

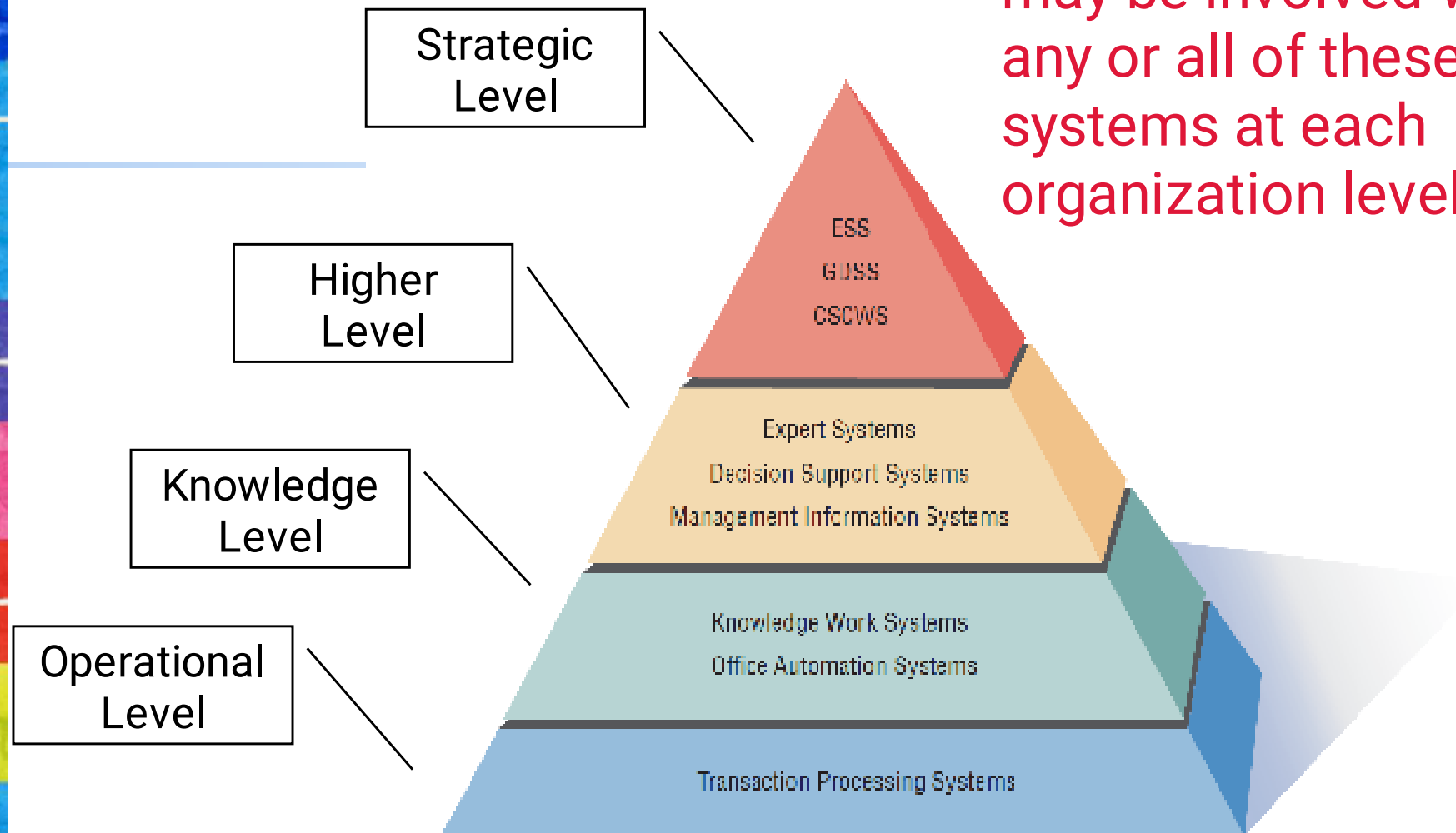
# Systems, Roles, and Development Methodologies

