


- 
- **Systems Analysis and Design**
 - **Course Code: CSE 305**
 - **Fall 2023 Semester**

Instructor:
Dr. Shah Murtaza Rashid Al Masud
Assoc. Prof. CSE Dept., UAP



Learning Resources

Textbook	▪ Modern System Analysis & Design - Jeffrey Hpffer, Joey George, Joseph Valacich, 6 th edition, Pearson, (Available in Prince Mishal Library)
Other References	▪ “Systems Analysis and Design” , (latest edition), Kendall & Kendall, Prentice-Hall

Evaluation Scheme

Components	Weight
Class Test	30%
Midterm	20%
Final Examination	50%

Course Summary

- Introduction and SDLC
- System Development in an Organizational Context
- Managing the Information System Projects
- Determining System Requirements
- Analyzing System Process Requirements
- Object Oriented Analysis and Design: Use cases
- Object Oriented Analysis and Design: Activity Diagrams
- Object Oriented Analysis and Design: Sequence Diagrams
- Object Modeling: Class Diagrams
- Designing Database, Forms & Reports Design
- Systems Repository, Implementing and Maintaining the System

Introduction

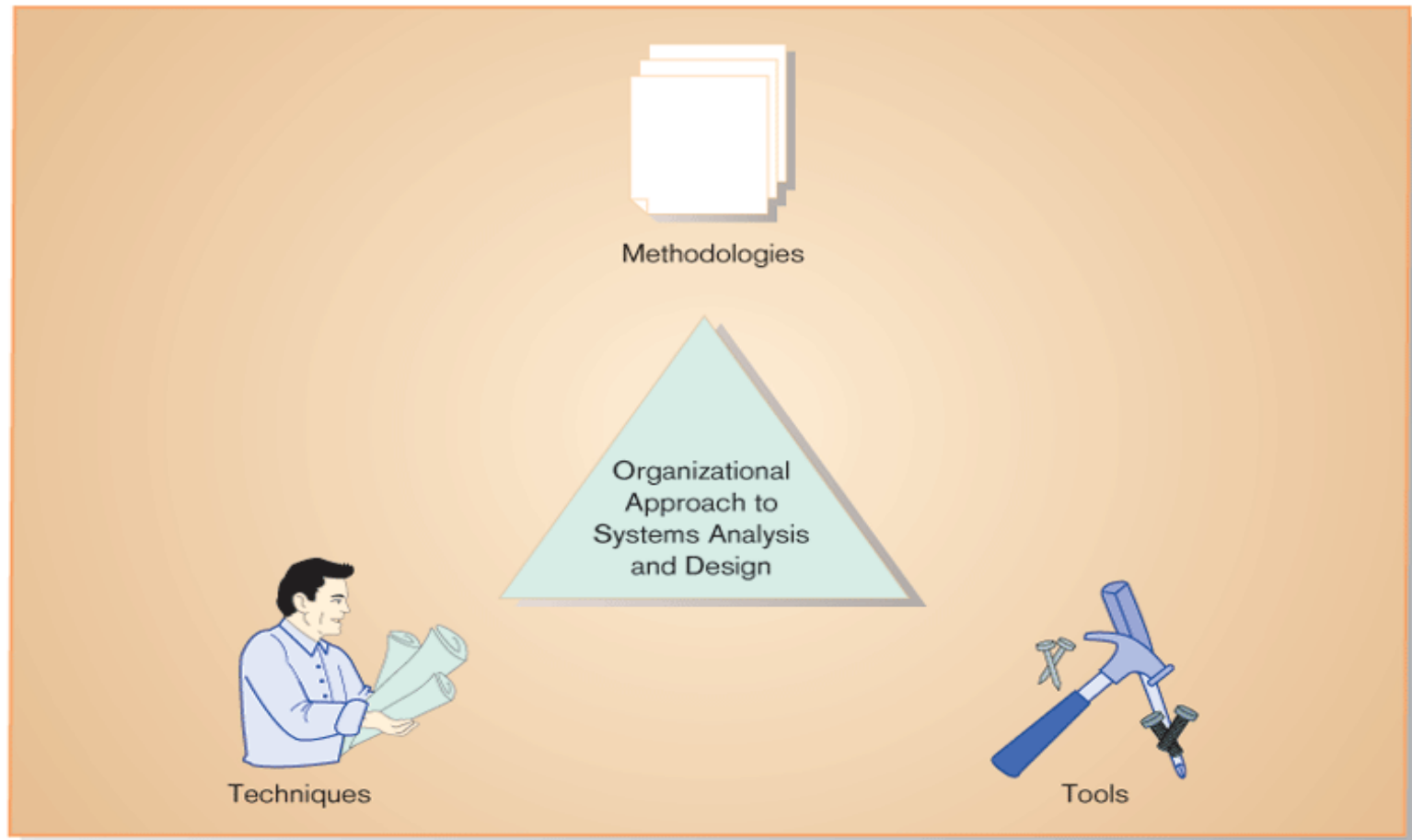
Systems Analysis and Design (SAD) is a **process of**
planning and developing high quality and new business
Information Systems (IS) which
combines Information Technology (IT),
People and
Data to support business requirements.

Introduction

- **Information Systems Analysis and Design:** Complex organizational/business process whereby computer-based information systems are developed and maintained.
- **Application Software:** Computer software designed to support organizational functions or processes.
- **Systems Analyst:** the people/person who Performs Organizational role the 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.



Information—A Key Resource

- Fuels business and can be the critical/key factor in determining the success or failure of a business
- Information needs to be managed correctly, carefully
- Managing computer-generated information differs from handling manually produced data

Roles of the Systems Analyst

- The analyst must be able to work with people of all descriptions and be experienced in working with computers.
- Three primary roles:
 - Consultant
 - Supporting expert
 - Agent of change

Qualities of the Systems Analyst

- Problem solver
- Communicator
- Strong personal and professional ethics
- Self-disciplined and self-motivated

Systems Development Life Cycle (SDLC) (Prof. Hoffer)

