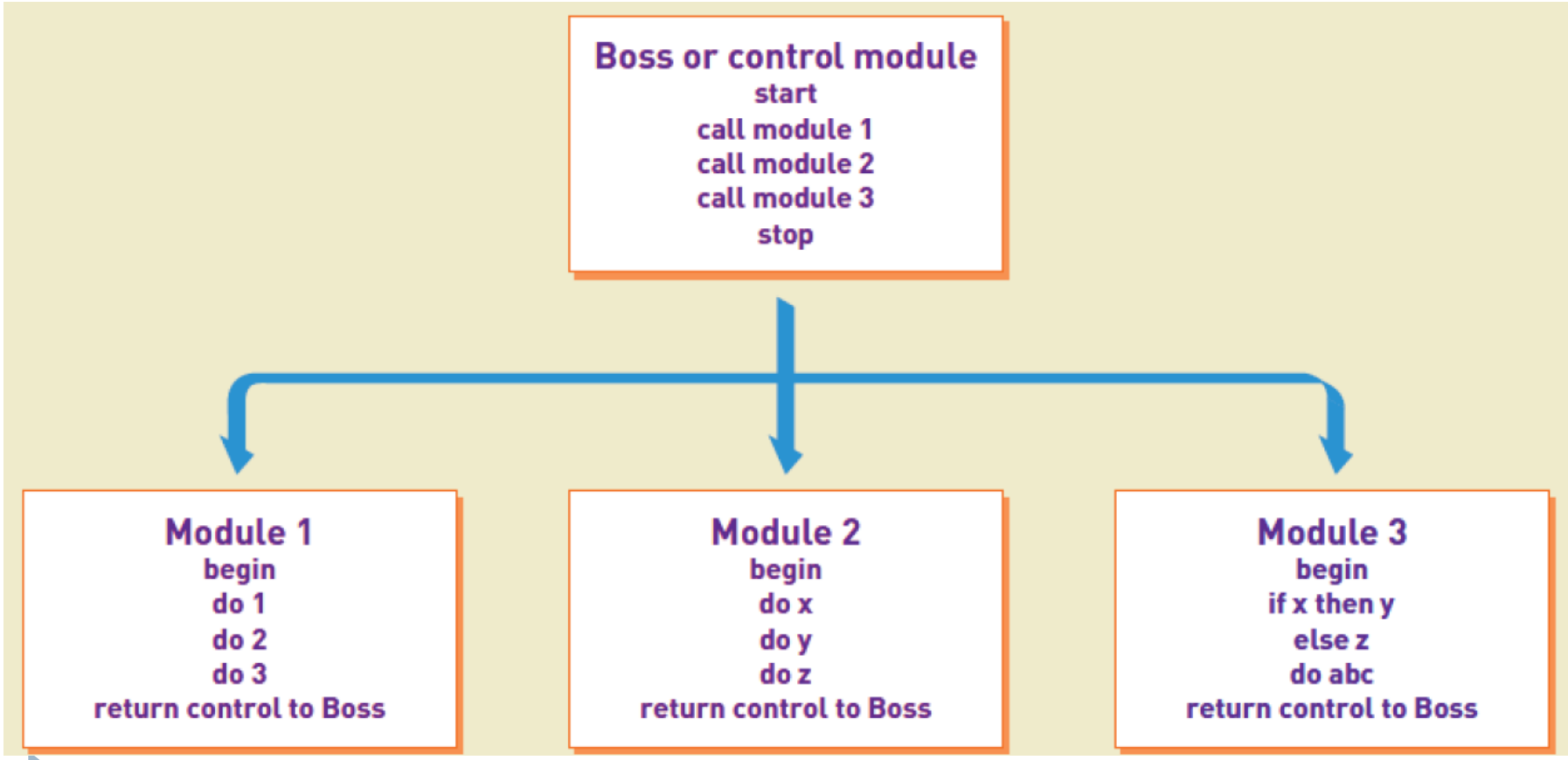
Structure Approach

- Structure programming: the specific of program is
  - A sequence of program statements
  - A decision point at which one set or another set of statements executions
  - A repetition of a set of statements



