Modern Systems Analysis and Design

Chapter 1

The Systems Development Environment

Learning Objectives

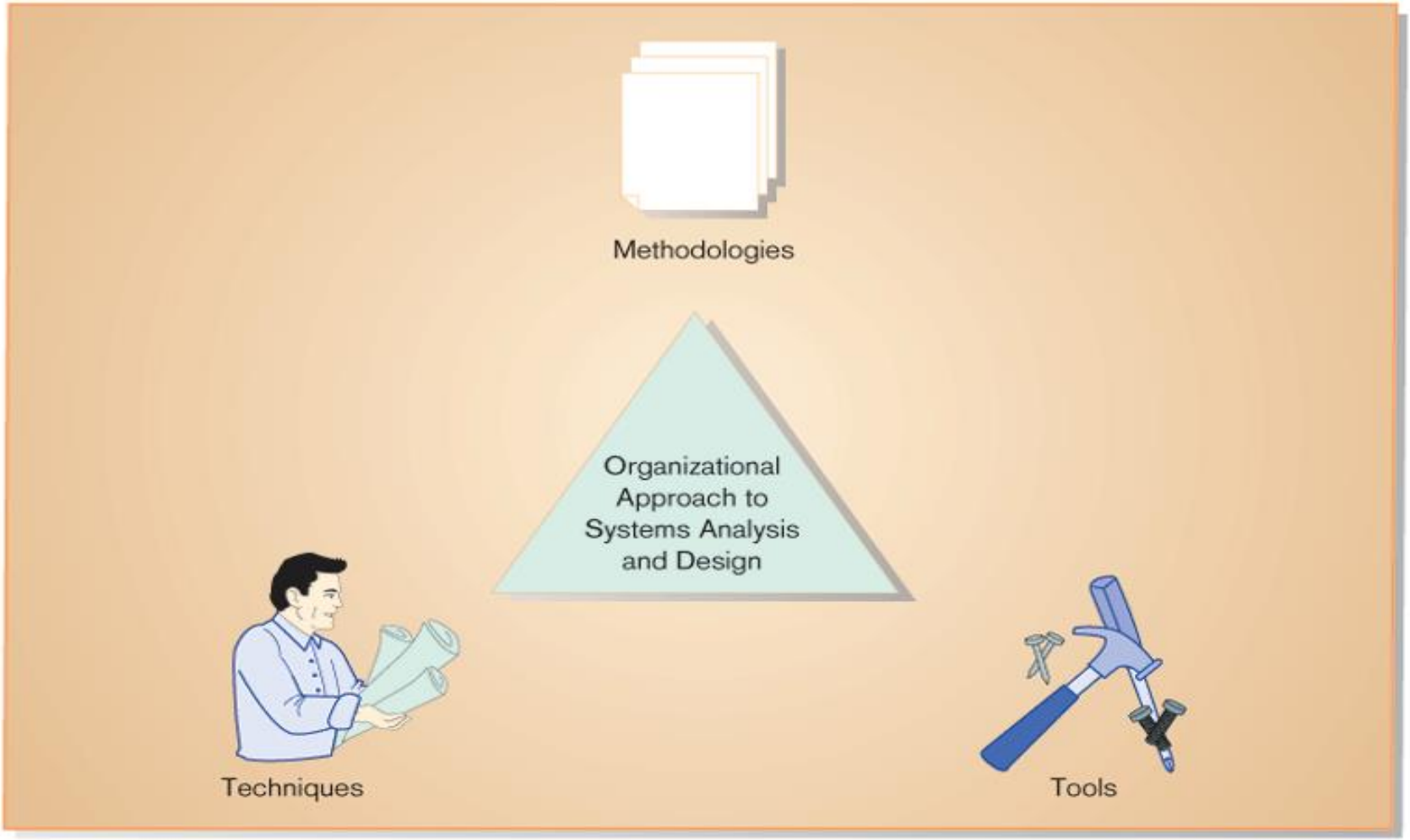
- ✓ Define information systems analysis and design.
- ✓ Describe the different types of information systems.
- ✓ Describe the information Systems Development Life Cycle (SDLC).
- ✓ Explain Rapid Application Development (RAD), prototyping, Joint Application Development (JAD), and Computer Aided Software Engineering (CASE).
- ✓ Describe agile methodologies and eXtreme programming.
- ✓ Explain Object Oriented Analysis and Design and the Rational Unified Process (RUP).

Introduction

- Information Systems Analysis and Design
  - Complex organizational process whereby computer-based information systems are developed and maintained
- Application Software
  - Computer software designed to support organizational functions or processes
- Systems Analyst
  - Organizational role most responsible for analysis and design of information systems

Introduction (cont.)

Figure 1-1 An organizational approach to systems analysis and design is driven by methodologies, techniques, and tools.



A Modern Approach to Systems Analysis and Design

- 1950s: focus on efficient automation of existing processes
- 1960s: advent of 3GL, faster and more reliable computers
- 1970s: system development becomes more like an engineering discipline
- 1980s: major breakthrough with 4GL, CASE tools, object oriented methods
- 1990s: focus on system integration, GUI applications, client/server platforms, Internet
- The new century: Web application development, wireless PDAs, component-based applications

Types of Information Systems and Systems Development

- Transaction Processing Systems (TPS)
  - Automate handling of data about business activities (transactions)
  - Process orientation
- Management Information Systems (MIS)
  - Converts raw data from transaction processing system into meaningful form
  - Data orientation
- Decision Support Systems (DSS)
  - Designed to help decision makers
  - Provides interactive environment for decision making
  - Involves data warehouses, executive information systems (EIS)
  - Database, model base, user dialogue

Types of Information Systems and Systems Development (cont.)