- SDLC is the 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

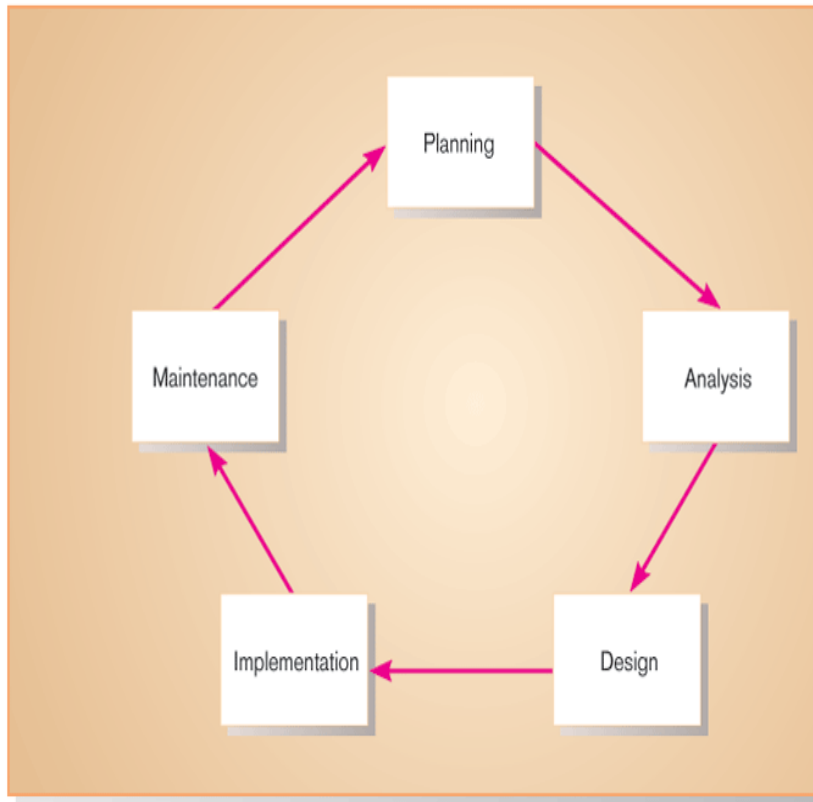
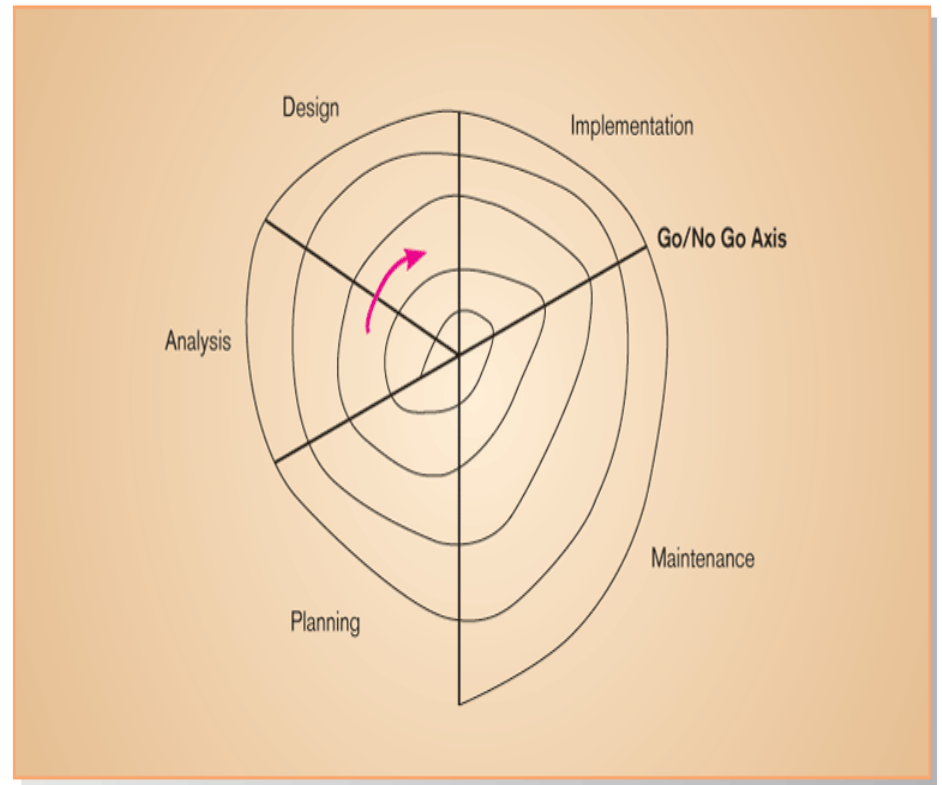
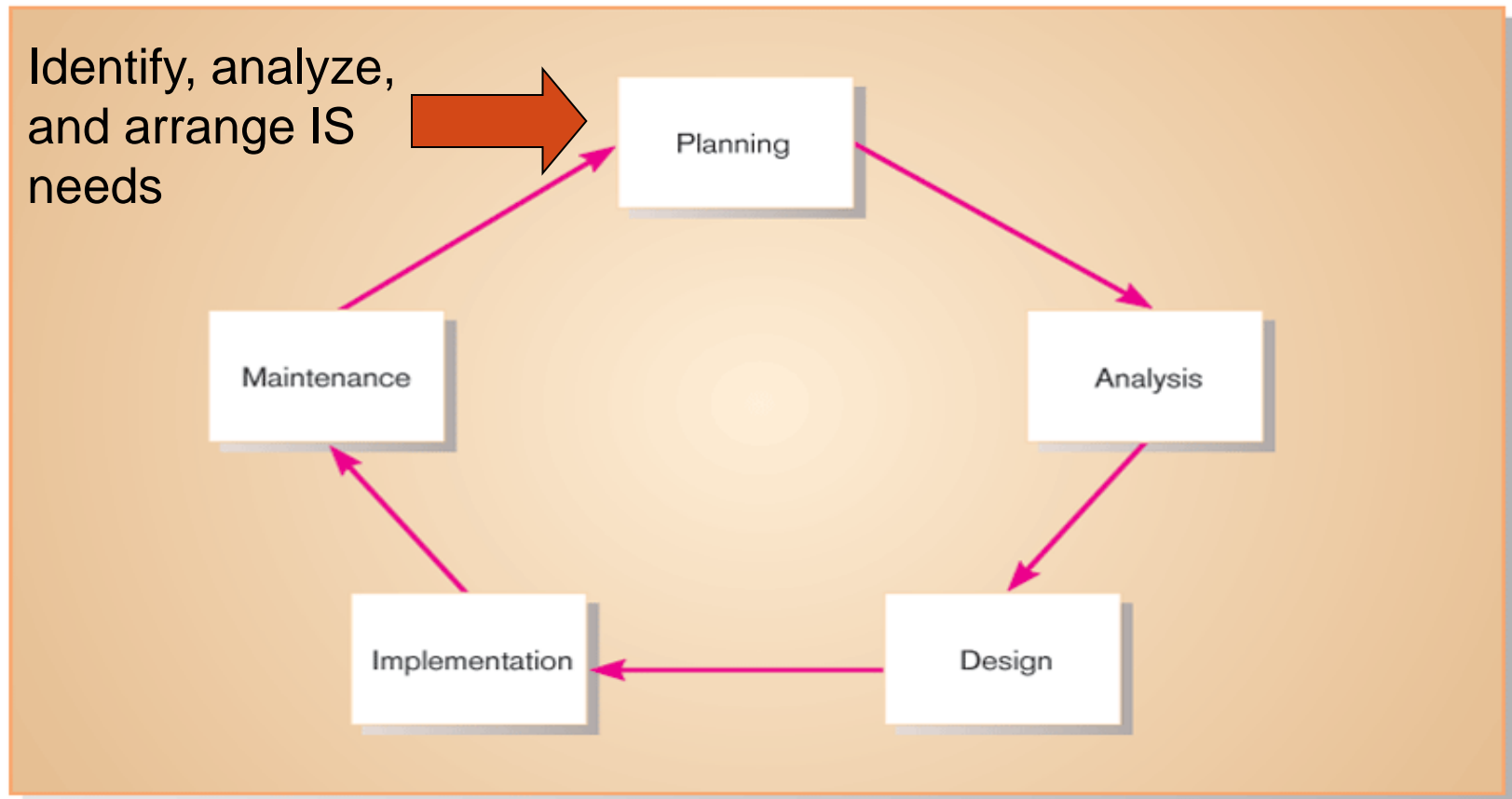


