INTRODUCTION TO SYSTEMS ANALYSIS AND DESIGN:

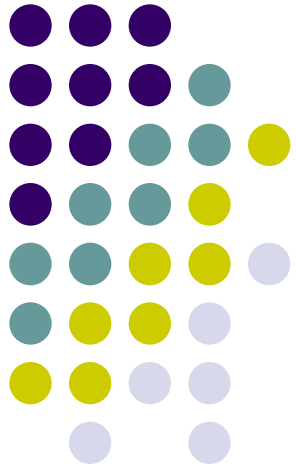
AN AGILE, ITERATIVE APPROACH

SATZINGER | JACKSON | BURD

CHAPTER 8

Approaches to System Development

Chapter 8



Introduction to Systems Analysis and Design:
An Agile, Iteractive Approach
6th Ed

Satzinger, Jackson & Burd

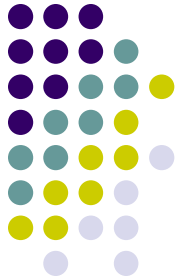
Chapter 8 Outline

- The Systems Development Life Cycle (SDLC)
- The Support Phase of the SDLC
- Methodologies, Models, Tools and Techniques
- Two Approaches to Software Construction and Modeling
- Agile Development

Learning Objectives

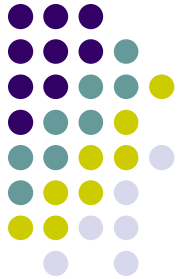
- Compare the underlying assumptions and uses of a predictive and an adaptive system development life cycle (SDLC)
- Describe the key activities and tasks of information system support
- Explain what comprises a system development methodology—the SDLC as well as models, tools, and techniques
- Describe the two overall approaches used for software construction and modeling: the structured approach and the object-oriented approach
- Describe the key features of Agile development

Overview



- Chapter 1 demonstrated a system development project that used an iterative and agile system development life cycle (SDLC)
- Later chapters focused on Systems Analysis activities and tasks and some System Design activities and tasks
- Now we return to look at the SDLC and related concepts in more detail
  - Predictive versus Adaptive SDLC variations
  - Activities and Tasks of System Support
  - Models, Methodologies, Tools and Techniques
  - Impacts of Traditional versus OO development
  - Agile Development

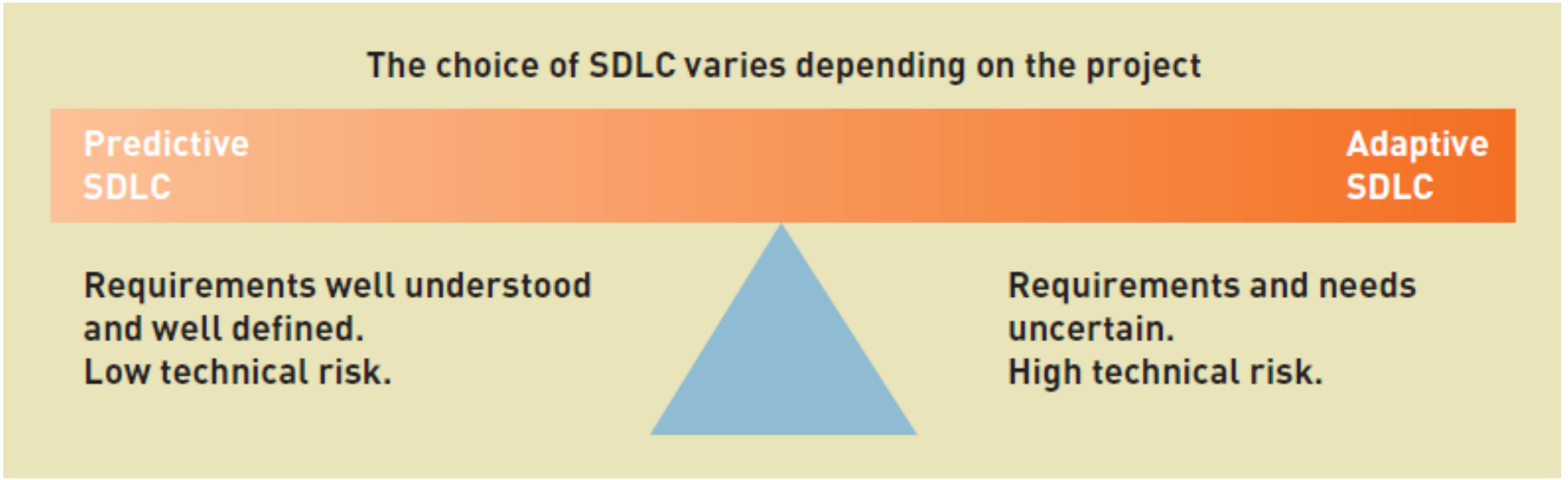
The System Development Life Cycle (SDLC)



- There are two general approaches to the SDLC
- Predictive Approach
  - Waterfall model
  - Assumes the project can be planned in advance and that the information system can be developed according to the plan
  - Requirements are well understood and/or low technical risk
- Adaptive Approach to the SDLC
  - Iterative model (as see in this text)
  - Assumes the project must be more flexible and adapt to changing needs as the project progresses
  - Requirements and needs are uncertain and/or high technical risk

The System Development Life Cycle (SDLC)

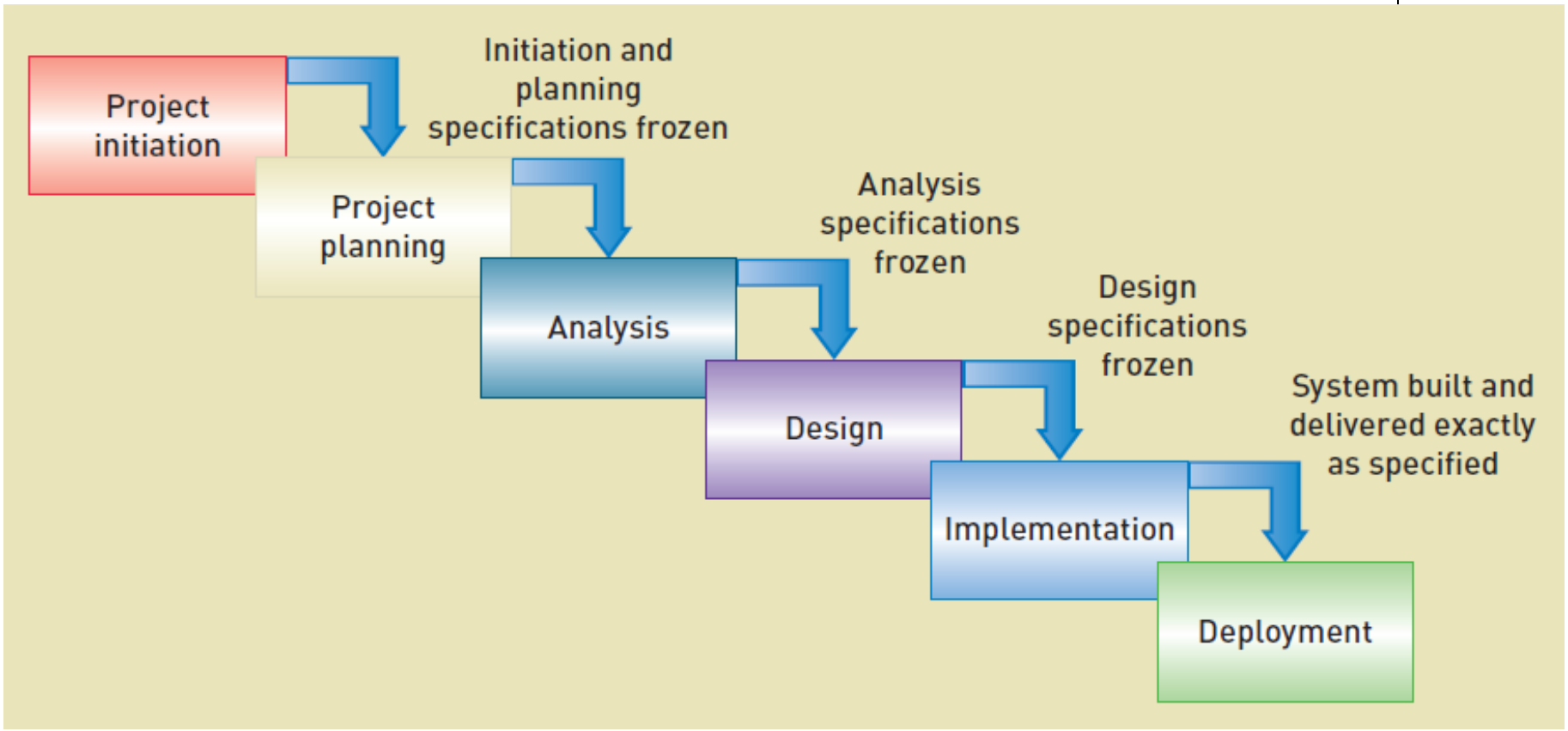
- Most projects fall on a continuum between Predictive and Adaptive