Table 1-1 Systems Development for Different IS Types

IS Type	*IS Characteristics*	*Systems Development Methods*
Transaction processing system	High-volume, data capture focus; goal is efficiency of data movement and processing and interfacing different TPSs	Process orientation; concern with capturing, validating, and storing data and with moving data between each required step
Management information system	Draws on diverse yet predictable data resources to aggregate and summarize data; may involve forecasting future data from historical trends and business knowledge	Data orientation; concern with understanding relationships among data so data can be accessed and summarized in a variety of ways; builds a model of data that supports a variety of uses
Decision support system	Provides guidance in identifying problems, finding and evaluating alternative solutions, and selecting or comparing alternatives; potentially involves groups of decision makers; often involves semi-structured problems and the need to access data at different levels of detail	Data and decision logic orientations; design of user dialogue; group communication may also be key, and access to unpredictable data may be necessary; nature of systems requires iterative development and almost constant updating

Developing Information Systems and the SDLC

- System Development Methodology
  - Standard process followed in an organization
  - Consists of:
    - Analysis
    - Design
    - Implementation
    - Maintenance

Systems Development Life Cycle (SDLC)

- Traditional methodology for developing, maintaining, and replacing information systems
- Phases in SDLC:
  - Planning
  - Analysis
  - Design
  - Implementation
  - Maintenance

Standard and Evolutionary Views of SDLC

Figure 1-3 The systems development life cycle

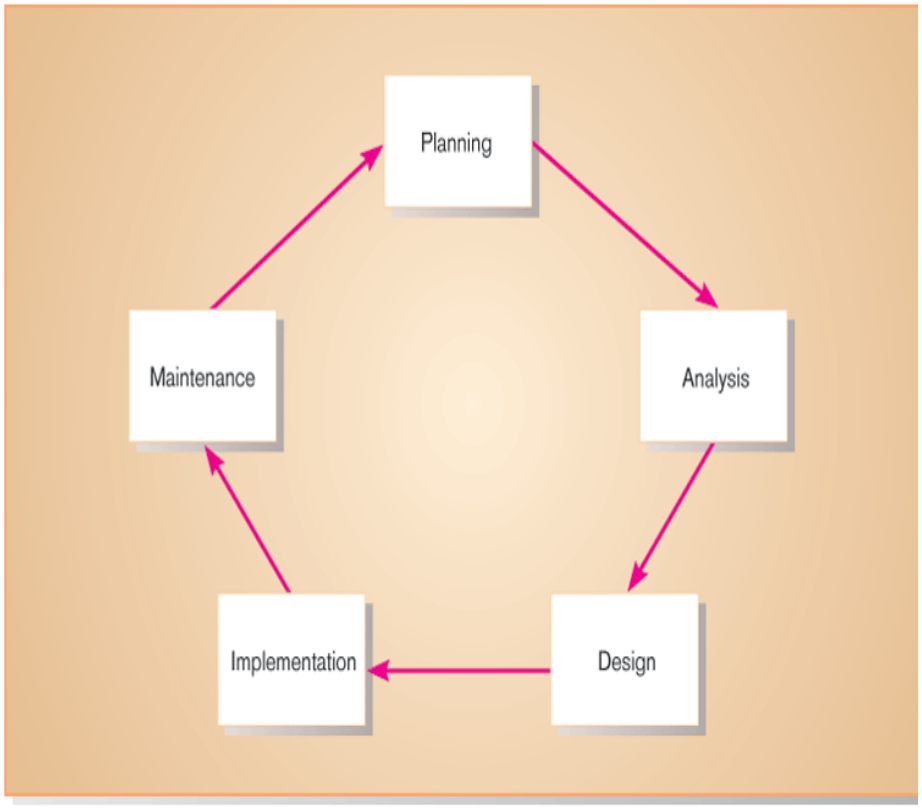
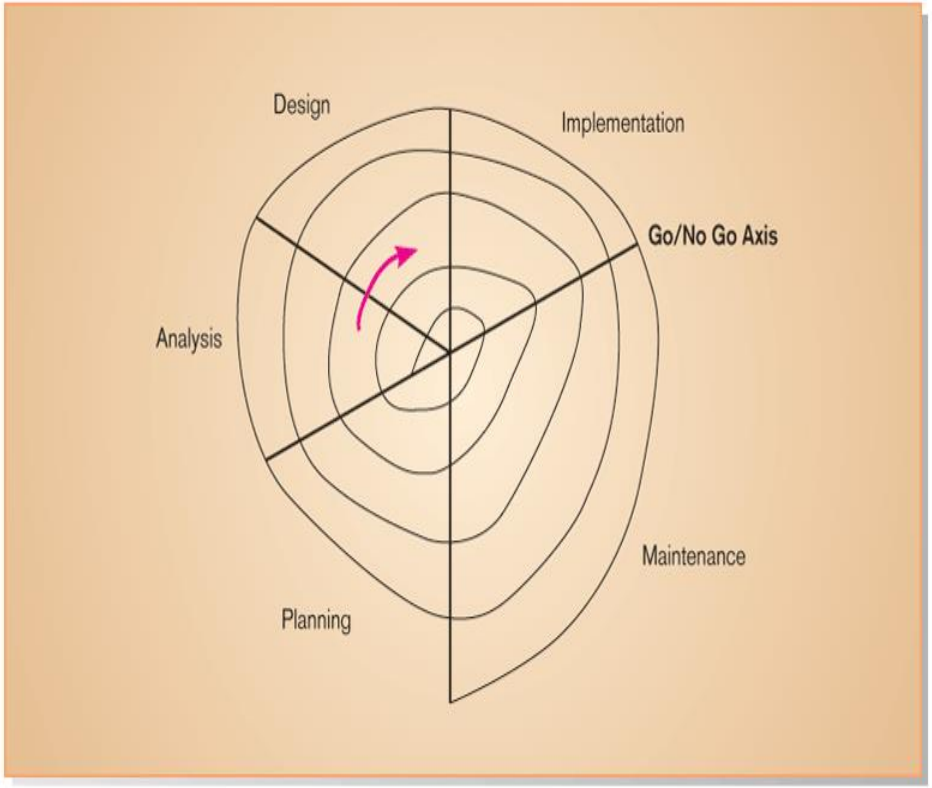
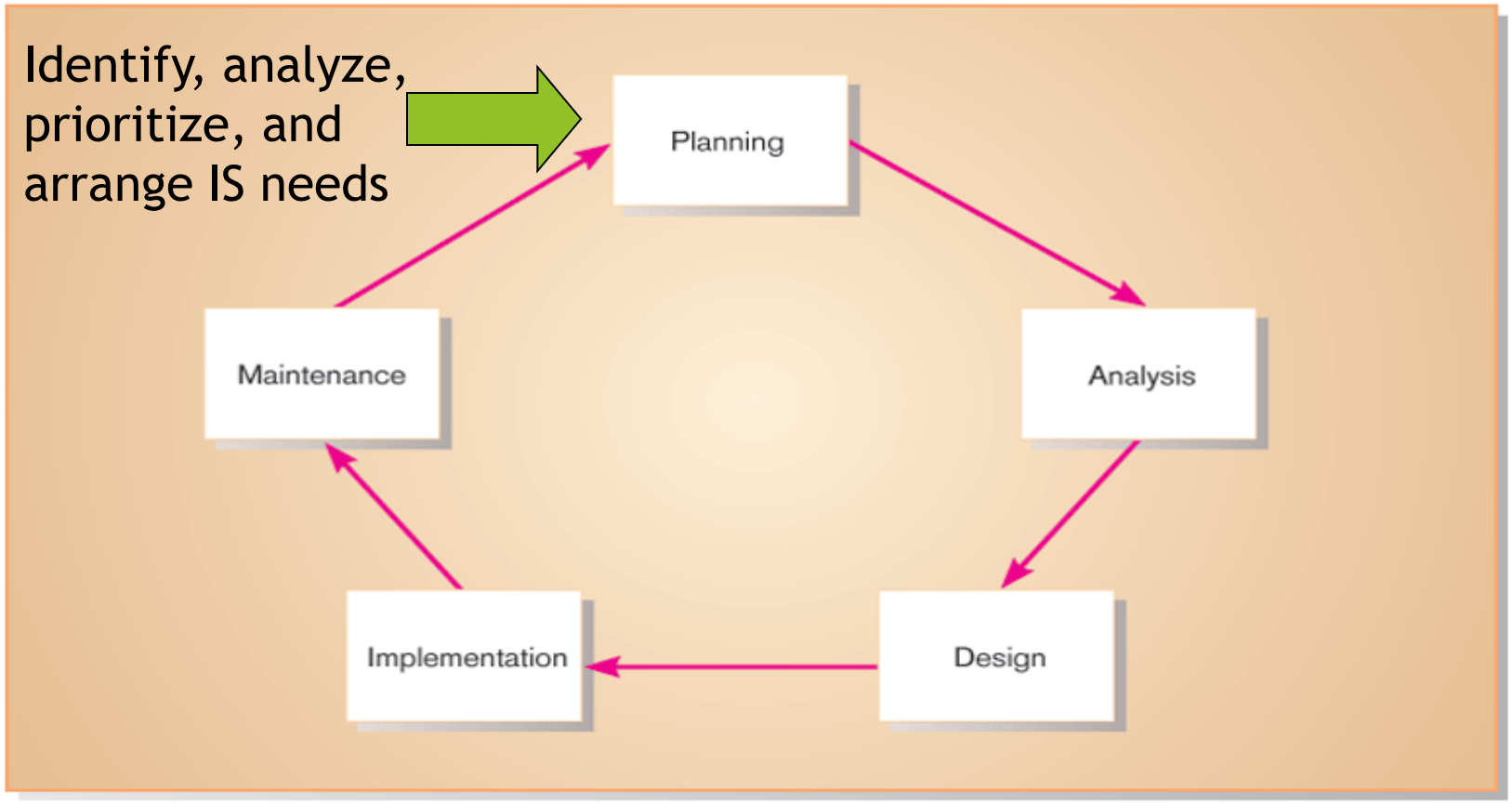