Systems Analysis and Design, 8e  
Kendall & Kendall

# Systems Analysts Recommend, Design, and Maintain Many Types of Systems for Users

- 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
  - Boundary-spanning
  - Support the 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 a broad spectrum of organizational tasks including decision analysis and decision making
  - Examples: profit margin by sales region, expenses vs. budgets
- Decision Support System (DSS)
  - Aids 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
  - Researching understanding natural language and the ability to reason through a problem to its logical conclusion

# 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





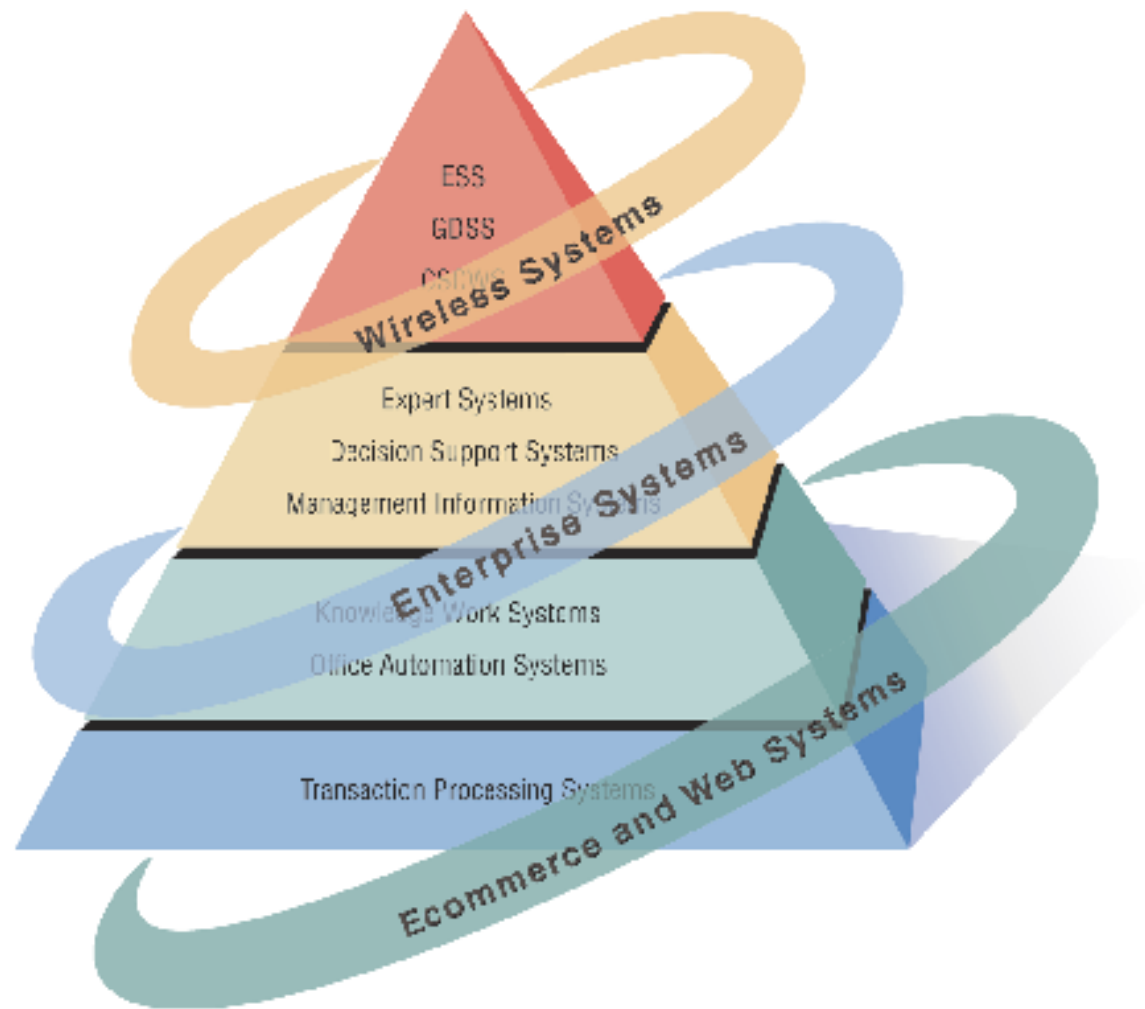
# Integrating New Technologies into Traditional Systems