Figure 1-4 Evolutionary model SDLC



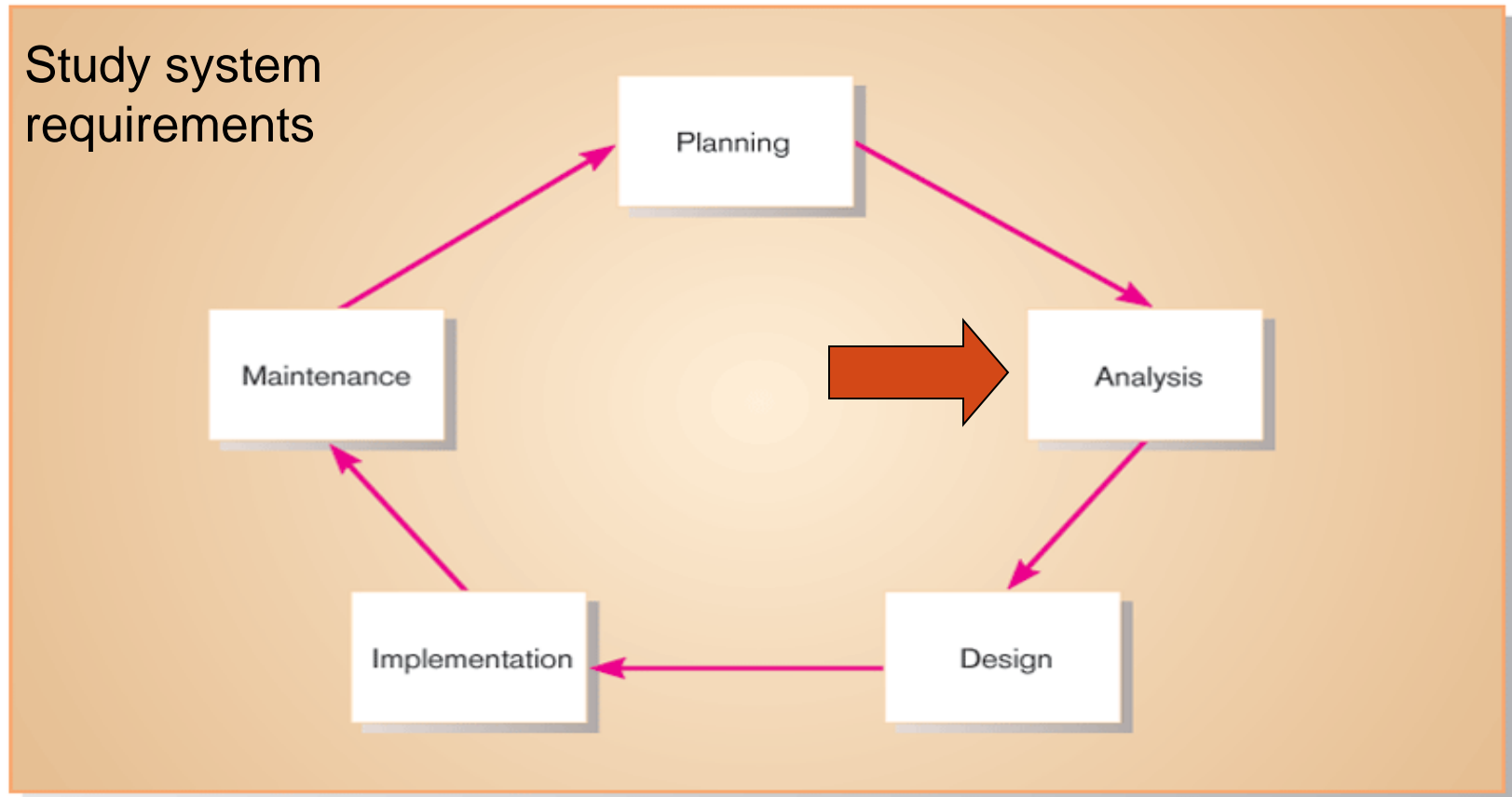
SDLC Planning Phase

Figure 1-3 The systems development life cycle



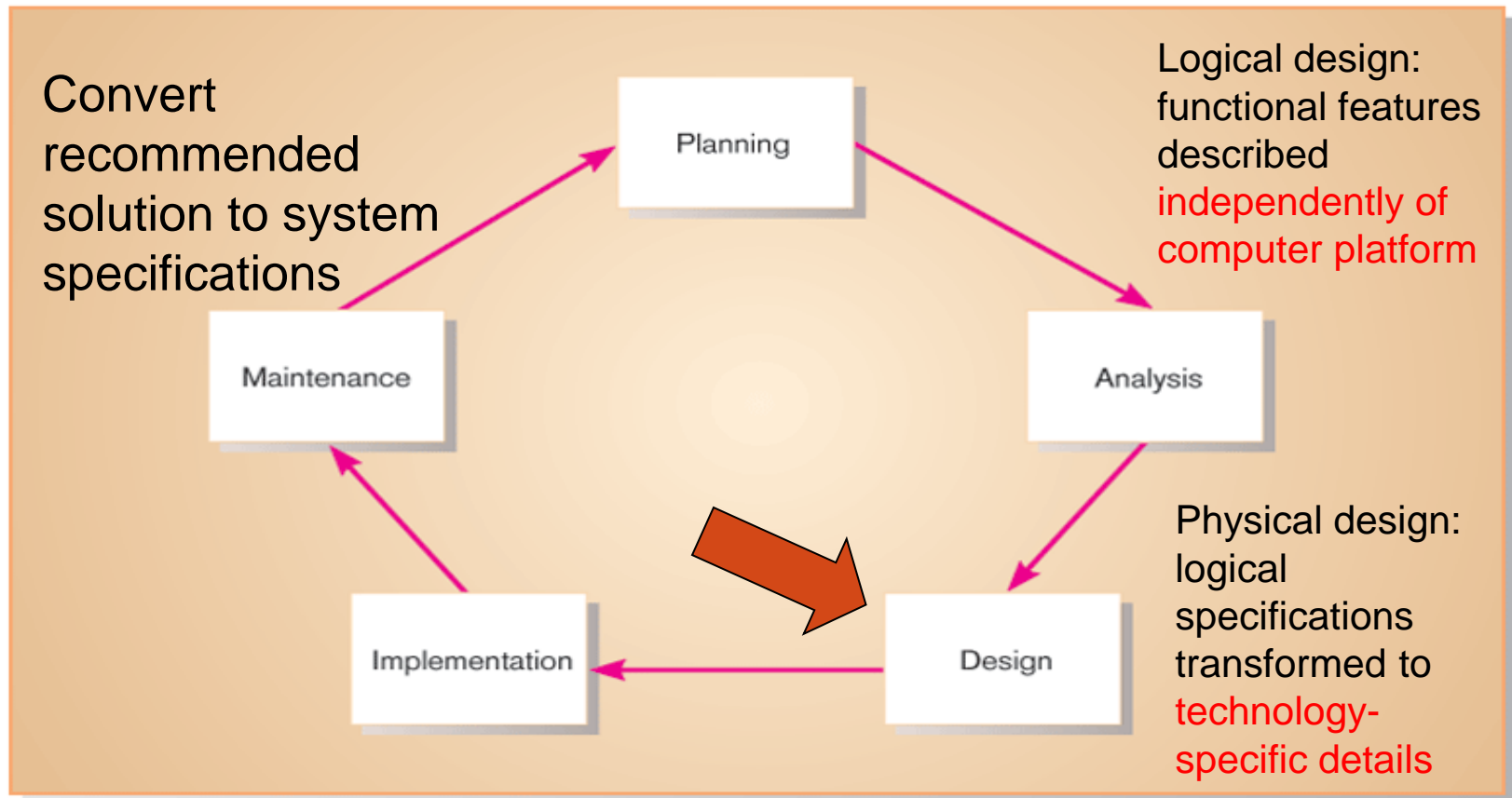
SDLC Analysis Phase

Figure 1-3 The systems development life cycle



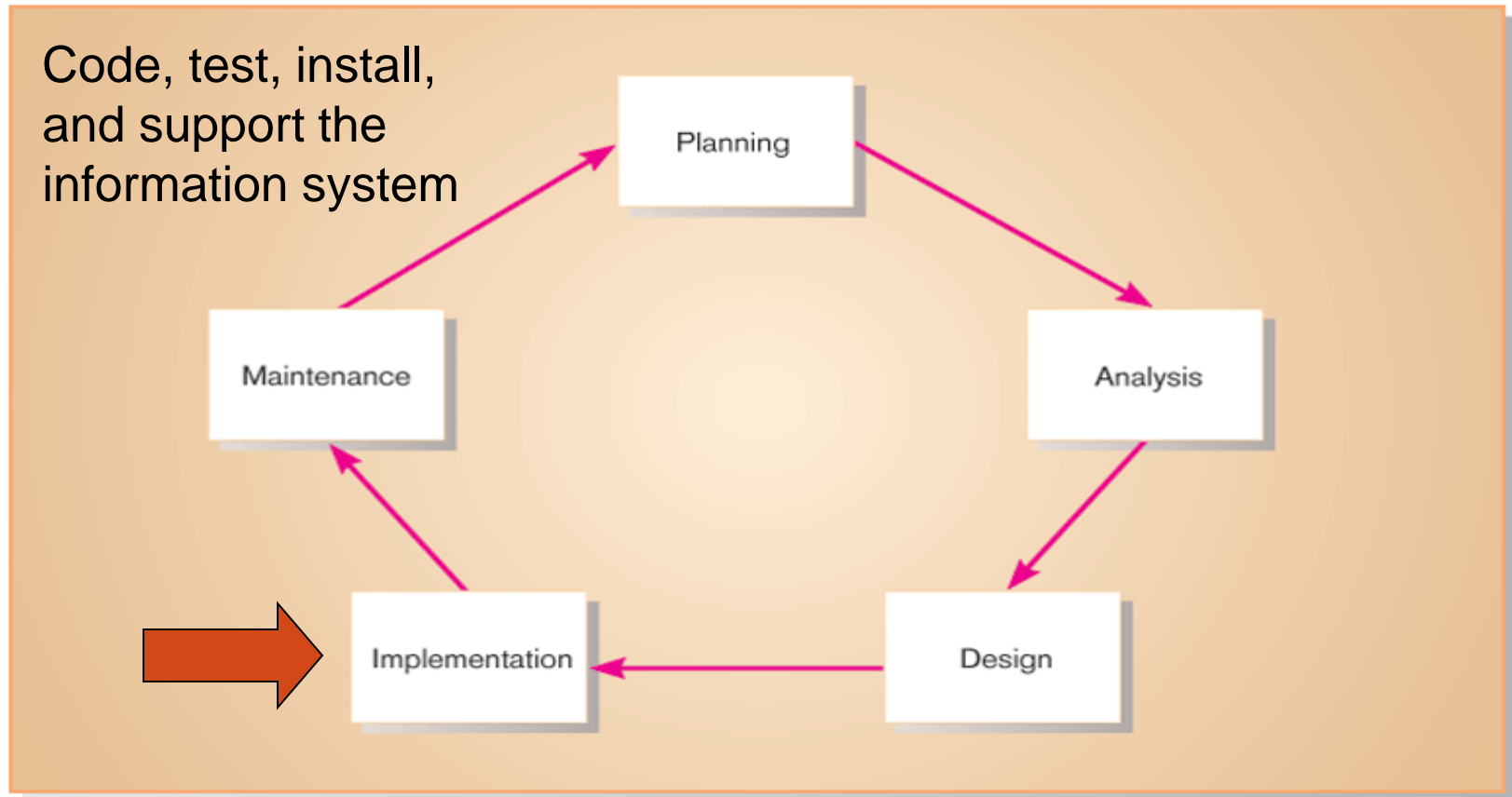
SDLC Design Phase

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

Systematically
repair and improve
the information
system

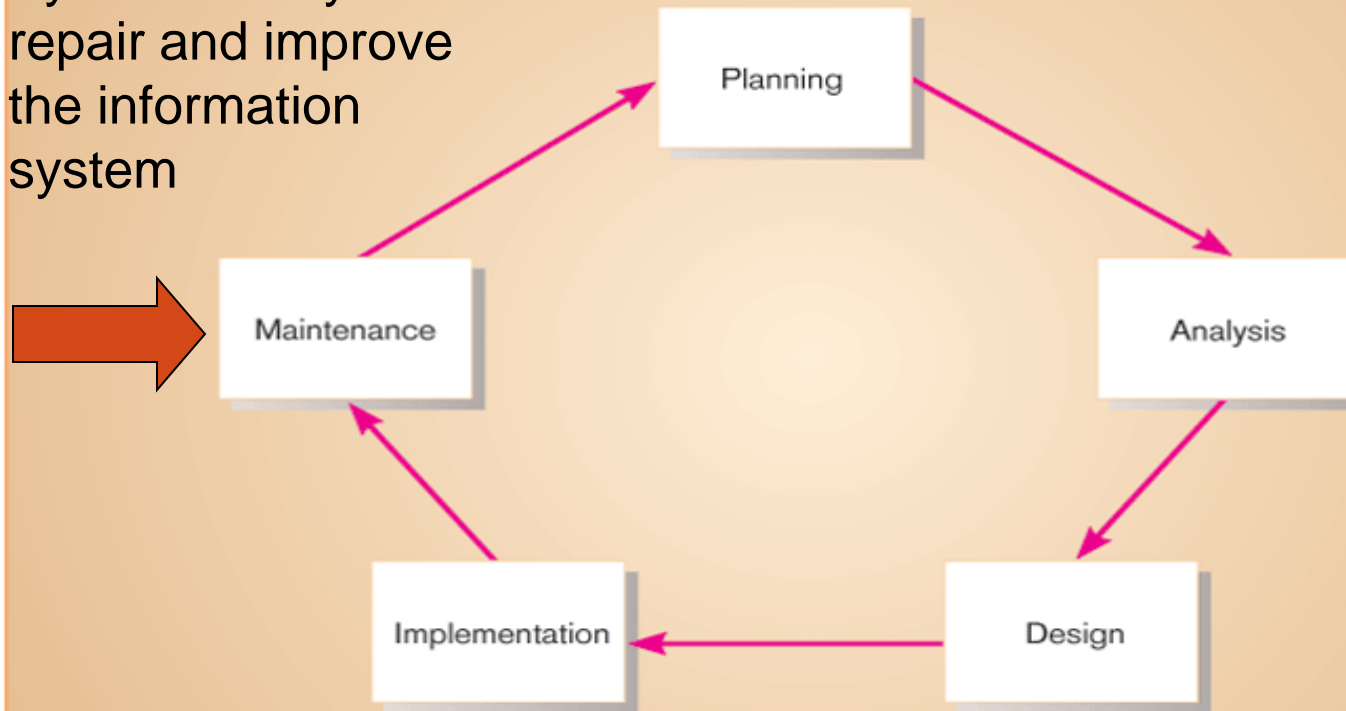


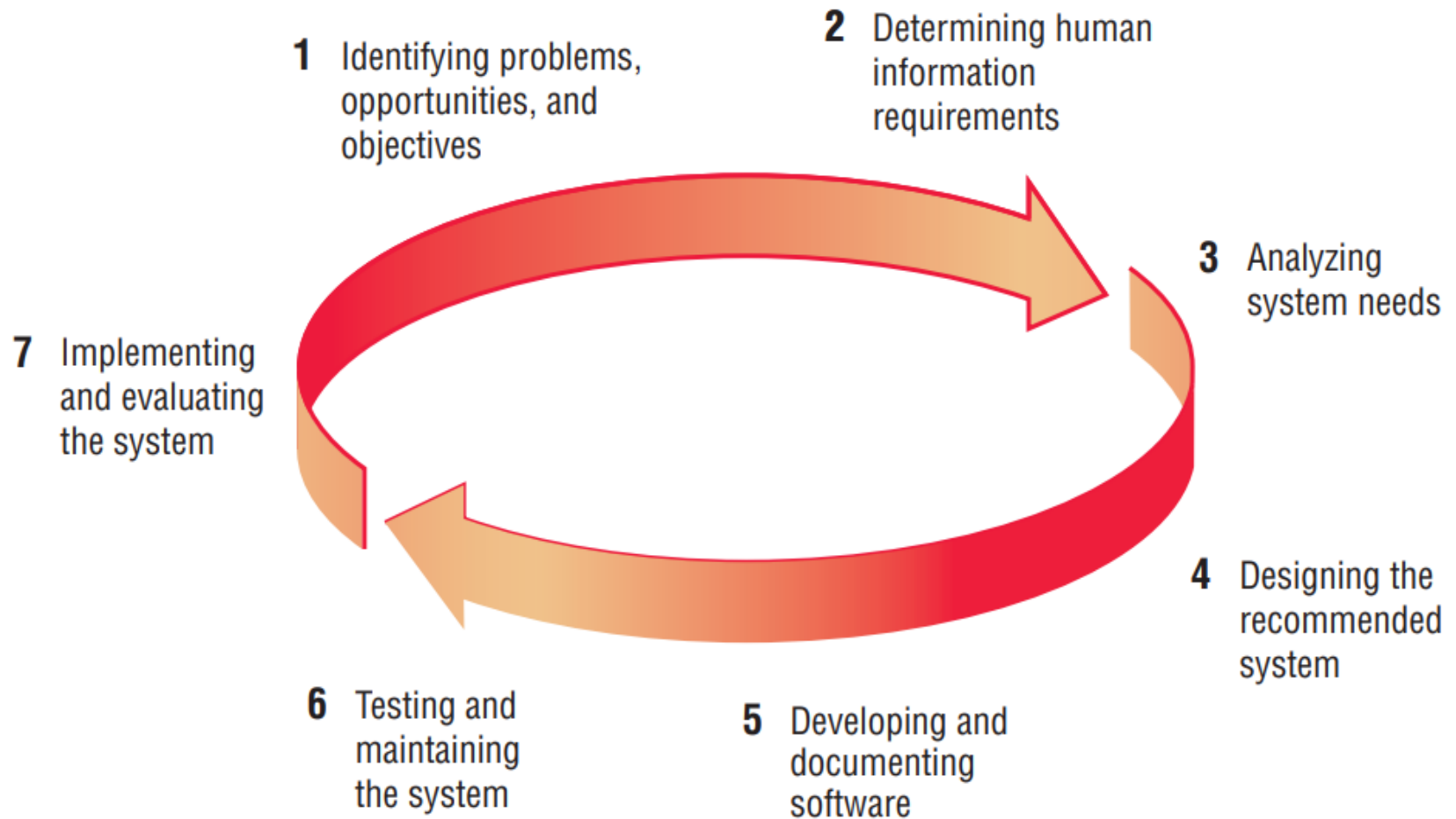
Table 1-2 Products of SDLC Phases

<i>Phase</i>	<i>Products, Outputs, or Deliverables</i>
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

Systems Development Life Cycle (SDLC) (Prof. Kendall)

- Traditional methodology for developing, maintaining, and replacing information systems
- **Phases in SDLC:**
 - Identifying problems
 - Determining requirements
 - Analysis systems need
 - Design the systems
 - Development and documentation
 - Testing and maintenance
 - Implementation

Systems Development Life Cycle (SDLC) (Prof. Kendall)



Systems Development Life Cycle (SDLC) (Prof. Kendall)

Identifying Problems, Opportunities, and Objectives

In this first phase of the systems development life cycle, the analyst is concerned with **correctly identifying problems, opportunities, and objectives**. This stage is critical to the **success of the rest of the project**, because no one wants to waste subsequent time addressing the wrong problem.

Systems Development Life Cycle (SDLC) (Prof. Kendall)

The people involved in the first phase are the users, analysts, and systems managers coordinating the project.

Systems Development Life Cycle (SDLC) (Prof. Kendall)

Determining Requirements

In the information **requirements phase** of the SDLC, the systems analyst determines **what information users need to perform their jobs.**

At this point the analyst is examining how to make the system useful to the people involved.

Systems Development Life Cycle (SDLC) (Prof. Kendall)