Figure 1-4 Evolutionary model SDLC



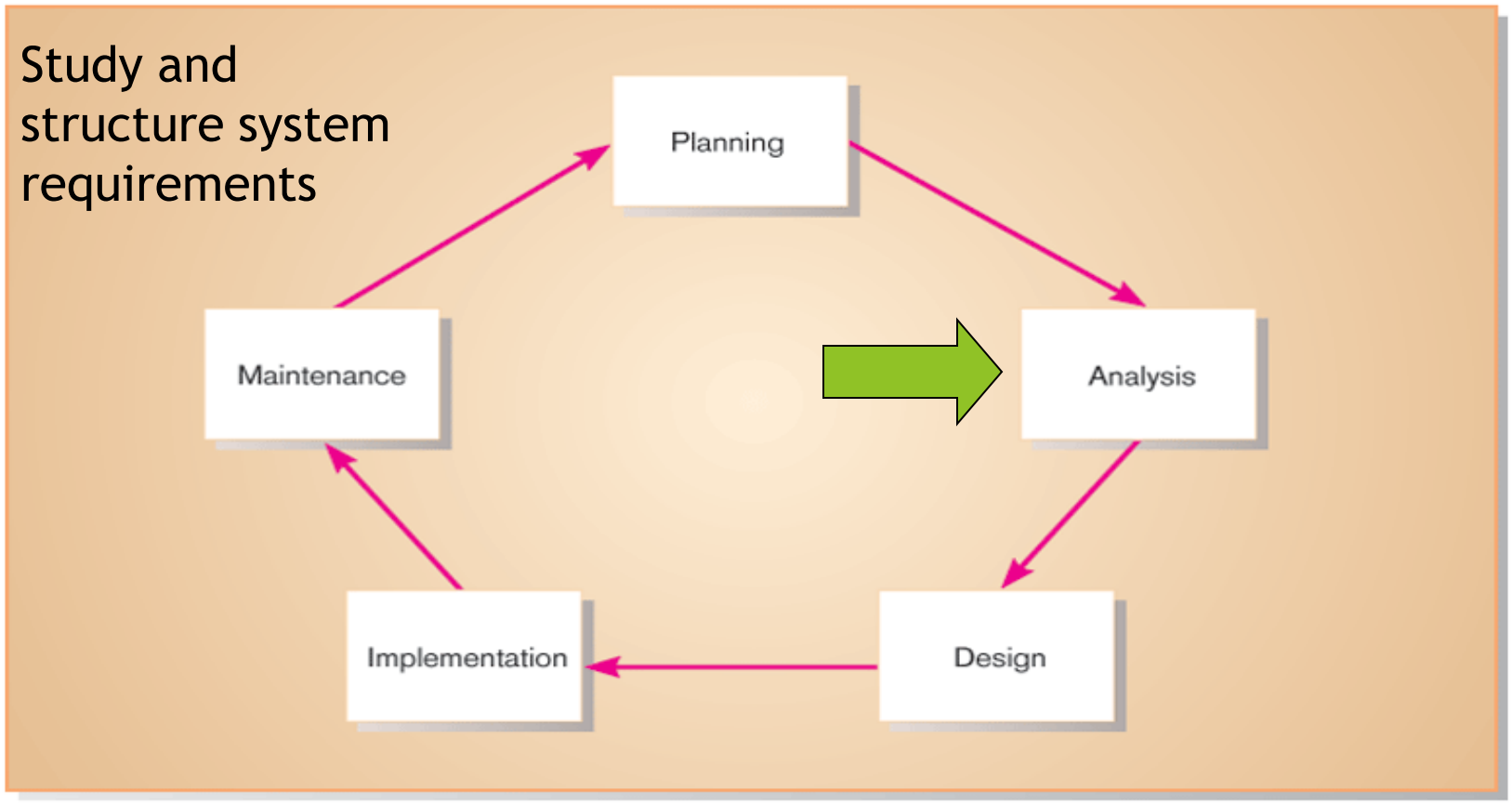
SDLC Planning Phase

Figure 1-3 The systems development life cycle



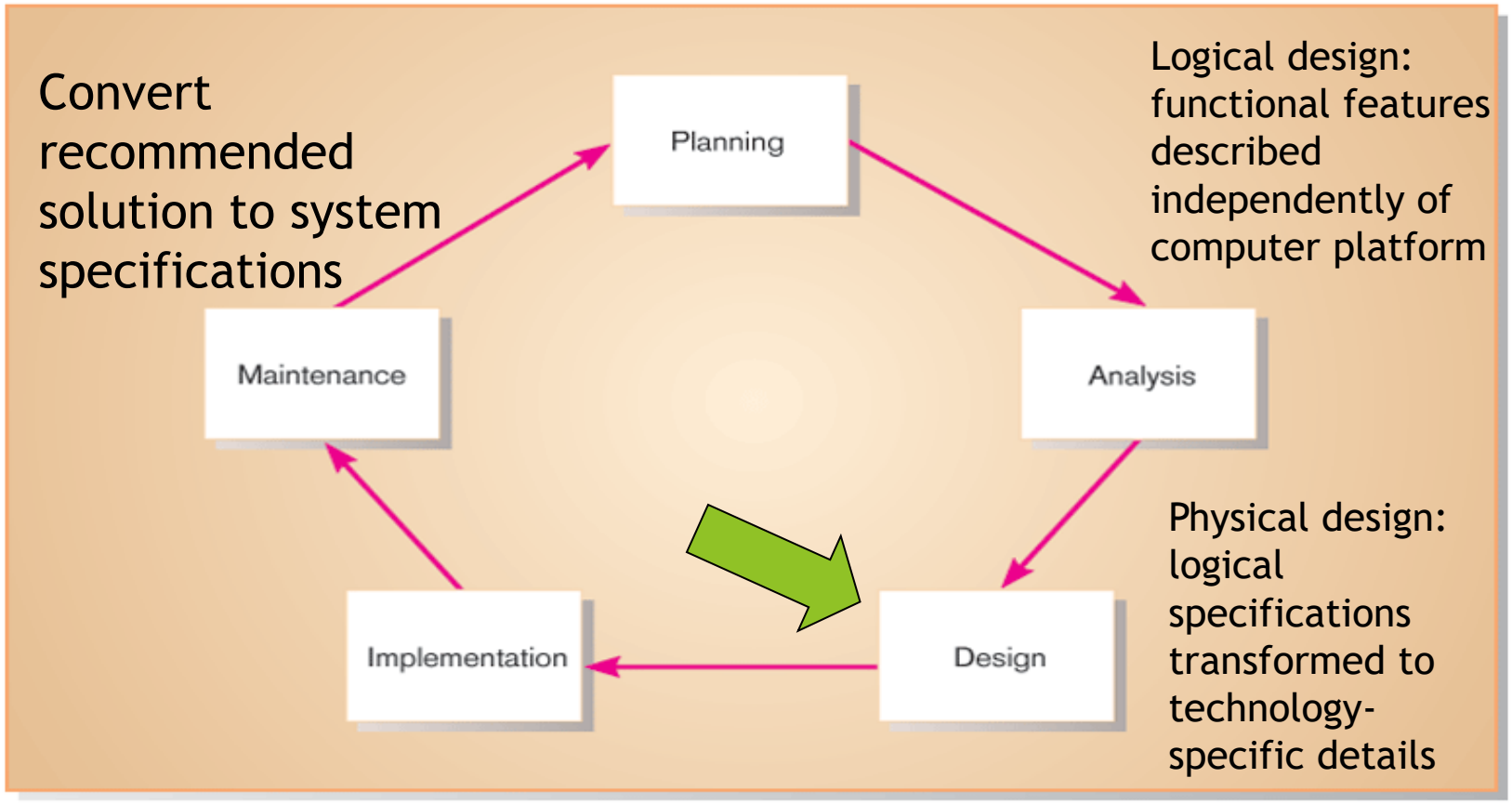
SDLC Analysis Phase

Figure 1-3 The systems development life cycle



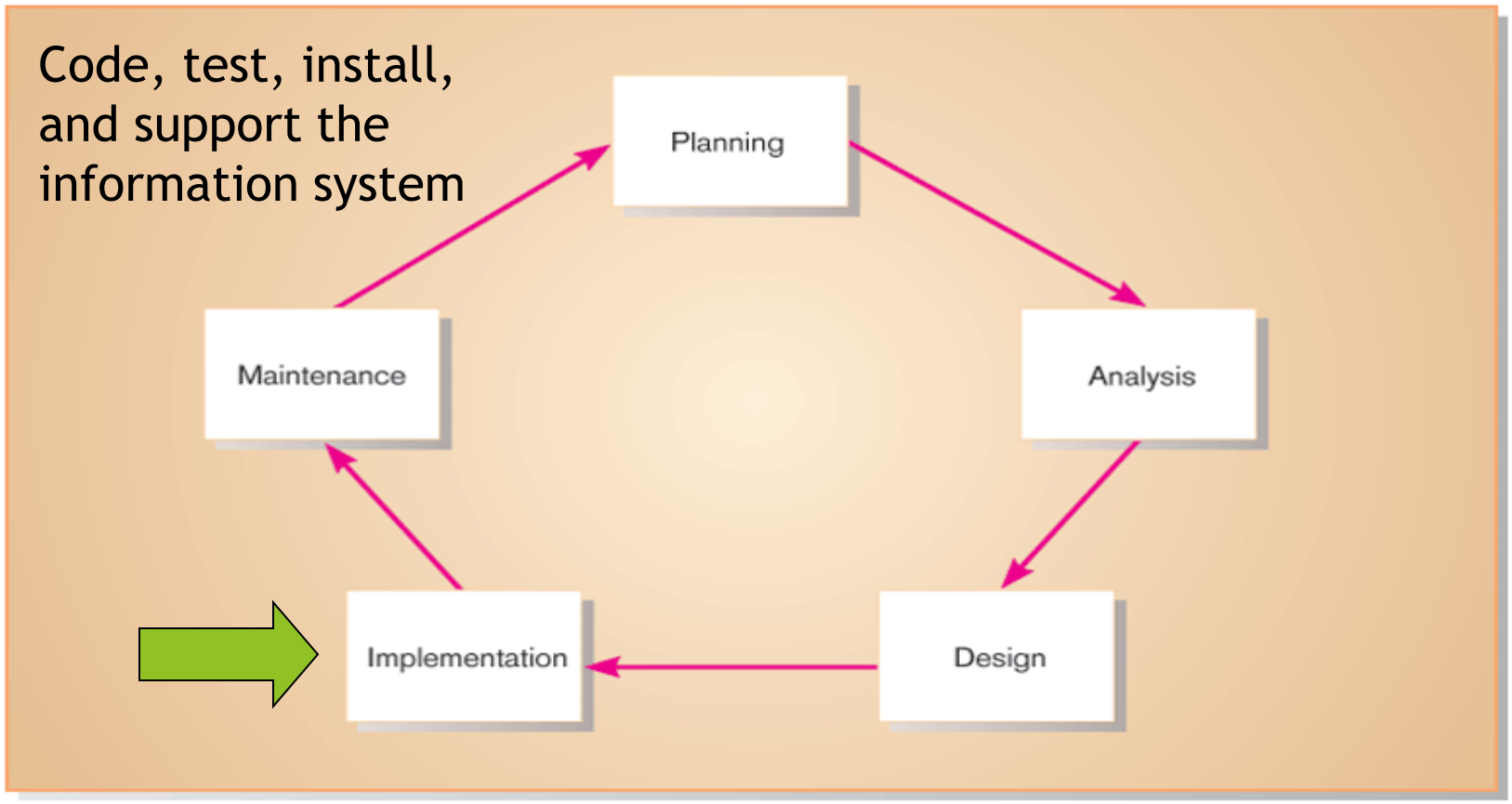
SDLC Design Phase

Figure 1-3 The systems development life cycle



SDLC Implementation Phase

Figure 1-3 The systems development life cycle



SDLC Maintenance Phase

Figure 1-3 The systems development life cycle

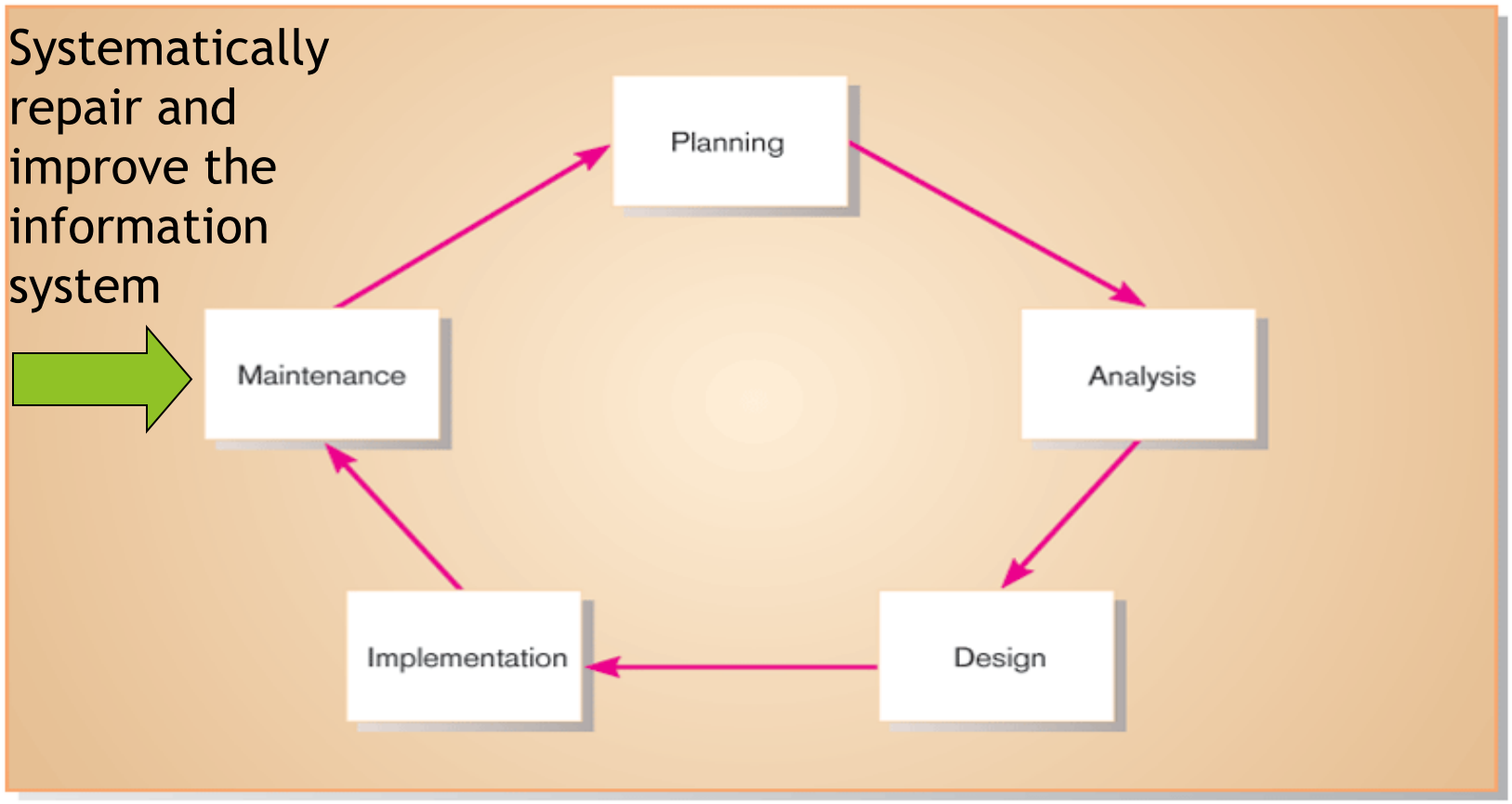


Table 1-2 Products of SDLC Phases

Phase	*Products, Outputs, or Deliverables*
Planning	Priorities for systems and projects; an architecture for data, networks, and selection hardware, and IS management are the result of associated systems;
	Detailed steps, or work plan, for project;
	Specification of system scope and planning and high-level system requirements or features;
	Assignment of team members and other resources;
	System justification or business case
Analysis	Description of current system and where problems or opportunities are with a general recommendation on how to fix, enhance, or replace current system;
	Explanation of alternative systems and justification for chosen alternative
Design	Functional, detailed specifications of all system elements (data, processes, inputs, and outputs);
	Technical, detailed specifications of all system elements (programs, files, network, system software, etc.);
	Acquisition plan for new technology
Implementation	Code, documentation, training procedures, and support capabilities
Maintenance	New versions or releases of software with associated updates to documentation, training, and support