Traditional Predictive SDLC

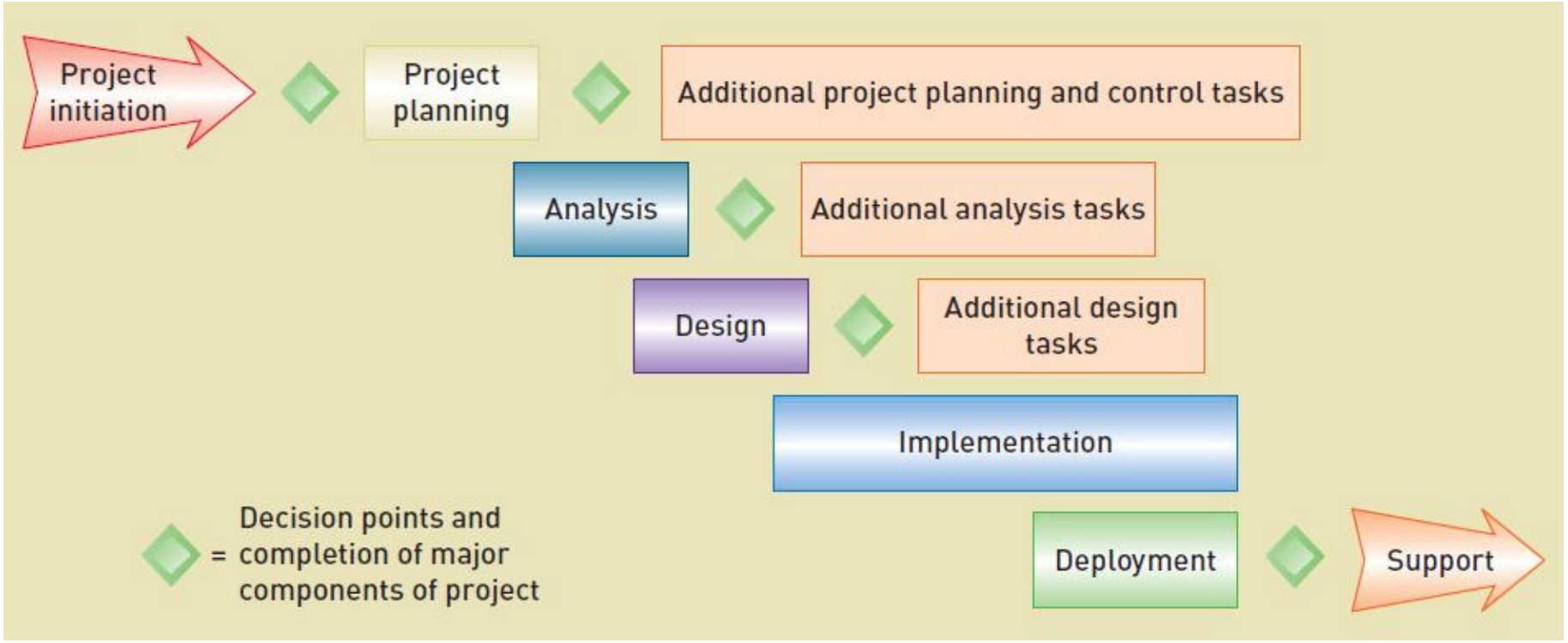
- Earlier approach based on engineering
- Typically have sequential *Phases*
  - Phases are related groups of development activities, such as planning, analysis, design, implementation, and deployment
- Waterfall model
  - SDLC that assumes phases can be completed sequentially with no overlap or iteration
  - Once one phase is completed, you fall over the waterfall to the next phase, no going back