Determining Requirements

The **people involved in this phase** are the analysts and users, typically operations managers and operations workers.

The **systems analyst needs to know the details of current system functions:**

who (the people who are involved),

what (the business activity),

where (the environment in which the work takes place),

when (the timing), and

how (how the current procedures are performed)

why the business uses the current system.

There may be good reasons for doing business using the **current methods**, and these should

be considered when designing any **new system**.

Systems Development Life Cycle (SDLC) (Prof. Kendall)

Systems Analysis

At this point in the SDLC, the systems analyst prepares a systems proposal that summarizes what has been found out about the users, usability, and usefulness of current systems; provides cost-benefit analyses of alternatives; and makes recommendations on what (if anything) should be done. If one of the recommendations is acceptable to management, the analyst proceeds along that course. Each systems problem is unique, and there is never just one correct solution.

Systems Development Life Cycle (SDLC) (Prof. Kendall)

Designing the Systems

In the design phase of the SDLC, the **systems analyst uses the information collected earlier to accomplish the design** of the information system.

The analyst designs procedures for users to help them accurately enter data so that data going into the information system are correct.

In addition, the analyst provides for users to complete effective input to the information system by using techniques of good form and Web page or screen design.

Systems Development Life Cycle (SDLC) (Prof. Kendall)

Developing and Documenting Software

In the fifth phase of the SDLC, the analyst works with programmers to develop any original software that is needed. During this phase the analyst works with users to develop effective documentation for software, including procedure manuals, online help, and Web sites featuring Frequently Asked Questions (FAQs), on Read Me files shipped with new software.