Structure Approach

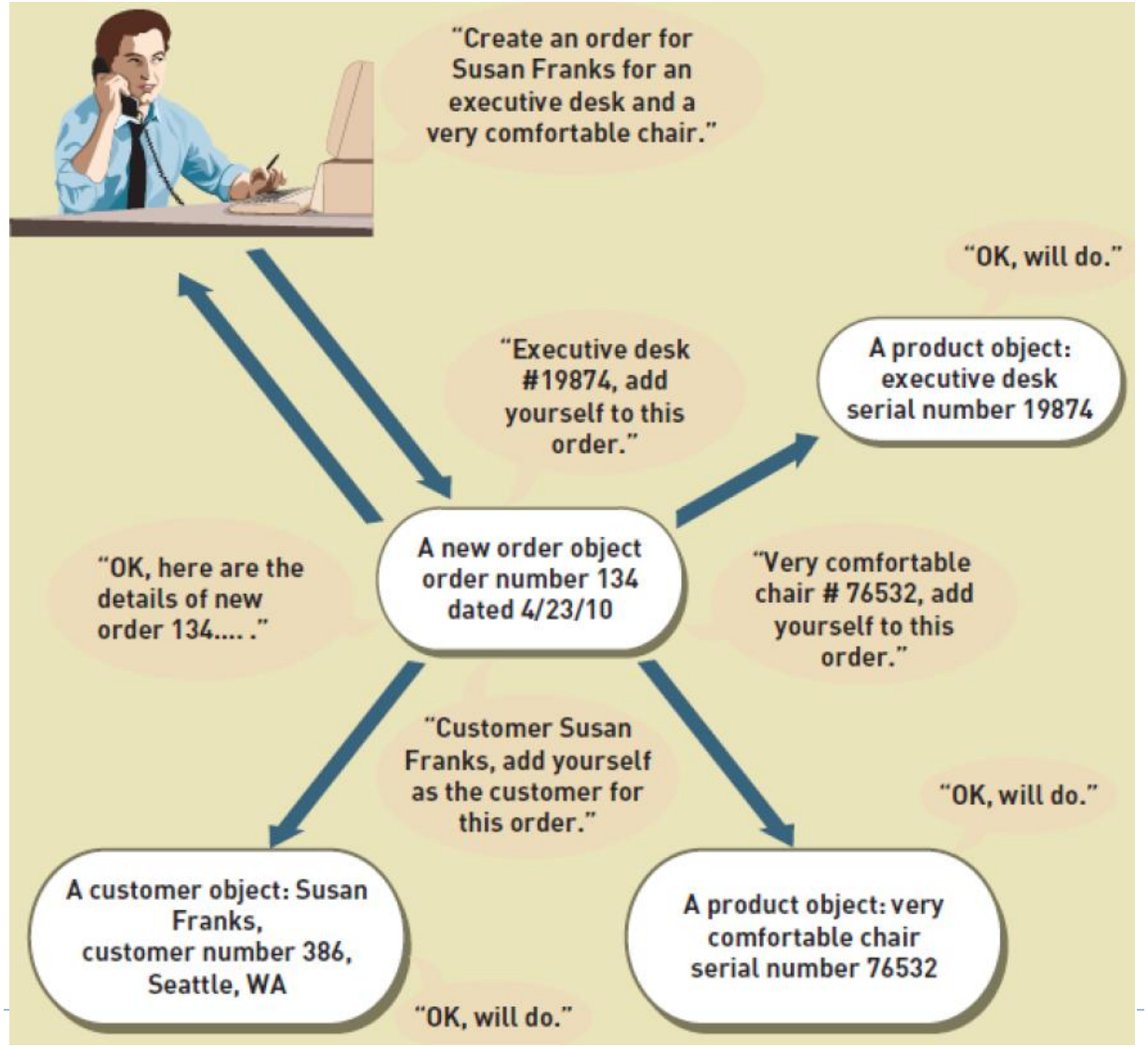
- Top-down programming divides more complex programs into hierarchy modules.