Traditional Predictive SDLC

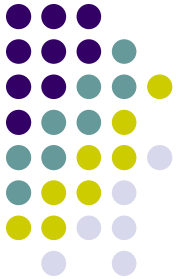


Newer Overlapping Phases Predictive SDLC

- More flexibility, but still assumes predictive planning and sequential phases

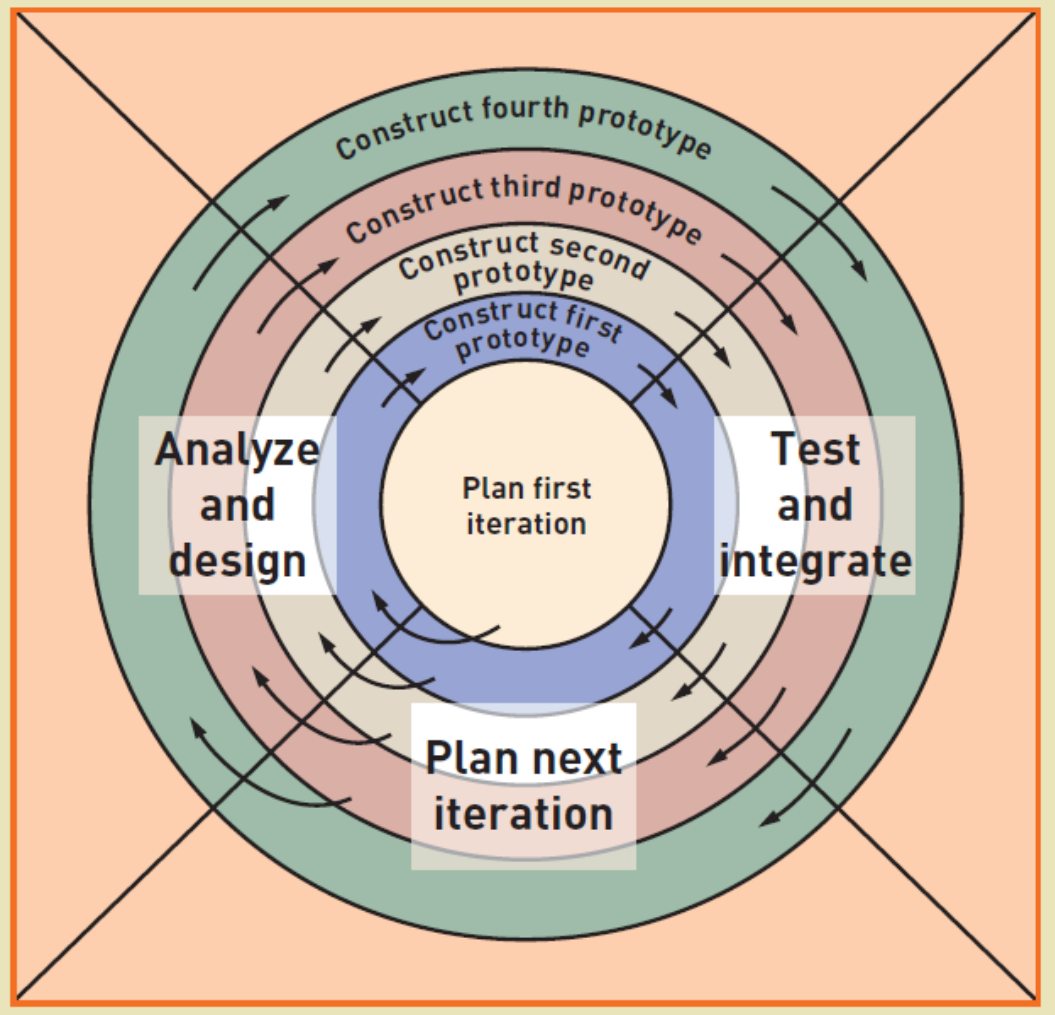


Newer Adaptive SDLC



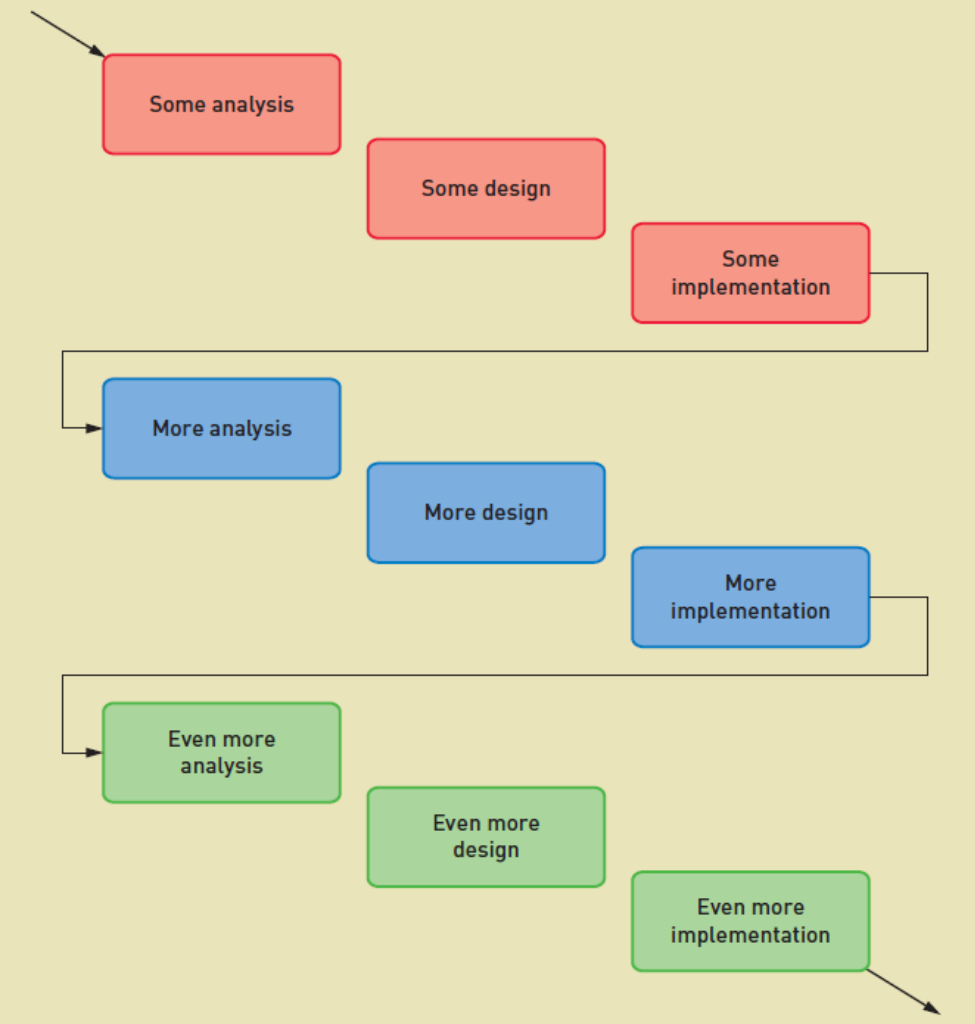
- Emerged in response to increasingly complex requirements and uncertain technological environments
- Always includes iterations where some of design and implementation is done from the beginning
- Many developers claim it is the ***only*** way to develop information systems
- Many IS managers are still sceptical

Spiral Model
The First Adaptive SDLC



Iterative Model

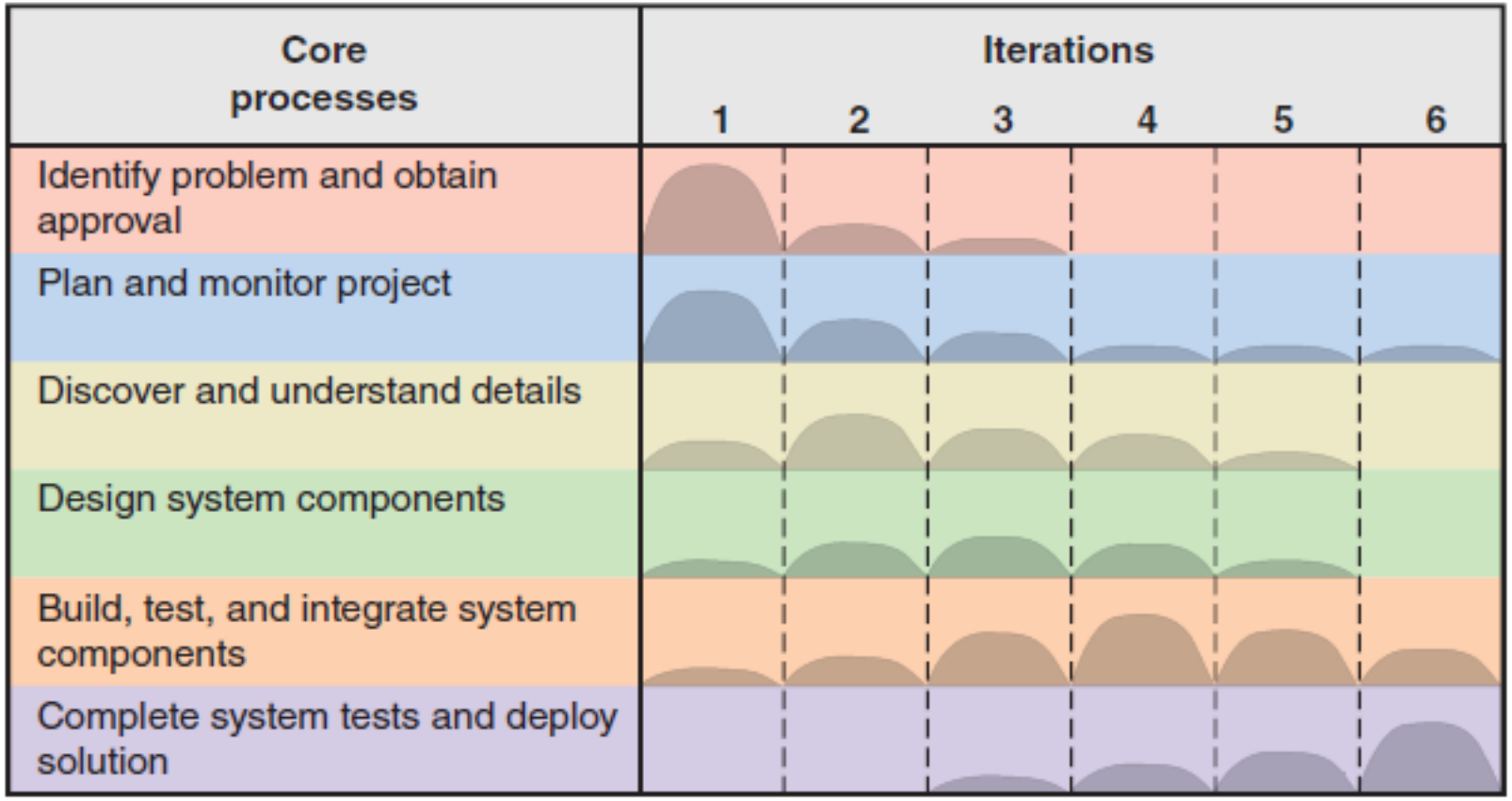
Popular Way to Represent Adaptive SDLC



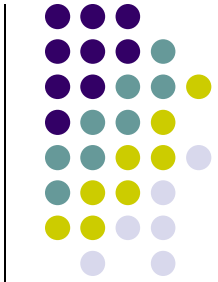
Core Processes vs. Iterations Model

The Adaptive SDLC used in this Text

- Shows core processes, not phases, plus iterations in a sequence for management checkpoints
- Based on the Unified Process SDLC (see chapter 14)



Additional Adaptive Concepts



- Incremental Development
  - An approach that completes portions of the system in increments
  - A system is implemented and partially deployed in steps during the project
  - Gets part of working system into users' hands sooner
- Walking Skeleton
  - An approach in which the complete system structure is built early, but with bare-bones functionality

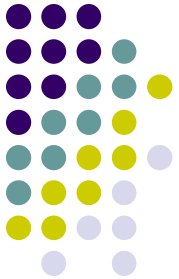
The SDLC Support Phase

- All information systems need to be supported once completed
- Predictive SDLCs typically include support as a project phase
- Adaptive SDLCs treat support as a separate project
- Support Activities
  - Activities whose objective is to maintain and enhance the system after it is installed and in use

Support Activities

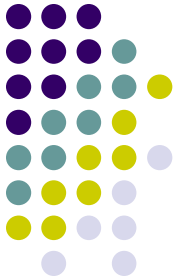
- Maintaining the system
  - Fix problems/error
  - Make minor adjustments
  - Update for changes in operating systems or environments
- Enhancing the system
  - Add desired functionality
  - Add or change functionality to comply with regulations or legislation
- Supporting the users
  - Ongoing user training
  - Help desk