The Heart of the Systems Development Process

Figure 1-8 The analysis–design–code–test loop

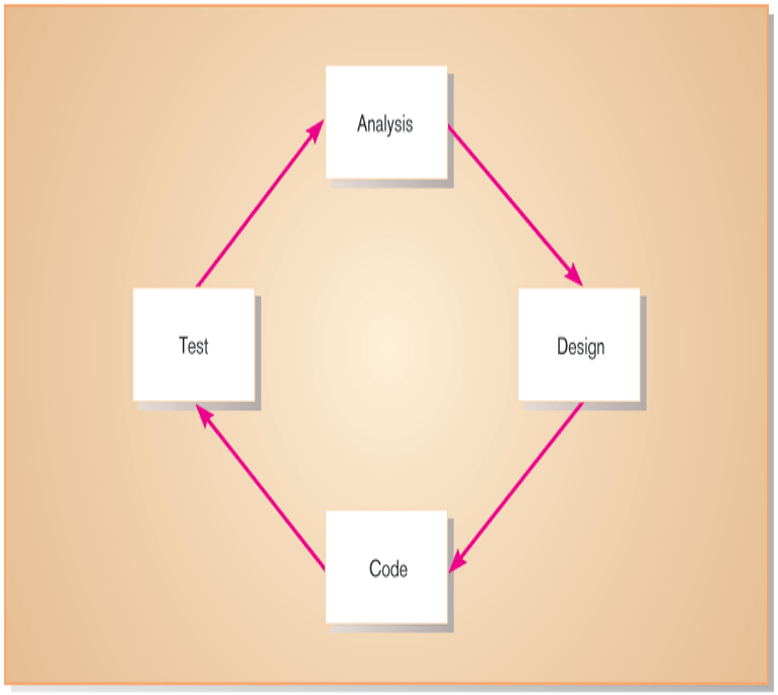
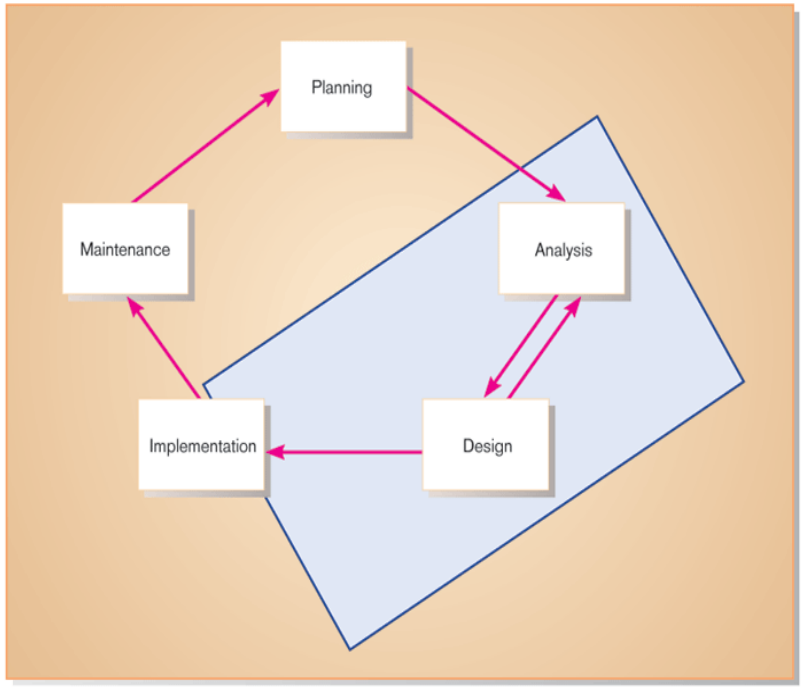


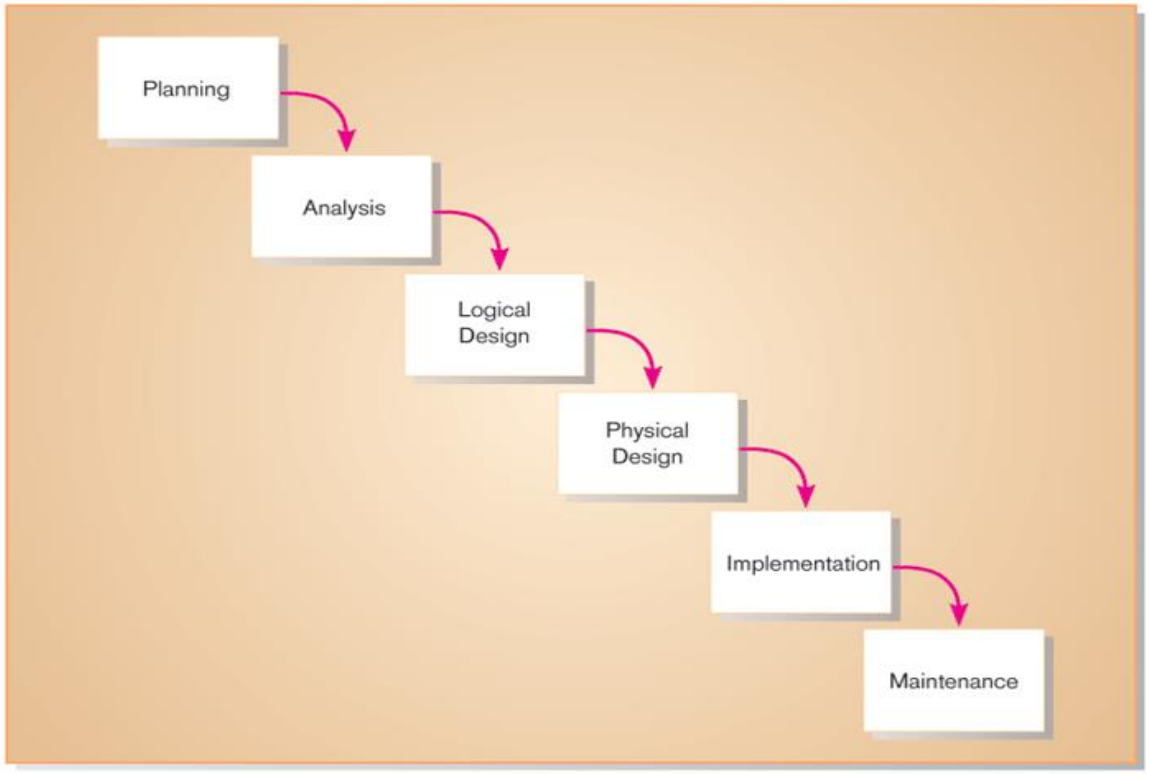
Figure 1-9 The heart of systems development