Systems Development Life Cycle (SDLC) (Prof. Kendall)

Testing and Maintaining the System

Before the information system can be used, **it must be tested**. It is much less costly to catch problems before the system is signed over to users.

Maintenance of the system and its documentation begins in this phase and is carried out routinely throughout the life of the information system.

Systems Development Life Cycle (SDLC) (Prof. Kendall)

Implementing and Evaluating the System

In this last phase of systems development, the analyst helps implement the information system.

This phase involves **training users** to handle the system.

In addition, the analyst needs to **plan for a smooth conversion from the old system to the new one**. This process includes

converting files from old formats to new ones, or building a database, installing equipment, and bringing the new system into production

Systems Development Life Cycle (SDLC) (Prof. Kendall)

The systems development life cycle : is a systematic approach to solving business problems .

Reasons for enhancement the system :

- Adding additional features.
- Requirements change over time
- Technology H\W and S\W change rapidly.

Alternatives to Traditional SDLC

- CASE tools
- Agile Methodologies
- Object oriented method

CASE Tools

- Computer-Aided Software Engineering
- Software tools providing automated support for systems development
- **Case tools** : a software package for system analysis and design .
- **Reasons for using case tools** :
 - Increase analyst productivity .
 - Facilitate communication.
 - Continuity between phases .
 - Assess the impact of maintenance.

Case Tool Classifications

Upper CASE tools perform analysis and design.

Lower CASE tools generate programs from CASE design.

Upper CASE Tools

Create and modify the system design.

Help in modeling organizational requirements and defining system boundaries.