The Object-oriented approach

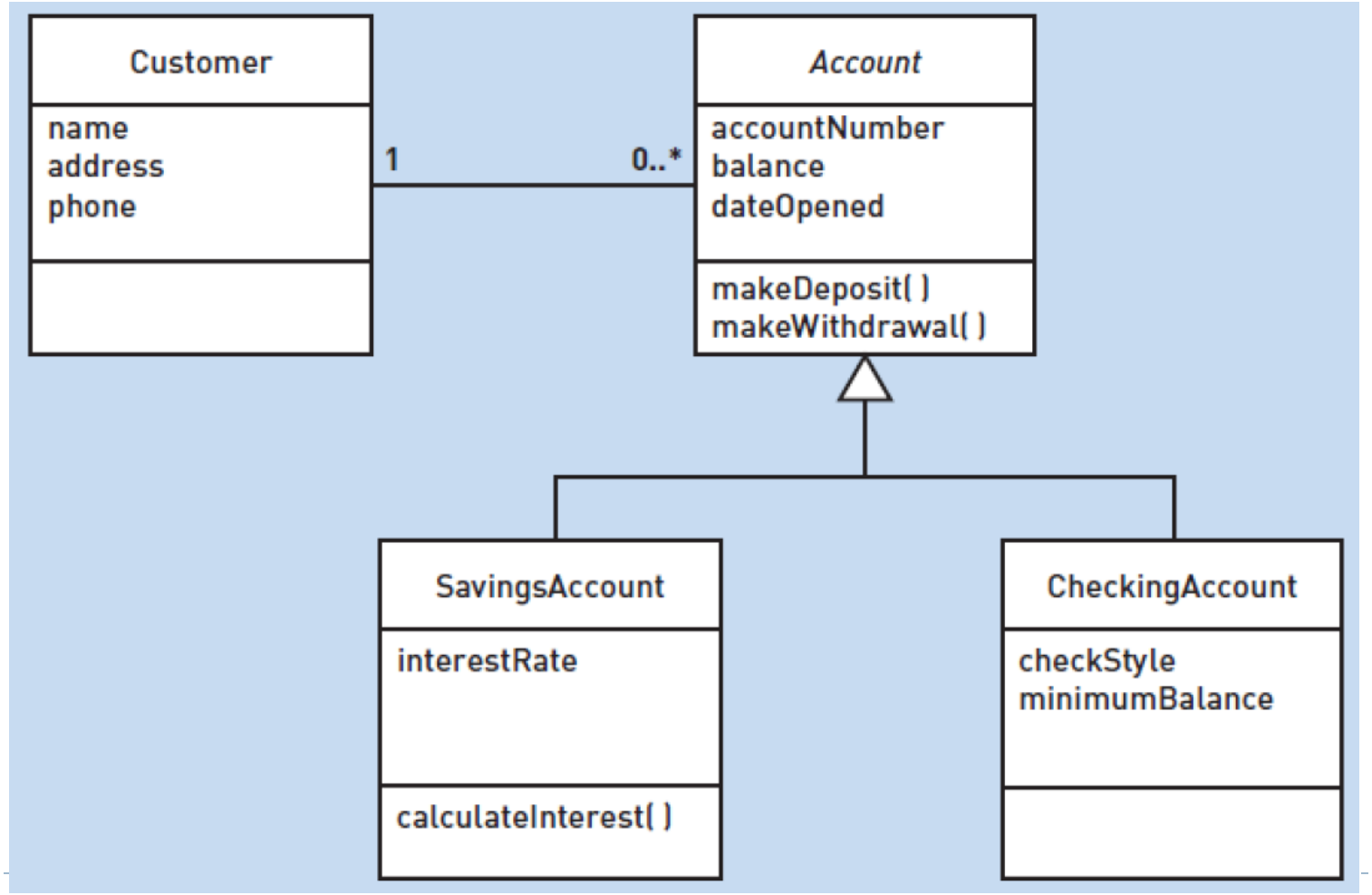
- "Object" A thing in an information system that response to message by exciting functions or methods
- OOA defines the objects that do the work and determine what user interactions (called use cases).
- OOD defines all the additional type of objects that are necessary to communicate with people and devices in the system.
- OOP is the programming language that uses the object concept to generate code.