Current practice combines analysis, design, and implementation into a single iterative and parallel process of activities

Traditional Waterfall SDLC

Figure 1-10 A traditional waterfall SDLC



One phase begins when another completes, little backtracking and looping

Problems with Waterfall Approach

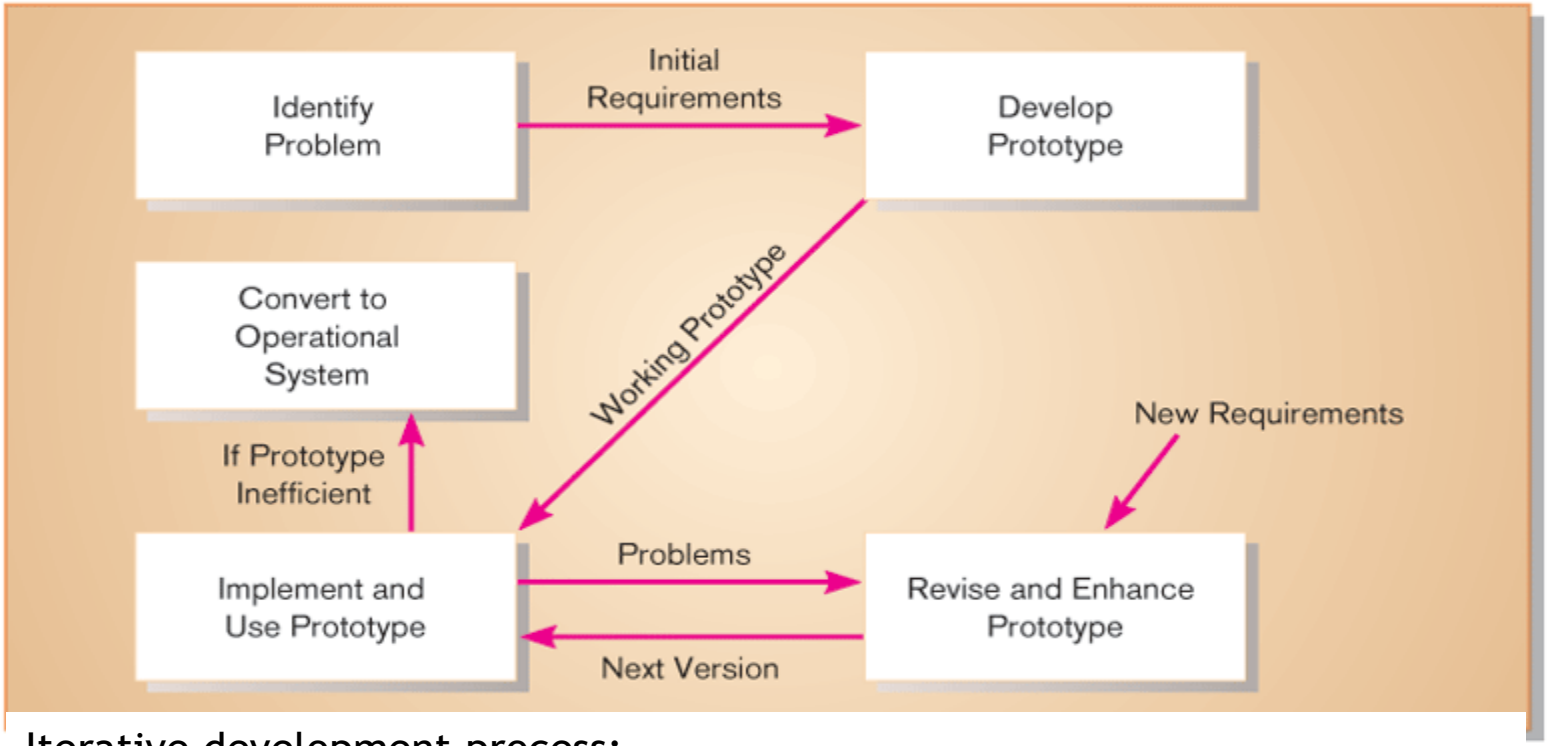
- System requirements “locked in” after being determined (can't change)
- Limited user involvement (only in requirements phase)
- Too much focus on milestone deadlines of SDLC phases to the detriment of sound development practices

Alternatives to Traditional Waterfall SDLC

- Prototyping
- CASE tools
- Joint Application Design (JAD)
- Rapid Application Development (RAD)
- Agile Methodologies
- eXtreme Programming

Prototyping

Figure 1-11 The prototyping methodology



Iterative development process:
Requirements quickly converted to a working system
System is continually revised
Close collaboration between users and analysts

CASE Tools

- Computer-Aided Software Engineering
- Software tools providing automated support for systems development
- Project dictionary/workbook: system description and specifications
- Diagramming tools
- Example products: Oracle Designer, Rational Rose

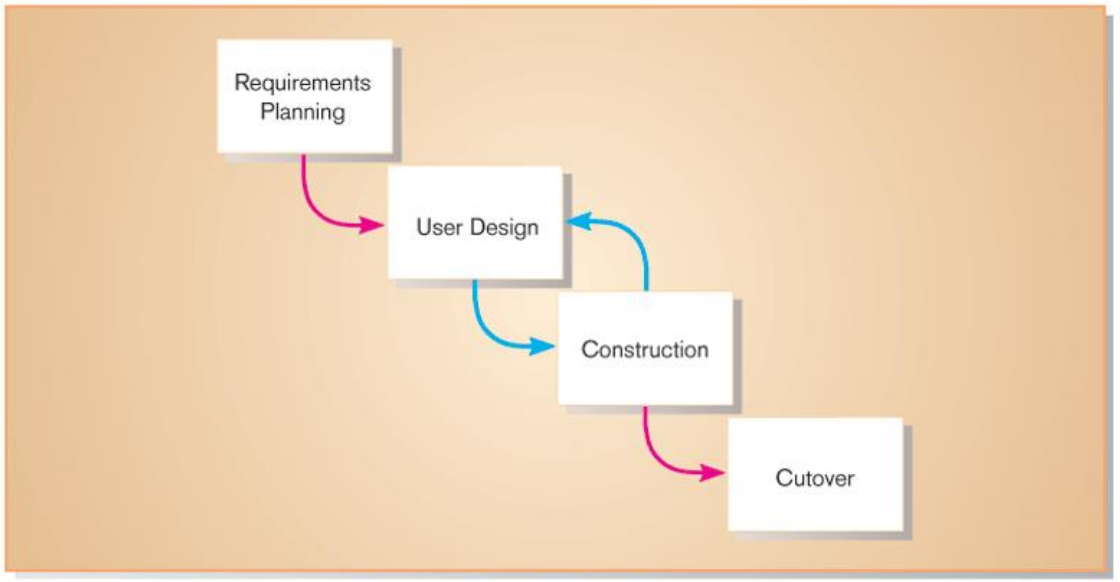
Joint Application Design (JAD)