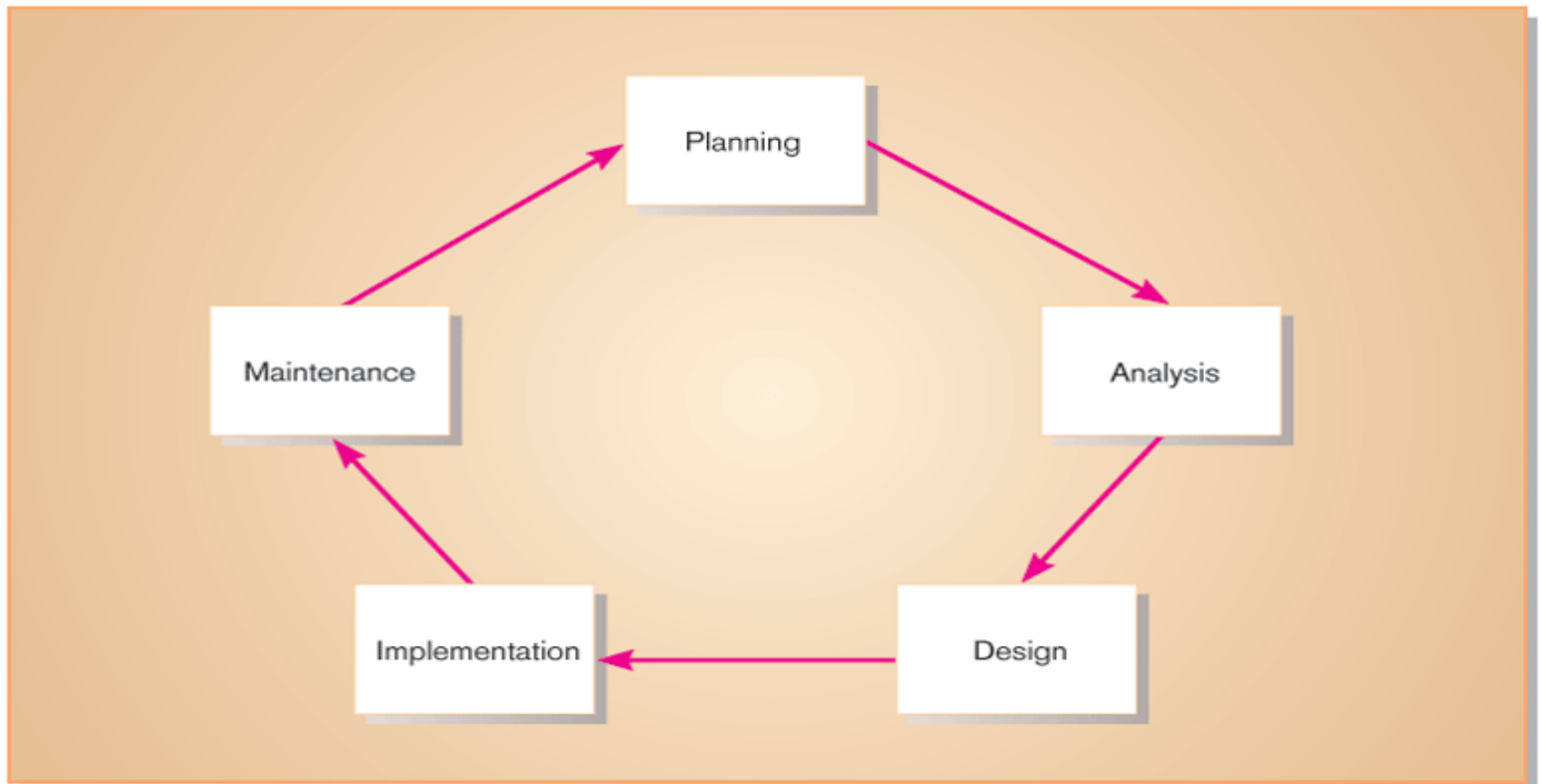
Lower CASE Tools

- Lower CASE tools generate computer source code from the CASE design.
- Source code is usually generated in several languages.
- Decreases maintenance time
- Generates error-free code

SDLC and its alternative methods

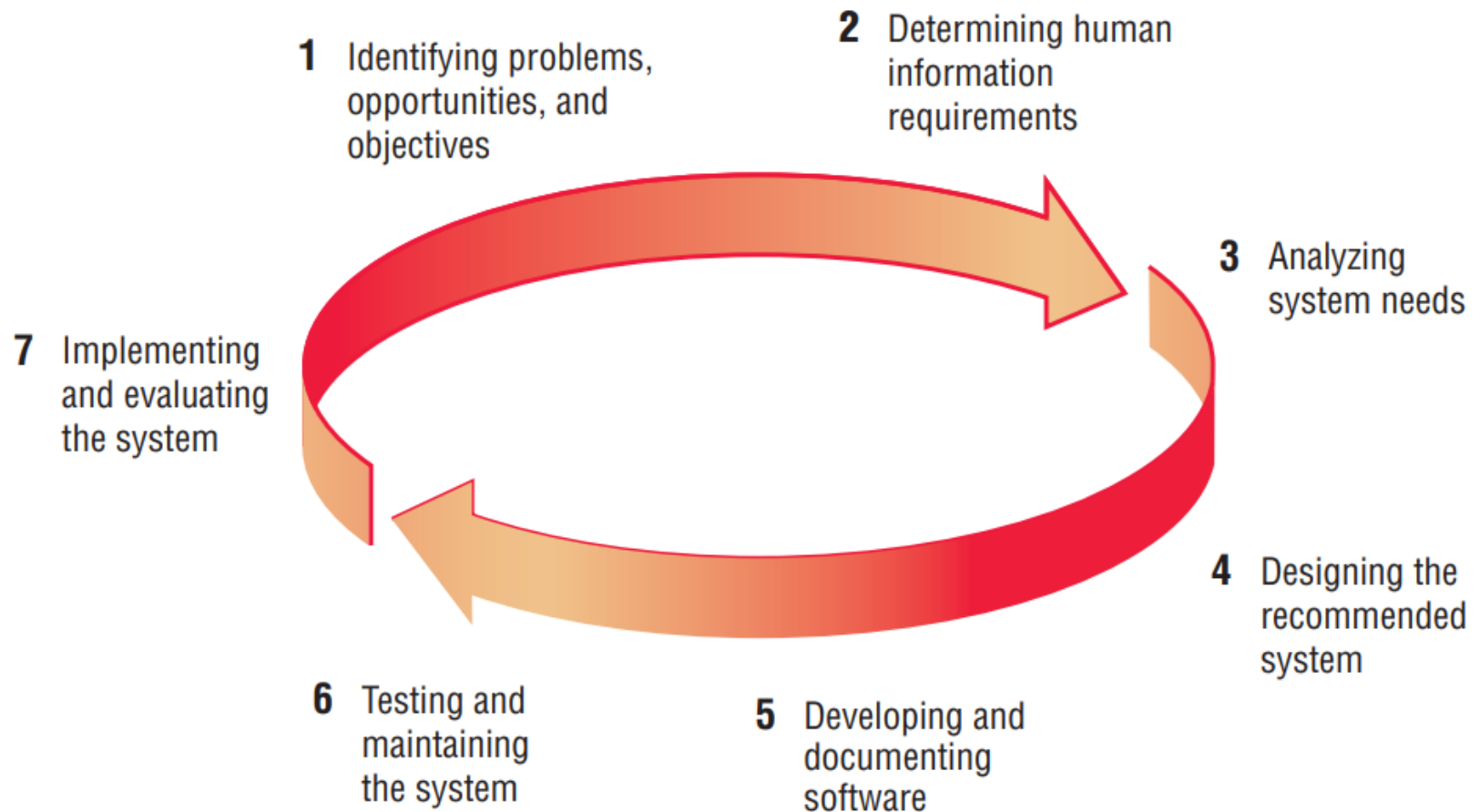
- Prof. Dr. Hoffer
- Prof. Dr. Kendall

Figure 1-3 The systems development life cycle



SDLC and its alternative methods

- **Prof. Dr. Hoffer**
- **Prof. Dr. Kendall**



The people involved in this phase are the **analysts** and users, operations managers and operations workers.

The **systems analyst** needs to know the details of **current system functions**:

who (the people who are involved),

what (the business activity),

where (the environment in which the work takes place),

when (the timing), and

how (how the current procedures are performed)

why the business uses the current system.

Alternatives to Traditional SDLC

- Agile Methodologies
- Object oriented method

The Agile Approach/Method

In software/systems development, **agile** approaches involve discovering requirements and developing solutions through

- ❑ the collaborative effort of self organizing and

- ❑ cross-functional teams and their customer(s)/end user(s).

The Agile Approach/Method

Agile method supports

- ❖ adaptive planning (having an ability to change to suit changing conditions),
- ❖ evolutionary development (the gradual development of information systems projects),
- ❖ early delivery,
and
- ❖ continual improvement

The Agile Approach/Method





Agile approach encourages flexible responses to change

based on

- values,
- principal
- core practices/
- methodology

Agile Values *(from the Agile Manifesto)*

The **Agile Manifesto** establishes a set of **four values** that are **people-centric** and **results-driven**:

Individuals and Interactions		<i>over</i>	Processes and Tools
Working Software		<i>over</i>	Comprehensive Documentation
Responding to Change		<i>over</i>	Following a Plan
Customer Collaboration		<i>over</i>	Contract Negotiation

That is, while there is value in the items on the right, we value the items on the left *more*.