---

- Ecommerce and Web Systems
- Enterprise Resource Planning Systems
- Wireless and Mobile Systems
- Open Source Software
- Need for Systems Analysis and Design



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



# Ecommerce and Web Systems

- Benefits
  - Increasing user awareness of the availability of a service, product, industry, person, or group
  - The possibility of 24-hour access for users
  - Improving the usefulness and usability of interface design
  - Creating a system that can extend globally rather than remain local, thus reaching people in remote locations without worry of the time zone in which they are located



# Enterprise Resource Planning Systems (ERP)

---

- Performs integration of many information systems existing on different management levels and within different functions
- Example: SAP, Oracle

# Wireless and Mobile Systems

- A system analyst may be asked to design standard or wireless and mobile communication networks that integrate voice, video, and email into organizational intranets or industry extranets.
- A system analyst may also be asked to develop intelligent agents.
- Example: iPhone, iPod, BlackBerry
- Wireless communication is referred to as m-commerce (mobile commerce).

# Open Source Software

- An alternative of traditional software development where proprietary code is hidden from the users
- Open source software is free to distribute, share, and modify.
- Characterized as a philosophy rather than simply the process of creating new software
- Example: Linux Operating System, Apache Web Server, Mozilla Firefox Web Browser



# Need for Systems Analysis and Design

- Installing a system without proper planning leads to great user dissatisfaction and frequently causes the system to fall into disuse.
- Lends structure to the analysis and design of information systems
- A series of processes systematically undertaken to improve a business through the use of computerized information systems