Methodologies, Models, Tools, and Techniques



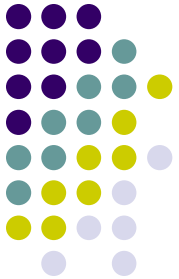
- Methodologies
  - Provides guidelines for every facet of system development: What to do when, why and how
  - Specifies an SDLC with activities and tasks
  - Specifies project planning and project management models and reporting
  - Specifies analysis and design models to create
  - Specifies implementation and testing techniques
  - Specifies deployment and support techniques
- Other term used is *System Development Process*

Methodologies, Models, Tools, and Techniques



- Model
  - An abstraction of an important aspect of the real world.
  - Makes it possible to understand a complex concept by focusing only on a relevant part
  - Each model shows a different aspect of the concept
  - Crucial for communicating project information
- In IS, some models are of system components
- Some models are used to manage the development process

Methodologies, Models, Tools, and Techniques



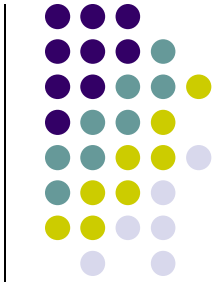
Some models of system components

- 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
- Organizational hierarchy chart
- Financial analysis models - NPV, payback period

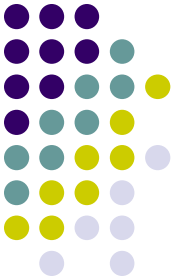
Methodologies, Models, Tools, and Techniques



- Tools
  - Software applications that assists developers in creating models or other components required for a project

Project management application
Drawing/graphics application
Word processor/text editor
Visual modeling tool
Integrated development environment (IDE)
Database management application
Reverse-engineering tool
Code generator tool

Methodologies, Models, Tools, and Techniques

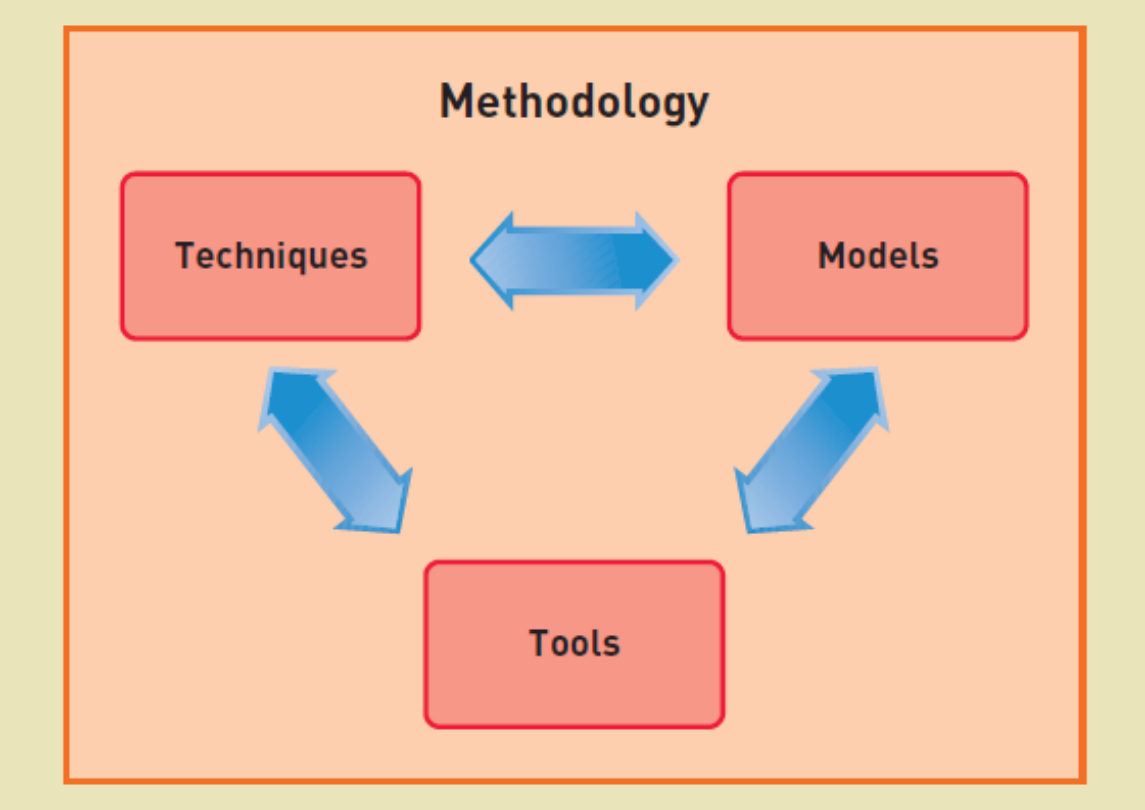


- Technique
  - A collection of guidelines that help an analyst complete an activity or task
  - Learning techniques is the key to having expertise in a field

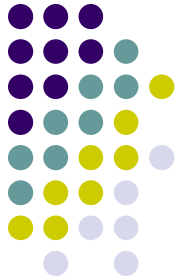
Strategic planning techniques
Project management techniques
User interviewing techniques
Data-modeling techniques
Relational database design techniques
Structured programming technique
Software-testing techniques
Process modeling techniques
Domain modeling techniques
Use case modeling techniques
Object-oriented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

Methodologies, Models, Tools, and Techniques

- A ***Methodology*** includes a collection of techniques that are used to complete activities and tasks, including modeling, for every aspect of the project



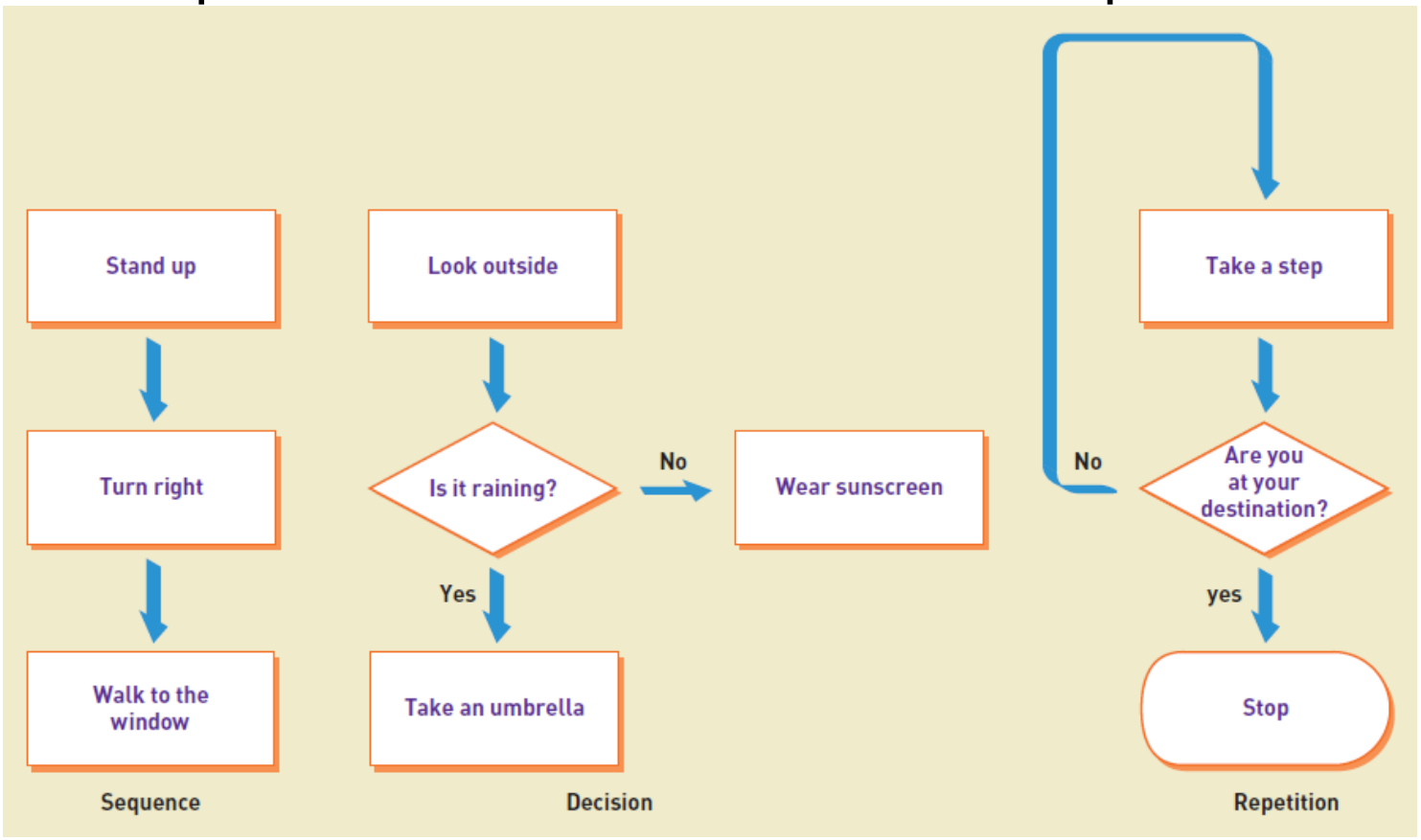
Two Approaches to Software Construction and Modeling