(www.agilemanifesto.org)

Applying Agile Principles

In Software Development



Customer satisfactions



Changing requirements



Frequent delivery



Communicate regularly



Support team member



Face-to-face communication



Measure work progress



Development process



Good design



Measure progress

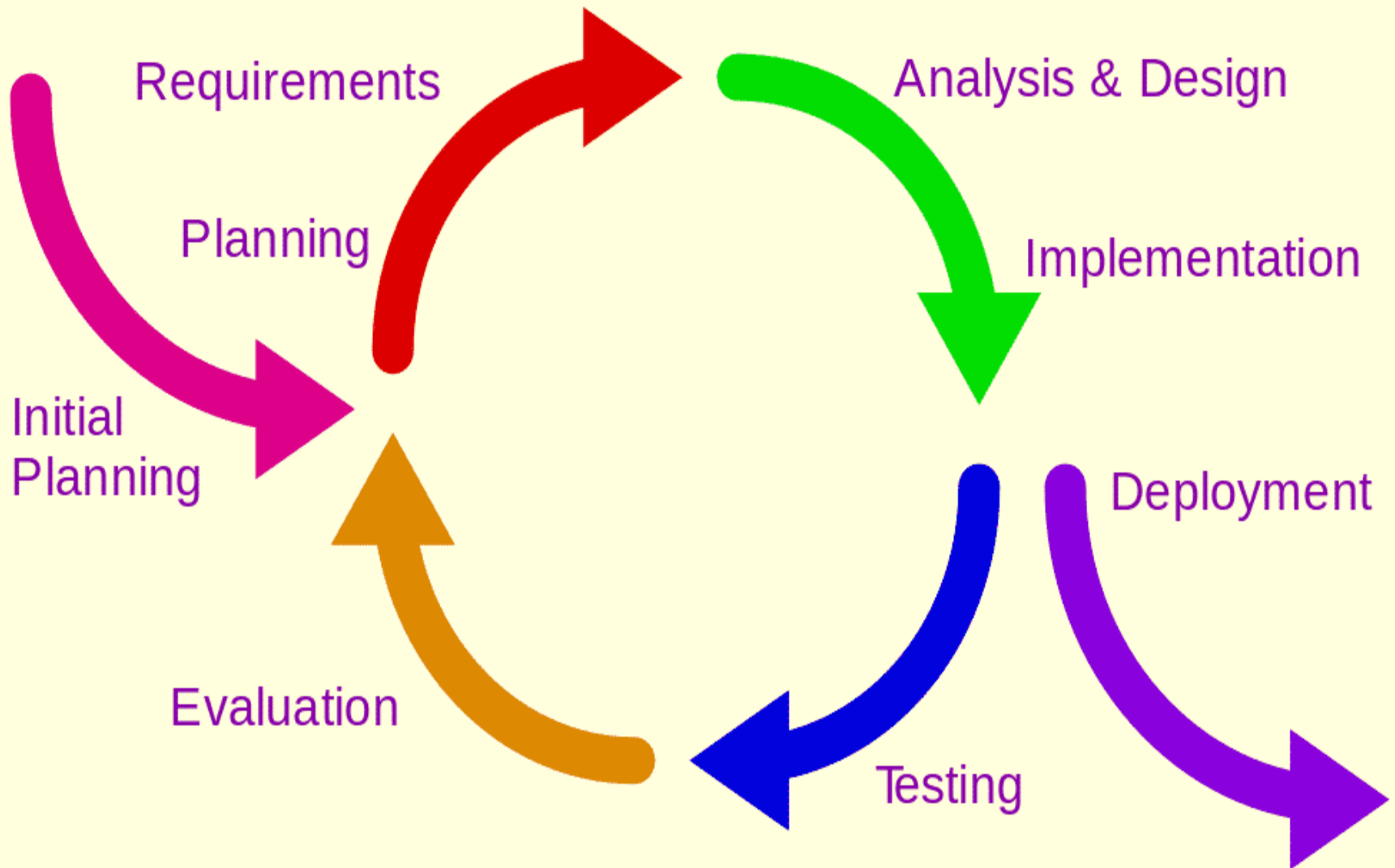


Continue seeking result



Reflect and adjust regularly

5 Agile development best practices





Four Agile Resources

Resources are adjusted to ensure successful project completion.

✓ Time

✓ Cost

✓ Quality

✓ Scope

Object-Oriented (O-O) Systems Analysis and Design

- Alternate approach to the structured approach of the SDLC that is intended to facilitate the development of systems that change rapidly in response to **dynamic business environments**
- Analysis is performed on a small part of the system followed by design and implementation.
- The cycle repeats with analysis, design, and implementation of the next part and this repeats until the project is complete.
- Examines the objects of a system

Unified Modeling Language (UML) Phases

- Define the use case model:
 - Use case diagram
 - Use case scenarios
- Create UML diagrams.
- Develop class diagrams.
- Modify the UML diagrams.
- Develop and document the system.

Choosing a Method

- Choose either:
 - SDLC
 - Agile
 - Object-oriented methodologies

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

When to Use SDLC

- Systems have been developed and documented using SLDC.
- It is important to document each step.
- Upper level management feels more comfortable or safe using SDLC.
- There are adequate resources and time to complete the full SDLC.
- Communication of how new systems work is important.

When to Use Agile

- There is a project champion of agile methods in the organization.
- Applications need to be developed quickly in response to a dynamic environment.
- A rescue takes place (the system failed and there is no time to figure out what went wrong).
- The customer is satisfied with incremental improvements.
- Executives and analysts agree with the principles of agile methodologies.