- Structured process involving users, analysts, and managers
- Several-day intensive workgroup sessions
- Purpose: to specify or review system requirements

Rapid Application Development (RAD)

- Methodology to decrease design and implementation time
- Involves: prototyping, JAD, CASE tools, and code generators

Figure 1-12 RAD life cycle



Agile Methodologies

- Motivated by recognition of software development as fluid, unpredictable, and dynamic
- Three key principles
  - Adaptive rather than predictive
  - Emphasize people rather than roles
  - Self-adaptive processes

eXtreme Programming

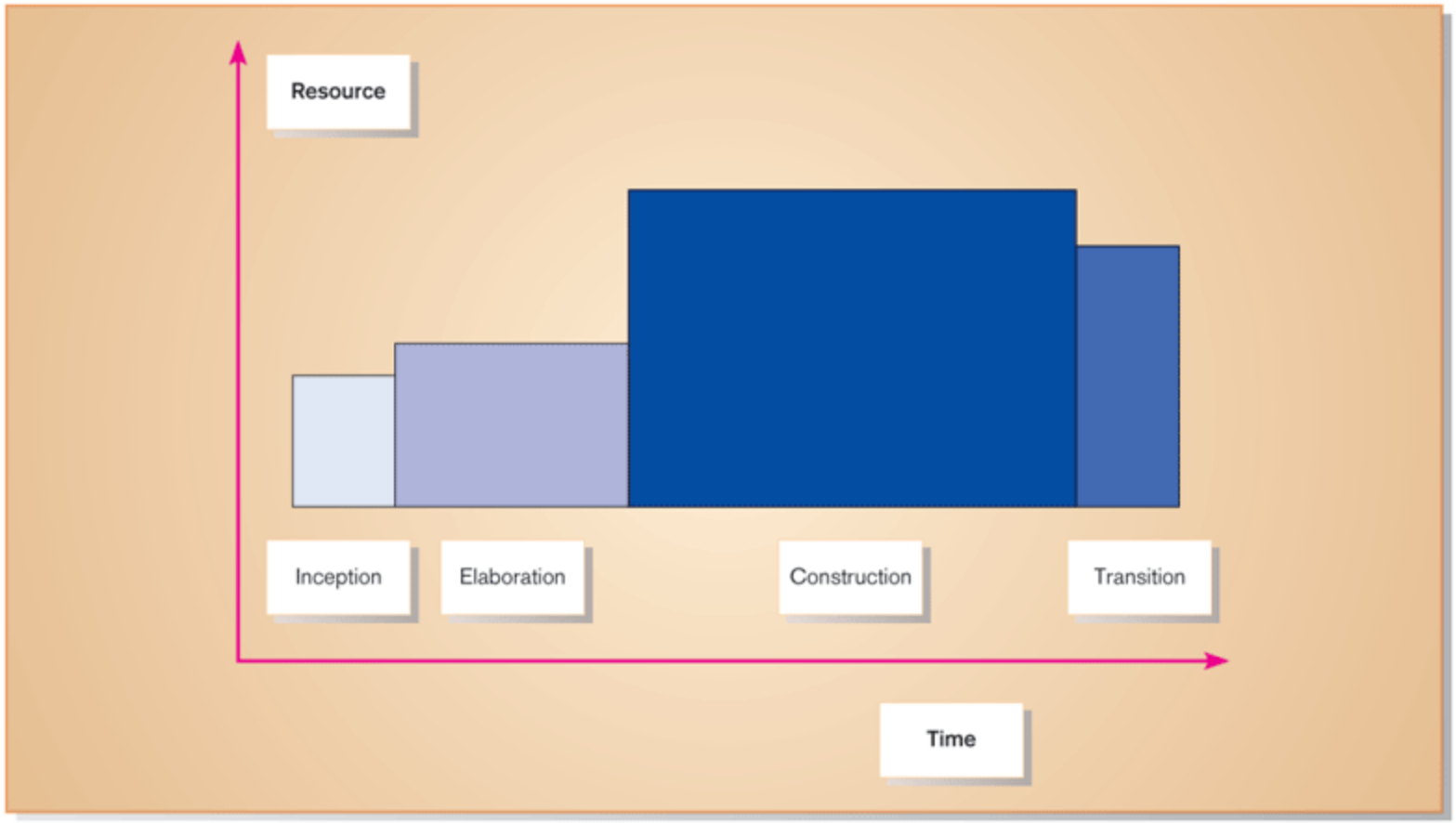
- Short, incremental development cycles
- Automated tests
- Two-person programming teams
- Coding and testing operate together
- Advantages:
  - Communication between developers
  - High level of productivity
  - High-quality code

Object-Oriented Analysis and Design

- Based on objects rather than data or processes
- Object: a structure encapsulating attributes and behaviors of a real-world entity
- Object class: a logical grouping of objects sharing the same attributes and behaviors
- Inheritance: hierarchical arrangement of classes enable subclasses to inherit properties of superclasses

Rational Unified Process (RUP) involves an iterative, incremental approach to systems development

Figure 1-13 Phases of OOSAD-based development



Summary

- In this chapter you learned how to:
  - ✓ Define information systems analysis and design.
  - ✓ Describe the different types of information systems.
  - ✓ Describe the information Systems Development Life Cycle (SDLC).
  - ✓ Explain Rapid Application Development (RAD), prototyping, Joint Application Development (JAD), and Computer Aided Software Engineering (CASE).
  - ✓ Describe agile methodologies and eXtreme programming.
  - ✓ Explain Object Oriented Analysis and Design and the Rational Unified Process (RUP).