- The Structured Approach
  - Earlier approach. Assumes a system is a collection of processes that interact with data
  - Structured analysis, structured design, and structured programming
- The Object-Oriented Approach
  - More recent approach. Assumes a system is a collection of objects that interact to complete tasks
  - OO analysis, OO design, and OO programming

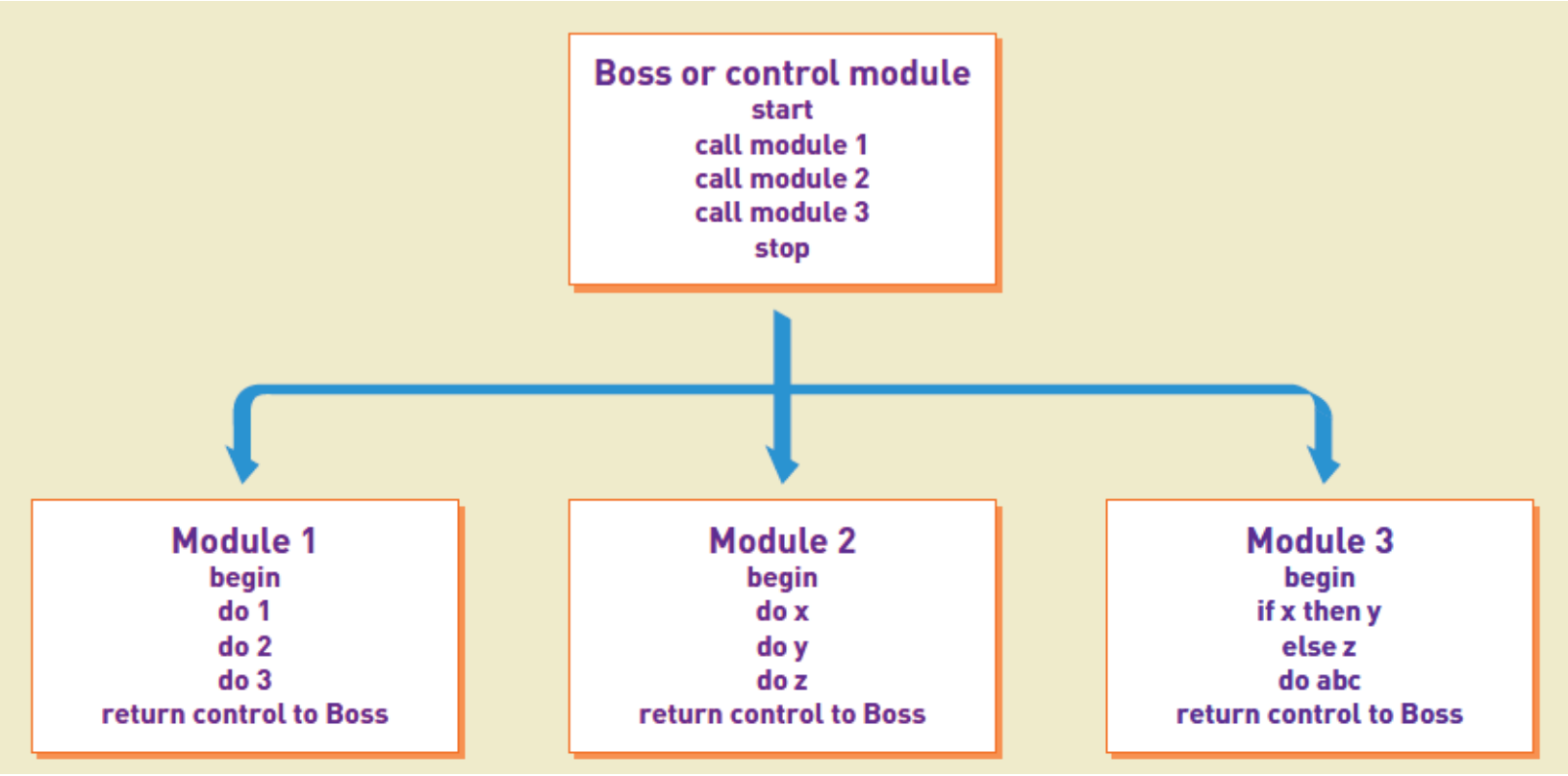
The Structured Approach