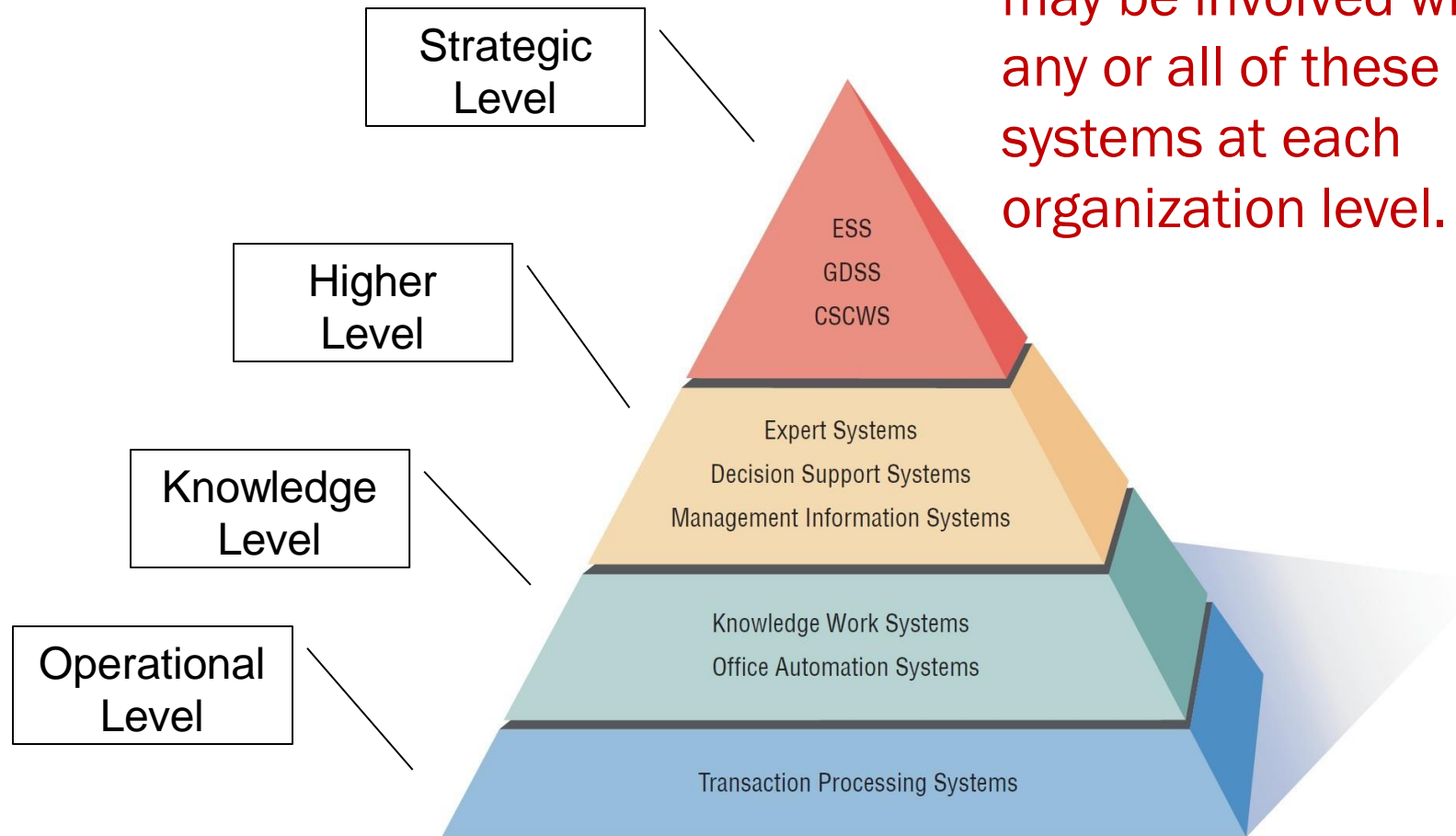
When to Use Object-Oriented

- The problems modeled lend themselves to classes.
- An organization supports the UML learning.
- Systems can be added gradually, one subsystem at a time.
- Reuse of previously written software is a possibility.
- It is acceptable to tackle the difficult problems first.

Different Types of Information Systems

- Transaction Processing Systems (TPS)
- Office Automation Systems (OAS)
- Knowledge Work Systems (KWS)
- Management Information Systems (MIS)
- Decision Support Systems (DSS)
- Expert Systems (ES)
- Executive Support Systems (ESS)
- Group Decision Support Systems (GDSS)
- Computer-Supported Collaborative Work Systems (CSCWS)

A systems analyst may be involved with any or all of these systems at each organization level.



Operational Level

Transaction Processing System (TPS)

- ❑ Process large amounts of data for routine business transactions
- ❑ Support day-to-day operations of the company
- ❑ Examples: Payroll Processing, Inventory Management

Knowledge Level

- Office Automation System (OAS)

- Supports data workers who share information, but do not usually create new knowledge
- Examples: word processing, spreadsheets, desktop publishing, electronic scheduling, communication through voice mail, email, teleconferencing

- Knowledge Work System (KWS)

- Supports professional workers such as scientists, engineers, and doctors
- Examples: computer-aided design systems, virtual reality systems, investment workstations

Higher Level

- **Management Information System (MIS)**
 - Supports organizational tasks including decision analysis and decision making
 - Examples: profit margin by sales region, expenses vs. budgets
- **Decision Support System (DSS)**
 - Supports decision makers in the making of decisions
 - Examples: financial planning with what-if analysis, budgeting with modeling
- **Expert System (ES) and Artificial Intelligence**
 - Captures and uses the knowledge of an expert for solving a particular problem which leads to a conclusion or recommendation

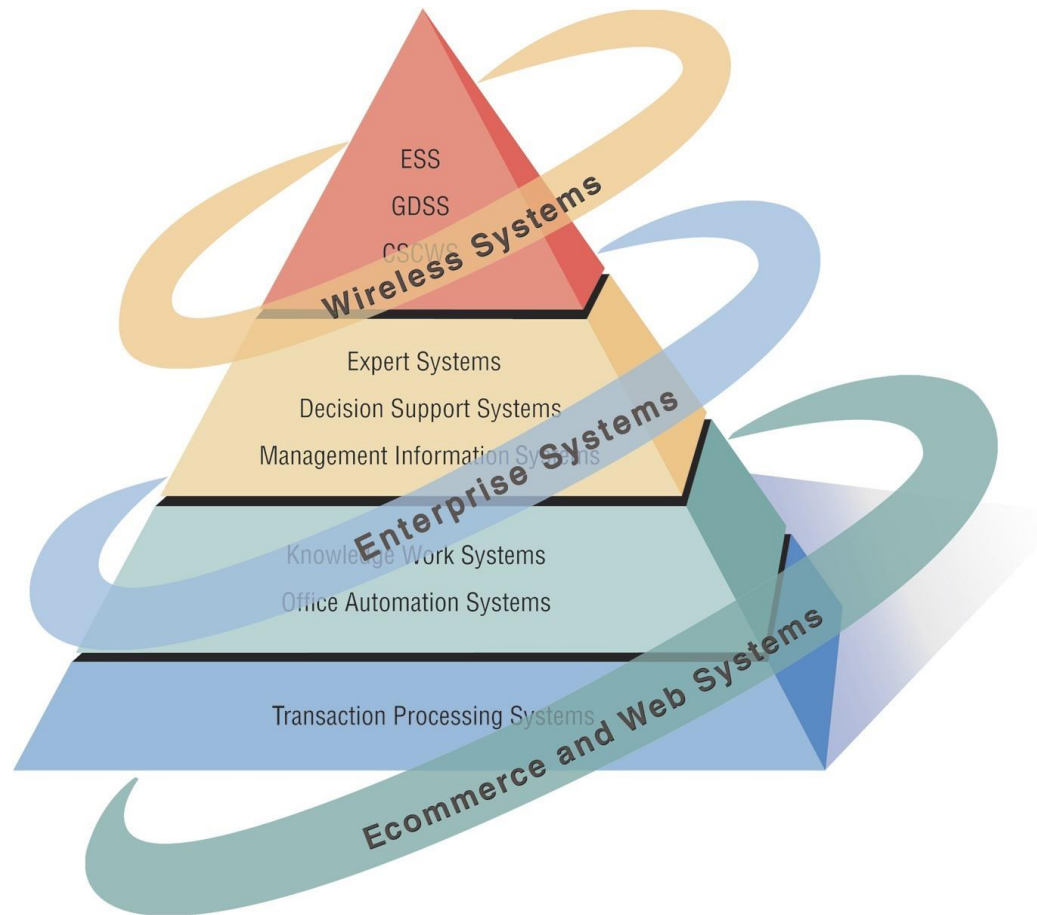
Strategic Level

- **Executive Support System (ESS)**
 - Helps executives to make unstructured strategic decisions in an informed way
 - Examples: drill-down analysis, status access
- **Group Decision Support System (GDSS)**
 - Permit group members to interact with electronic support.
 - Examples: email, Lotus Notes
- **Computer-Supported Collaborative Work System (CSCWS)**
 - CSCWS is a more general term of GDSS.
 - May include software support called *groupware* for team collaboration via network computers
 - Example: video conferencing, Web survey system

Integrating New Technologies into Traditional Systems

- Ecommerce systems
- Web based SW development Systems
- Wireless communication systems
- Mobile application SW/Systems
- Enterprise resource planning systems (Oracle, Performs integration of many information systems existing on different management levels and within different functions)

Systems Analysts Need to Be Aware that Integrating Technologies Affects all Types of Systems (Figure 1.2)



Summary

- Information is a key resource.
- Systems analysts deal with many types of information systems.
- Integration of traditional systems with new technologies
- Roles and qualities of the systems analyst
- The systems development life cycle
- CASE tools
- Agile systems development
- Object-oriented systems development