The Object-oriented approach



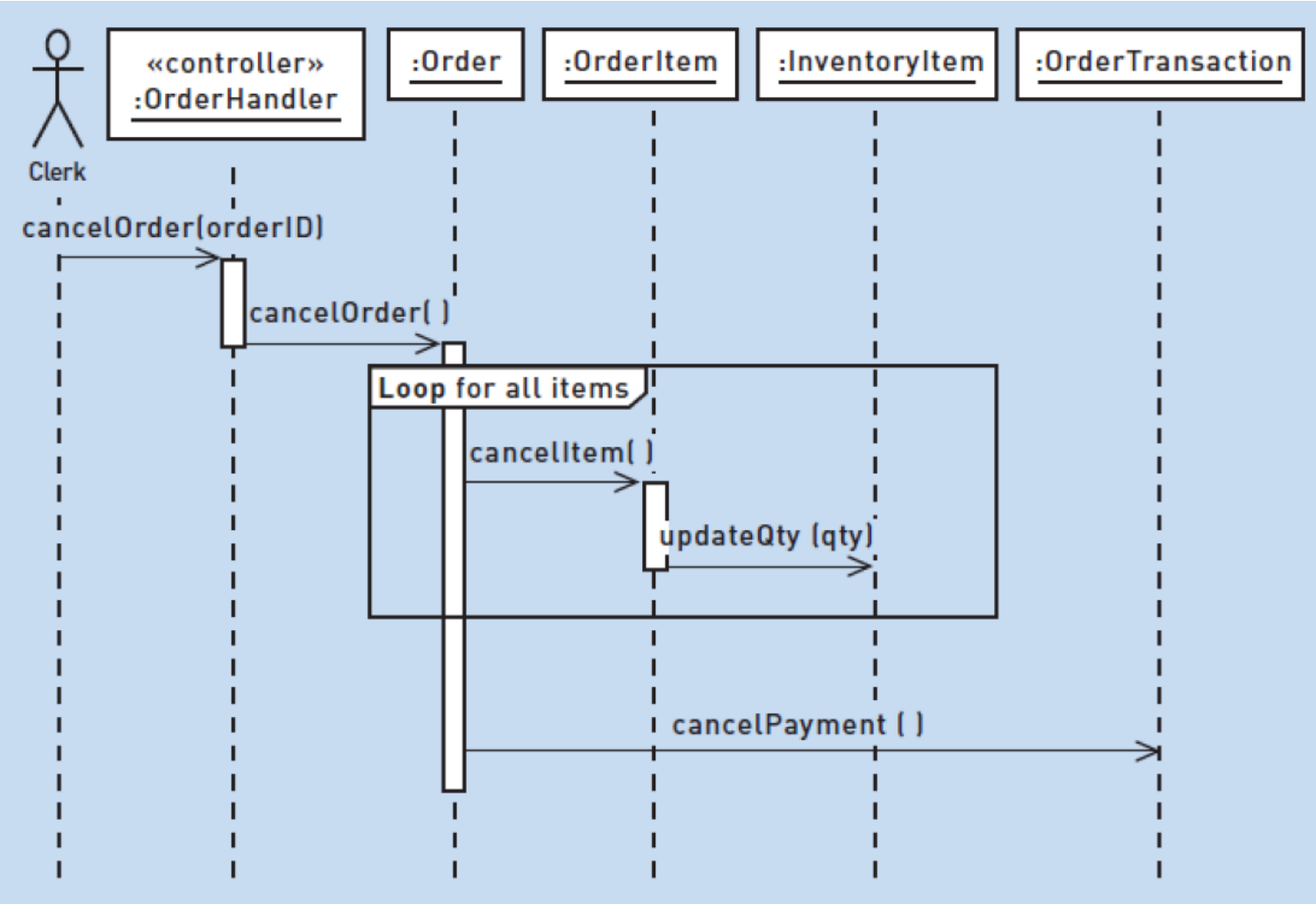
The Object-oriented approach

- UML design class diagram



The Object-oriented approach

- A UML sequence diagram showing object interaction for a use case.



8.5 Agile development

- The core philosophy

The value responding to change over following a plan

Value individual and interaction over processes and tool

Value working software over comprehensive documentation

Value customer collaboration over contract negotiation

Agile modeling (AM) principles

- AM is a guiding philosophy in models that necessary with a valid need and at the right level of detail are created.

1) Develop software as your primary goal.

2) Enable the next effort as your secondary goal.

3) Minimize your modeling activity—few and simple.

4) Embrace change, and change incrementally.

5) Model with a purpose.

Agile modeling (AM) principles

6) Build multiple models

7) Build high-quality models and get feedback rapidly.

8) Focus on content rather than representation.

9) Learn from each other with open communication.

10) Know your models and how to use them.

11) Adapt to specific project needs.

Summary

- Predictive approach to SDLC
  - Waterfall model
- Adaptive approach to SDLC
  - Spiral model
  - Incremental development
  - Waking skeleton
- Methodologies
  - Models
    - Abstraction presentation
    - Graphical presentation
  - Tools
  - Techniques
- Software construction
  - Structured approach
  - Object-oriented approach
- Agile development
  - Philosophy
  - Agile modeling principles