- Structured Programming
  - Sequence, selection/decision, and repetition



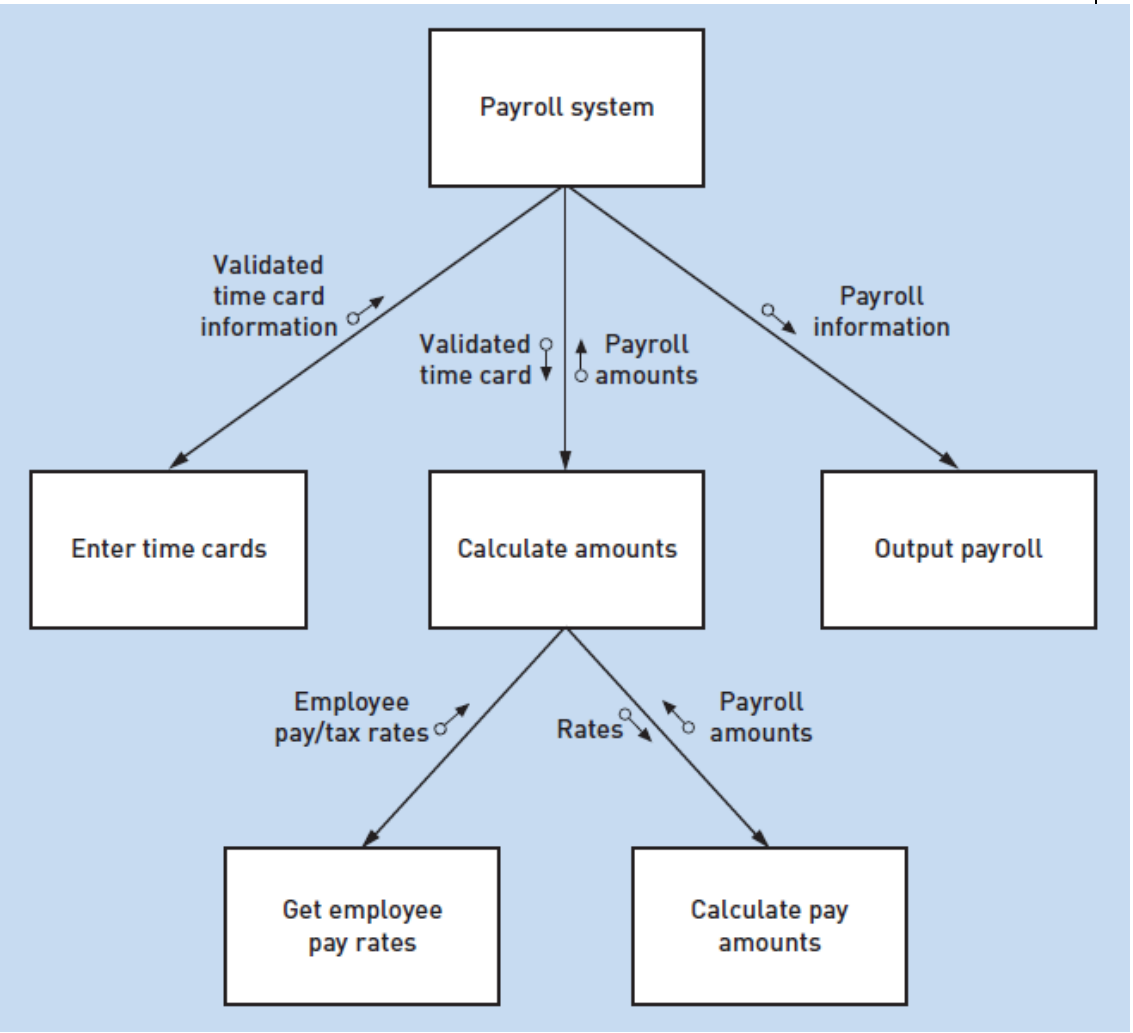
The Structured Approach

- Top down, modular programming



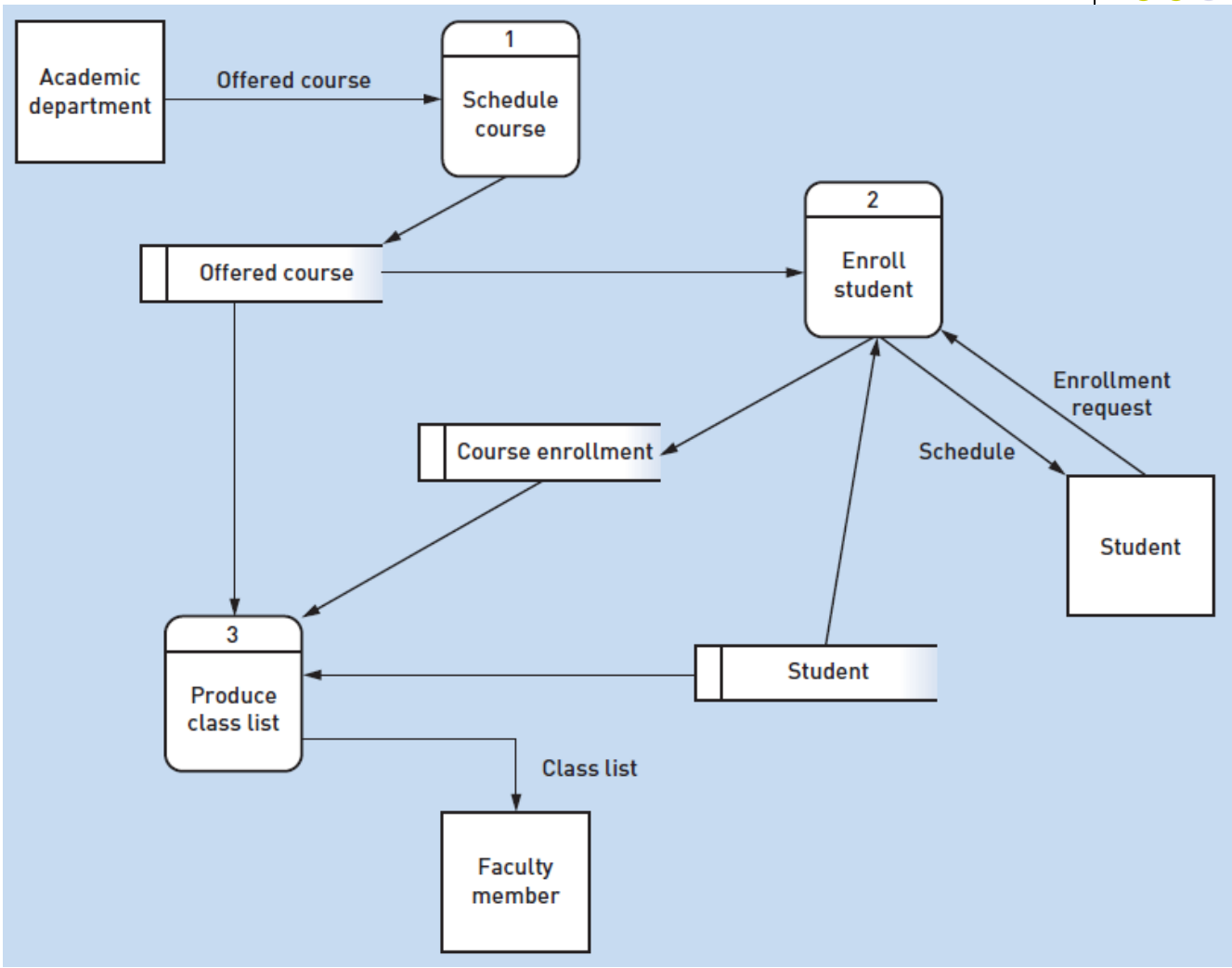
The Structured Approach

- Structured Design
  - Structure chart with data couples shown



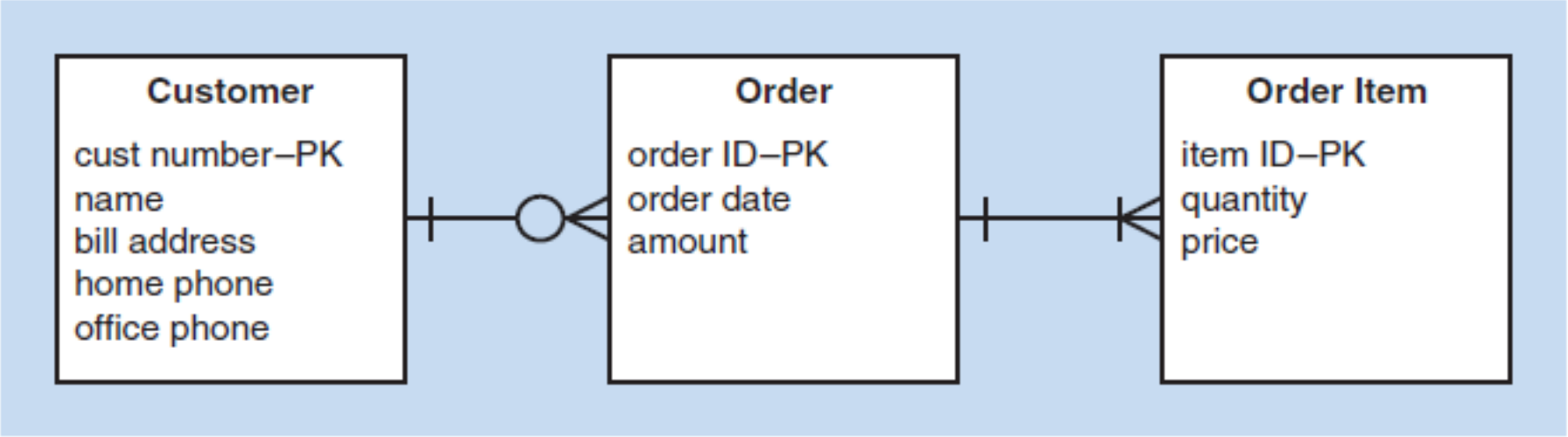
The Structured Approach

- Structured Analysis
  - Data flow diagram (DFD)