# 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

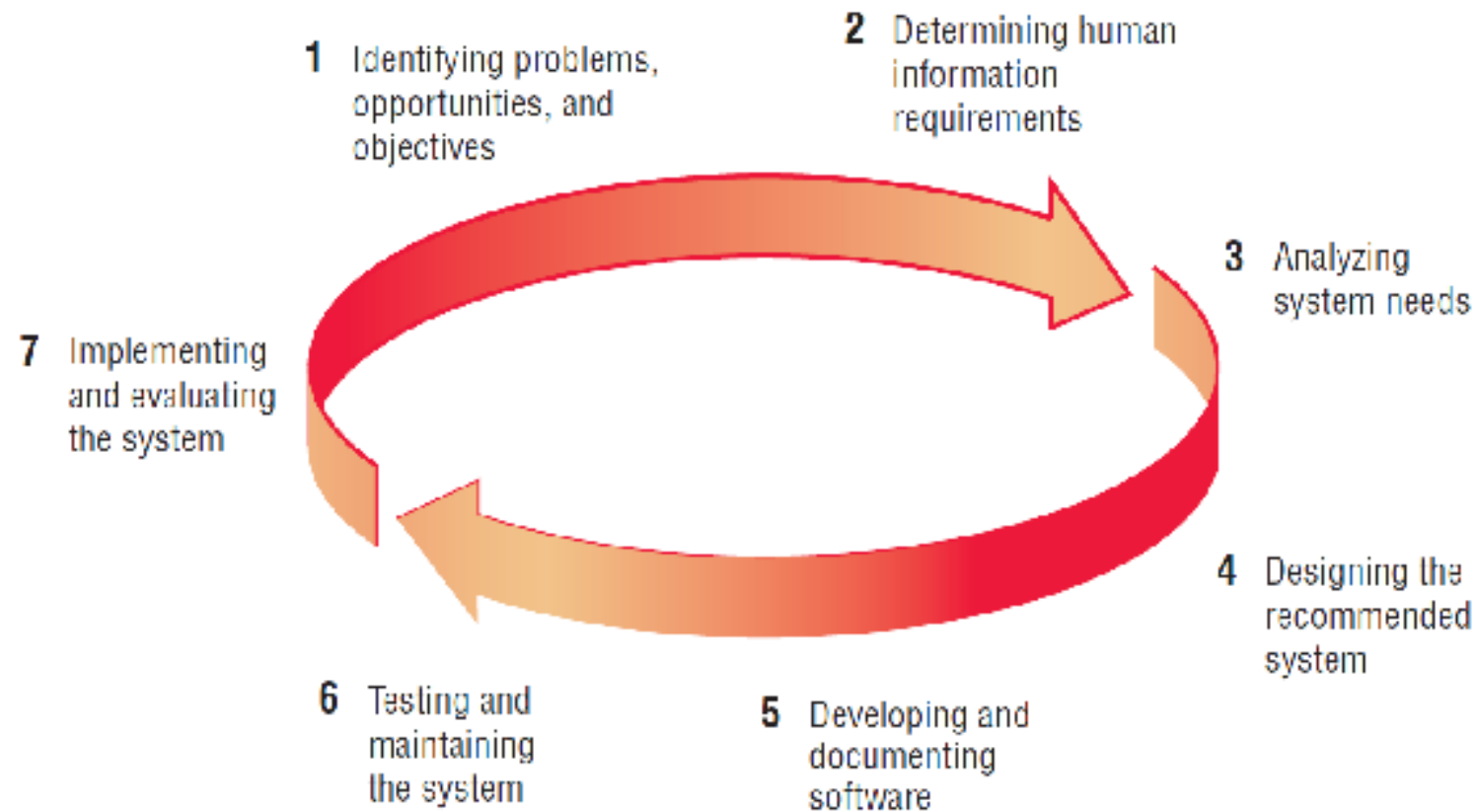


# Systems Development Life Cycle (SDLC)

---

- The systems development life cycle is a phased approach to solving business problems.
- Developed through the use of a specific cycle of analyst and user activities
- Each phase has unique user activities.

# The Seven Phases of the Systems Development Life Cycle (Figure 1.3)





# Identifying Problems, Opportunities, and Objectives

- Activity:
  - Interviewing user management
  - Summarizing the knowledge obtained
  - Estimating the scope of the project
  - Documenting the results
- Output:
  - Feasibility report containing problem definition and objective summaries from which management can make a decision on whether to proceed with the proposed project



# Determining Human Information Requirements

- Activity:
  - Interviewing
  - Sampling and investing hard data
  - Questionnaires
  - Observe the decision maker's behavior and environment.
  - Prototyping
  - Learn the who, what, where, when, how, and why of the current system.
- Output:
  - The analyst understands how users accomplish their work when interacting with a computer; and begin to know how to make the new system more useful and usable. The analyst should also know the business functions and have complete information on the people, goals, data, and procedure involved.

# Analyzing System Needs

- Activity:
  - Create data flow, activity, or sequence diagrams.
  - Complete the data dictionary.
  - Analyze the structured decisions made.
  - Prepare and present the system proposal.
- Output:
  - Recommendation on what, if anything, should be done

# Designing the Recommended System

- Activity:
  - Design procedures for data entry.
  - Design the human-computer interface.
  - Design system controls.
  - Design database and/or files.
  - Design backup procedures.
- Output
  - Model of the actual system



# Developing and Documenting Software

- Activity:
  - System analyst works with programmers to develop any original software.
  - Works with users to develop effective documentation.
  - Programmers design, code, and remove syntactical errors from computer programs.
  - Document software with help files, procedure manuals, and Web sites with Frequently Asked Questions.
- Output:
  - Computer programs
  - System documentation



# Testing and Maintaining the System

---

- Activity:
  - Test the information system.
  - System maintenance.
  - Maintenance documentation.
- Output:
  - Problems, if any
  - Updated programs
  - Documentation



# Implementing and Evaluating the System

---

- Activity:
  - Train users.
  - Analyst plans smooth conversion from old system to new system.
  - Review and evaluate system.
- Output:
  - Trained personnel
  - Installed system

# The Impact of Maintenance

- Maintenance is performed for two reasons:
  - Removing software errors
  - Enhancing existing software
- Over time the cost of continued maintenance will be greater than that of creating an entirely new system. At that point it becomes more feasible to perform a new systems study.