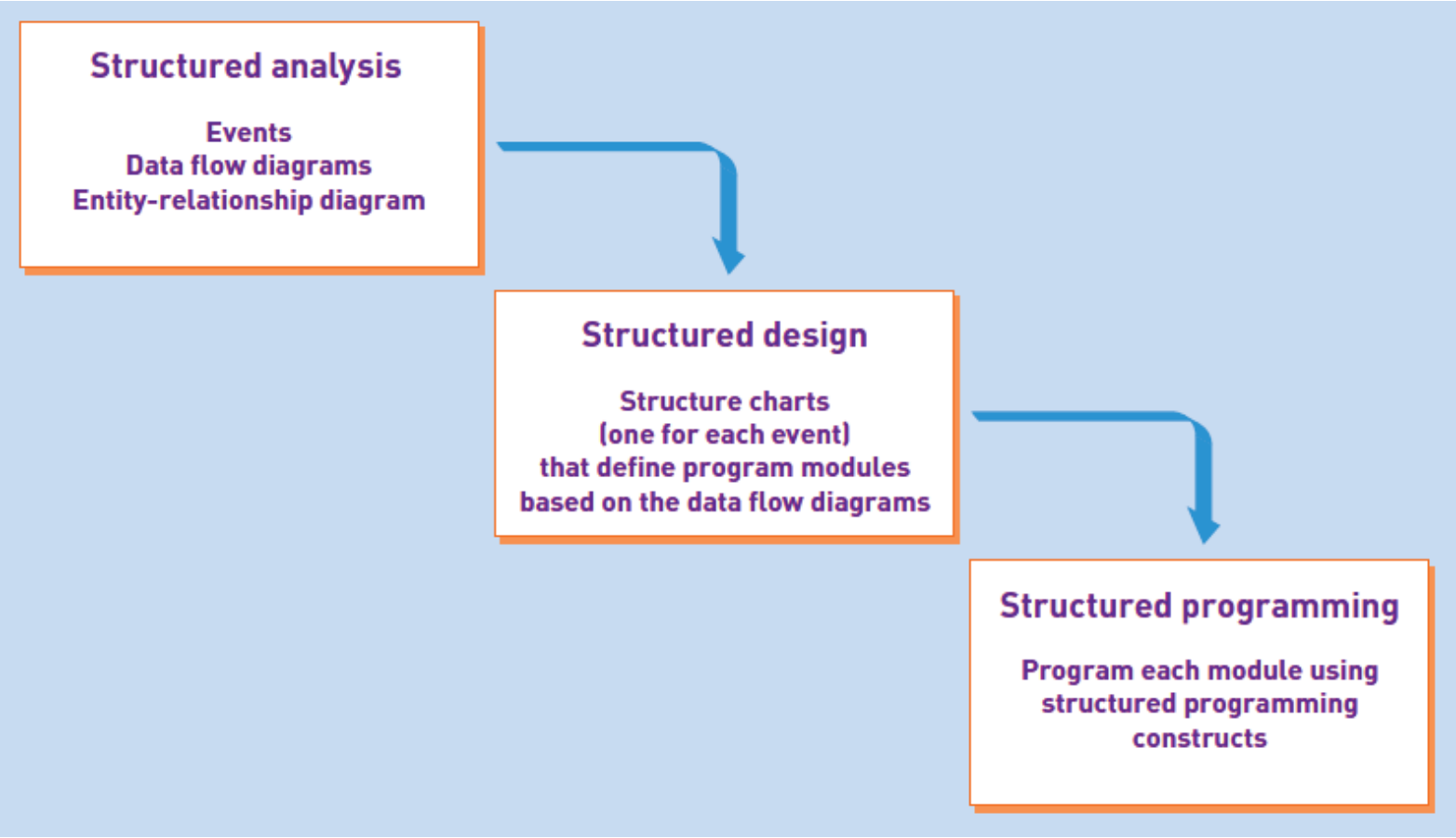
The Structured Approach

- Entity-relationship diagram

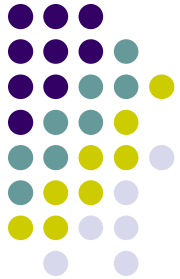


The Structured Approach

- How it fits together



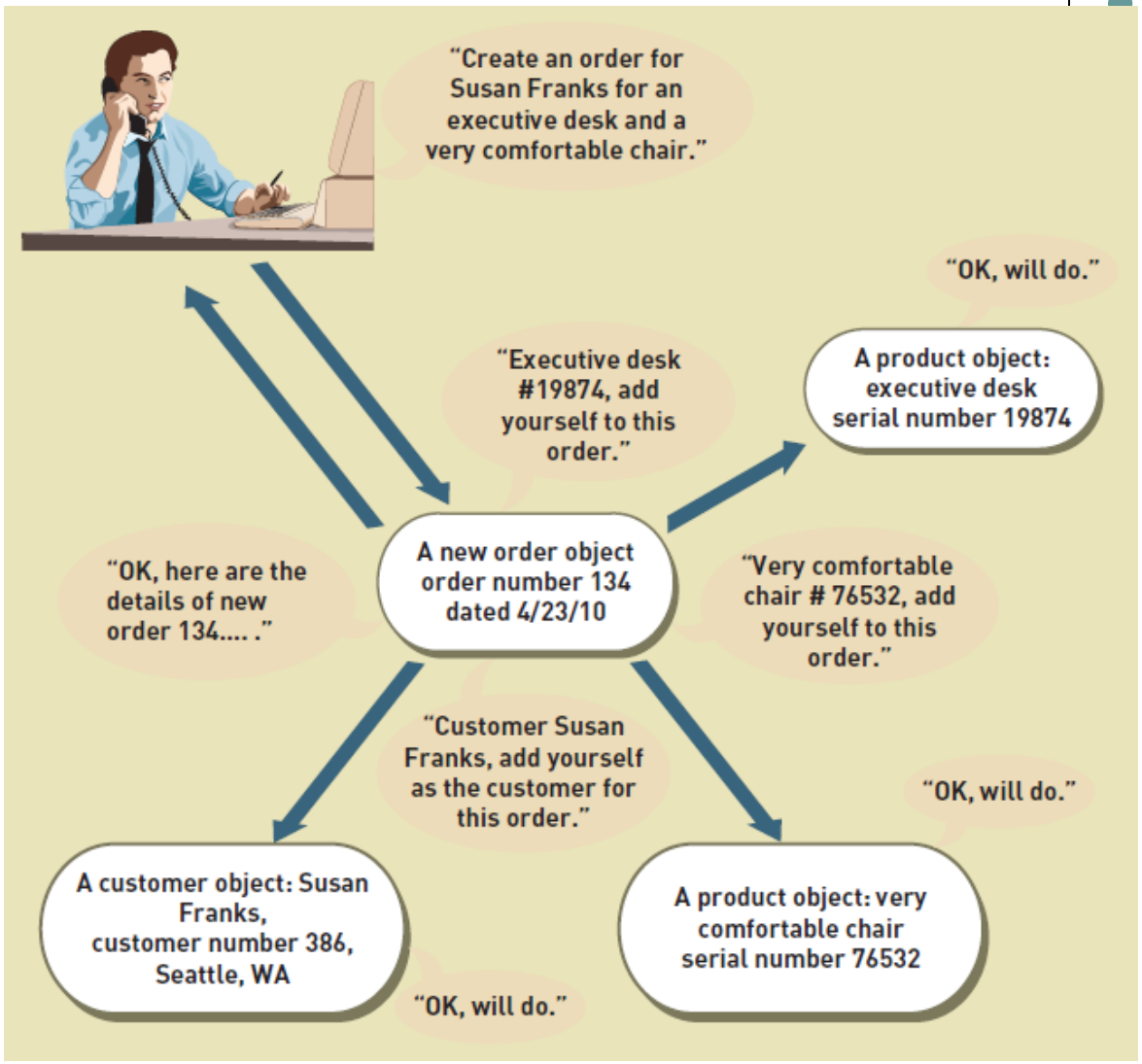
The Object-Oriented Approach



- Object-oriented analysis (OOA)
  - The process of identifying and defining the use cases and sets of objects (classes) in the new system
- Object-oriented design (OOD)
  - Defining all of the types of objects necessary to communicate with people and devices and showing how they interact to complete tasks
- Object-oriented programming (OOP)
  - Writing statements that define the actual classes and what each object of the class does

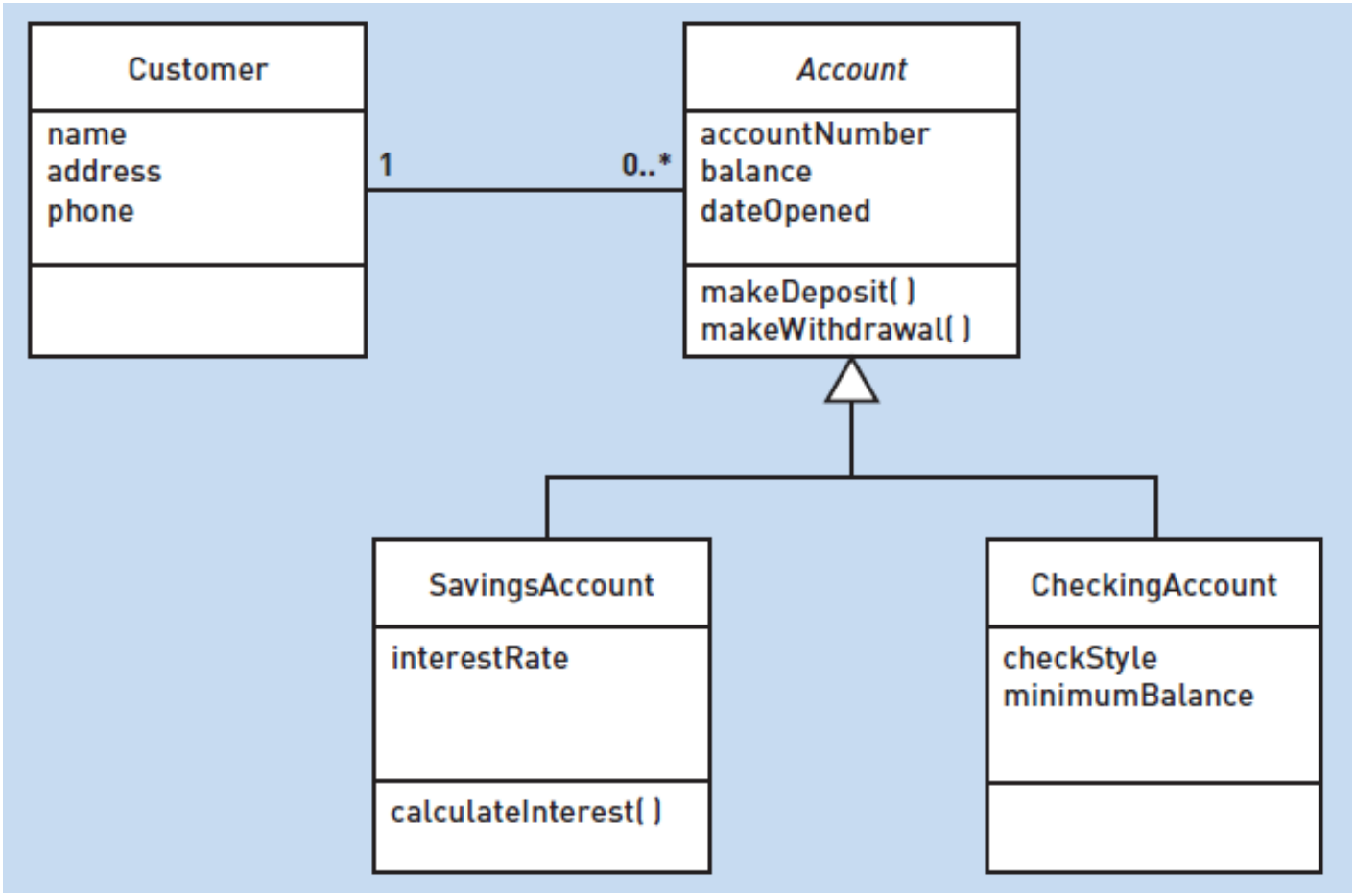
The Object-Oriented Approach

- Example showing the OO concept
- Objects collaborate to get a task done



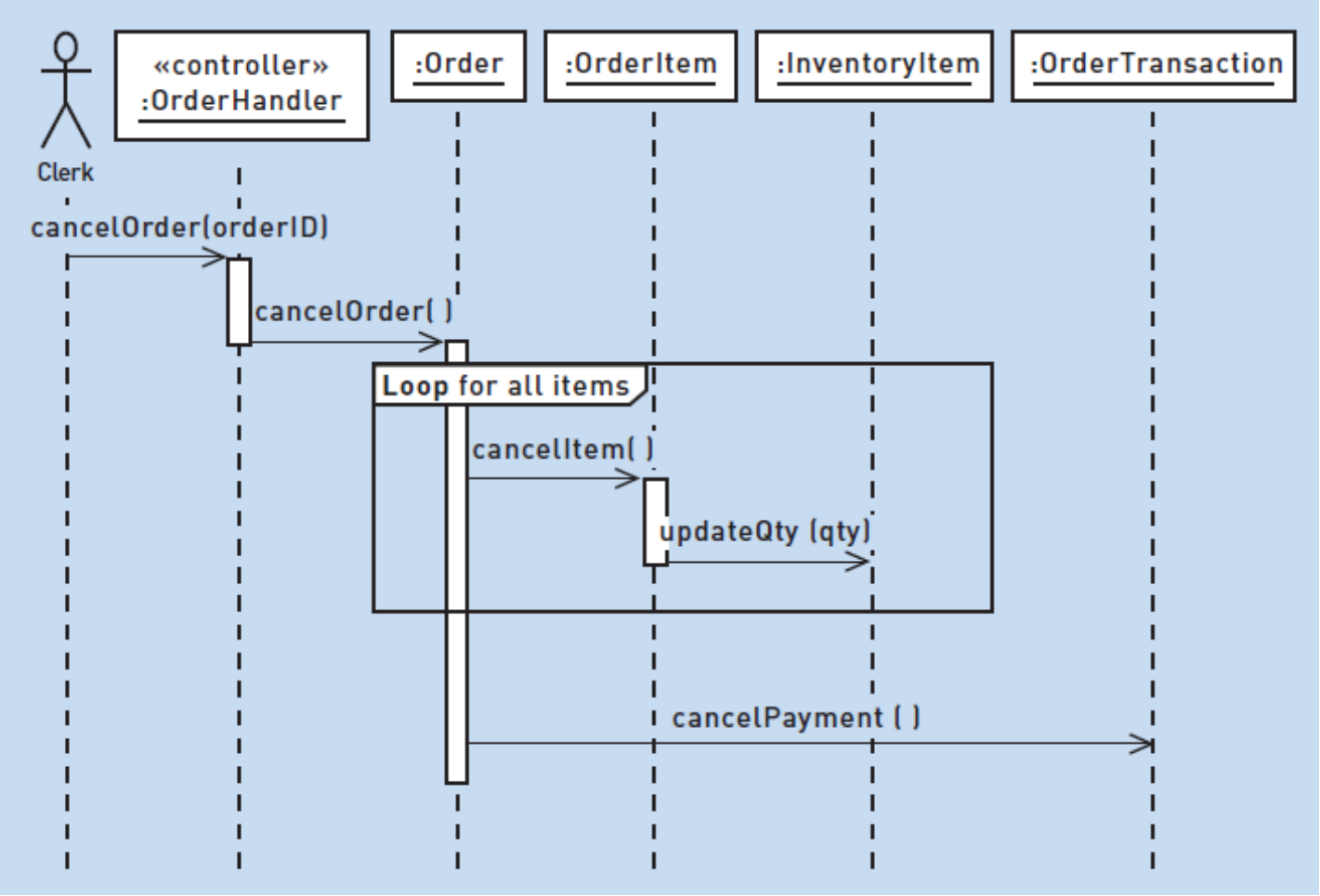
The Object-Oriented Approach

- UML Design Class Diagram

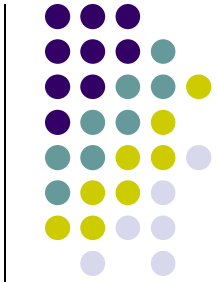


The Object-Oriented Approach

- UML Sequence Diagram

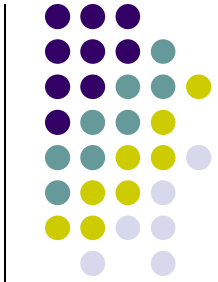


Agile Development



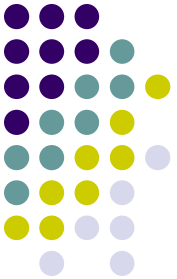
- A guiding philosophy and set of guidelines for developing information systems in an unknown, rapidly changing environment
- Complements Adaptive SDLCs and Methodologies that support it
- Takes adaptive and makes sure developers are fast on their feet to respond to changes
- Some specific examples of Agile Methodologies/Development Processes are covered in Chapter 14

Agile Development Philosophies and Values



- This text emphasizes agile values, as stated by the “Manifesto for Agile Development”
  - Value responding to change over following a plan
  - Value individuals and interactions over processes and tools
  - Value working software over comprehensive documentation
  - Value customer collaboration over contract negotiation

Agile Modeling



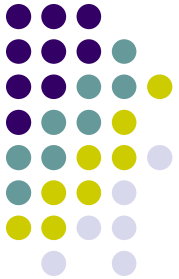
Agile Modeling principles

- Develop software as your primary goal.
- Enable the next effort as your secondary goal.
- Minimize your modeling activity—few and simple.
- Embrace change, and change incrementally.
- Model with a purpose.
- Build multiple models.
- Build high-quality models and get feedback rapidly.
- Focus on content rather than representation.
- Learn from each other with open communication.
- Know your models and how to use them.
- Adapt to specific project needs.

Summary

- This chapter covers approaches to system development in more detail
- There are two approaches to the SDLC: Predictive and Adaptive
- A predictive SDLC, also known as the waterfall model, is used when it is possible to plan the project completely in advance
- An Adaptive SDLC, which uses iteration, is used when the requirements are less certain and the project will need to react to changes
  - This text uses an adaptive approach to the SDLC

Summary (continued)



- All new information systems require support once completed
- System development project use a methodology (or development process) and many are available. A methodology includes an SDLC and tools, techniques, and models
- There are two approaches to construction and modeling software: the traditional structured approach and the newer object-oriented approach
- Agile development is the current trend in system development