# Approaches to Structured Analysis and Design and to the Systems Development Life Cycle

---

- Traditional systems development life cycle
- CASE systems development life cycle
- Object-oriented systems analysis and design

# Case Tools

- CASE tools are productivity tools for systems analysts that have been created explicitly to improve their routine work through the use of automated support.
- Reasons for using CASE tools
  - Increasing analyst productivity
  - Improving analyst-user communication
  - Integrating life cycle activities

# 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



# The Agile Approach

---

- Based on:
  - Values
  - Principles
  - Core practices



# Agile Values

---

- Communication
- Simplicity
- Feedback
- Courage



# Four Agile Resources

- Resources are adjusted to ensure successful project completion.
  - Time
  - Cost
  - Quality
  - Scope

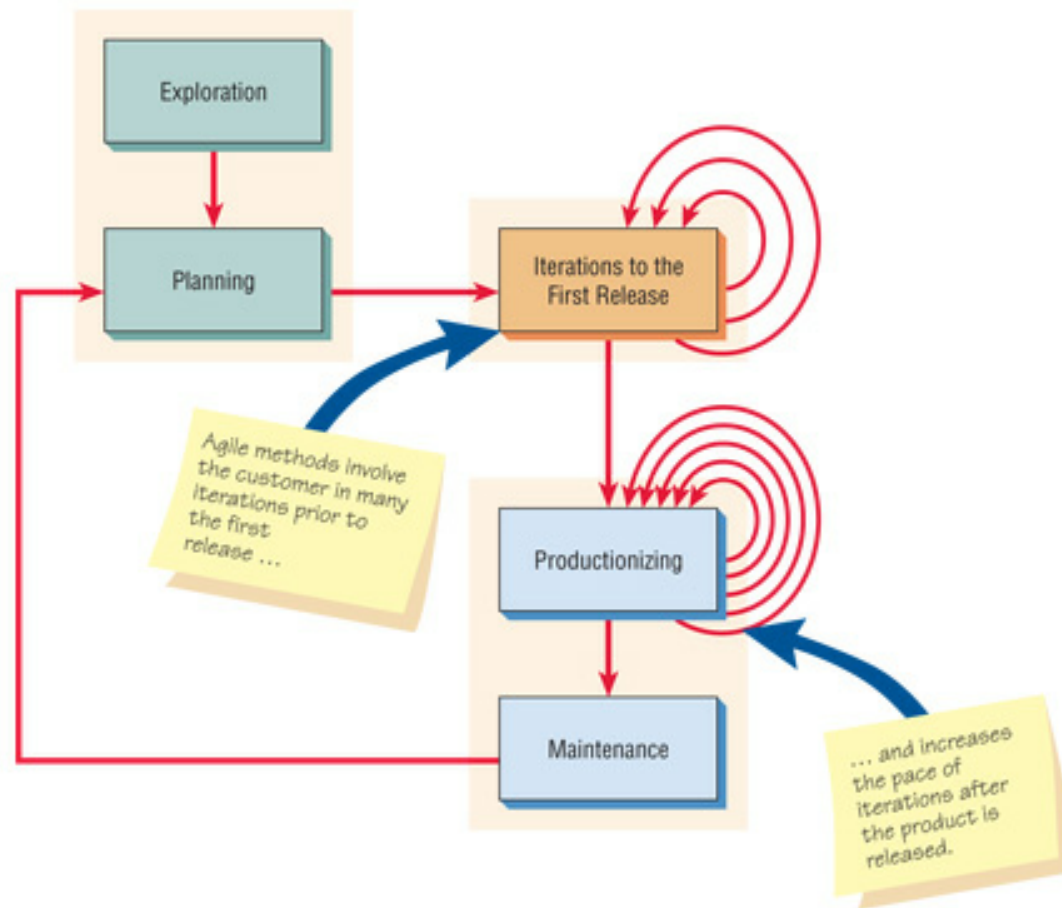


# Five Stages of Agile Development

---

- Exploration
- Planning
- Iterations to the first release
- Productionizing
- Maintenance

# Agile Project Development Process (Figure 1.7)





# 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.
- Draw statechart diagrams.
- Modify the UML diagrams.
- Develop and document the system.

# Choosing a Method

---

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

# 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.

# 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