

Systems Analysis and Design

Systems, Roles, and
Development
Methodologies

Why Project Fails

- Did not support business strategy & objectives
- Poor planning and project management
- Failure to understand user requirement
- inadequate cost vs benefit analysis
- Myriad of design defect/errors
- Installation of incompatible or inadequate technology
- No adequate control implemented
- Unstructured and un-maintainable software

The Bad News

- 30% of large IT projects are **cancelled** before completion
- 50% of IT projects are **over-budget** by more than 200%
- The majority of completed projects deliver *60% or less of prescribed functionality*
- Many delivered information systems are **under-used** because they don't meet user needs and/or expectations

Software Horror Stories

- **Bank of America** spent \$23,000,000 on a 5-year project to develop a new accounting system. Spent over \$60,000,000 trying to make new system work, finally abandoned it. Loss of business estimated in excess of \$1,000,000,000
- **The B1 Bomber** required an additional \$1,000,000,000 to improve its air defense software, but the software still isn't working to specification
- **Ariane 5, Flight 501**
 - The loss of a \$500,000,000 spacecraft was ultimately attributed to **errors in requirements, specifications** and inadequate **software reuse practices**.



How the customer explained it



How the Project Leader understood it



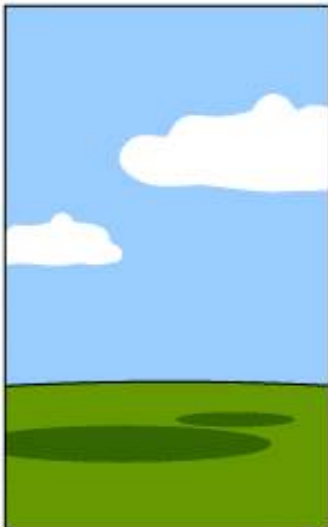
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



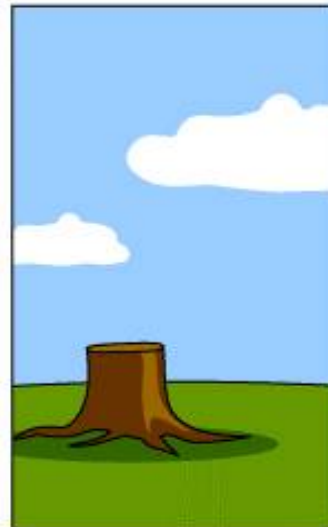
How the project was documented



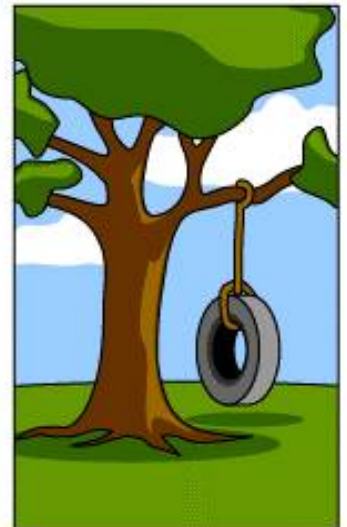
What operations installed



How the customer was billed



How it was supported



What the customer really needed

Information—A Key Resource

- Fuels business and can be the critical factor in **determining** the **success** or **failure** of a business
- Needs to be **managed correctly**
- Managing computer-generated information differs from handling manually produced data

System

What is system?

- A system is a **group of interrelated components** that **function together** to achieve a desired result.

What is information system?

An **IS** is an arrangement of

- *people, data, processes*, and **information technology**

that interact to **collect, process, store**, and **provide** as output the information needed to support an organization.

Information System (cont.)

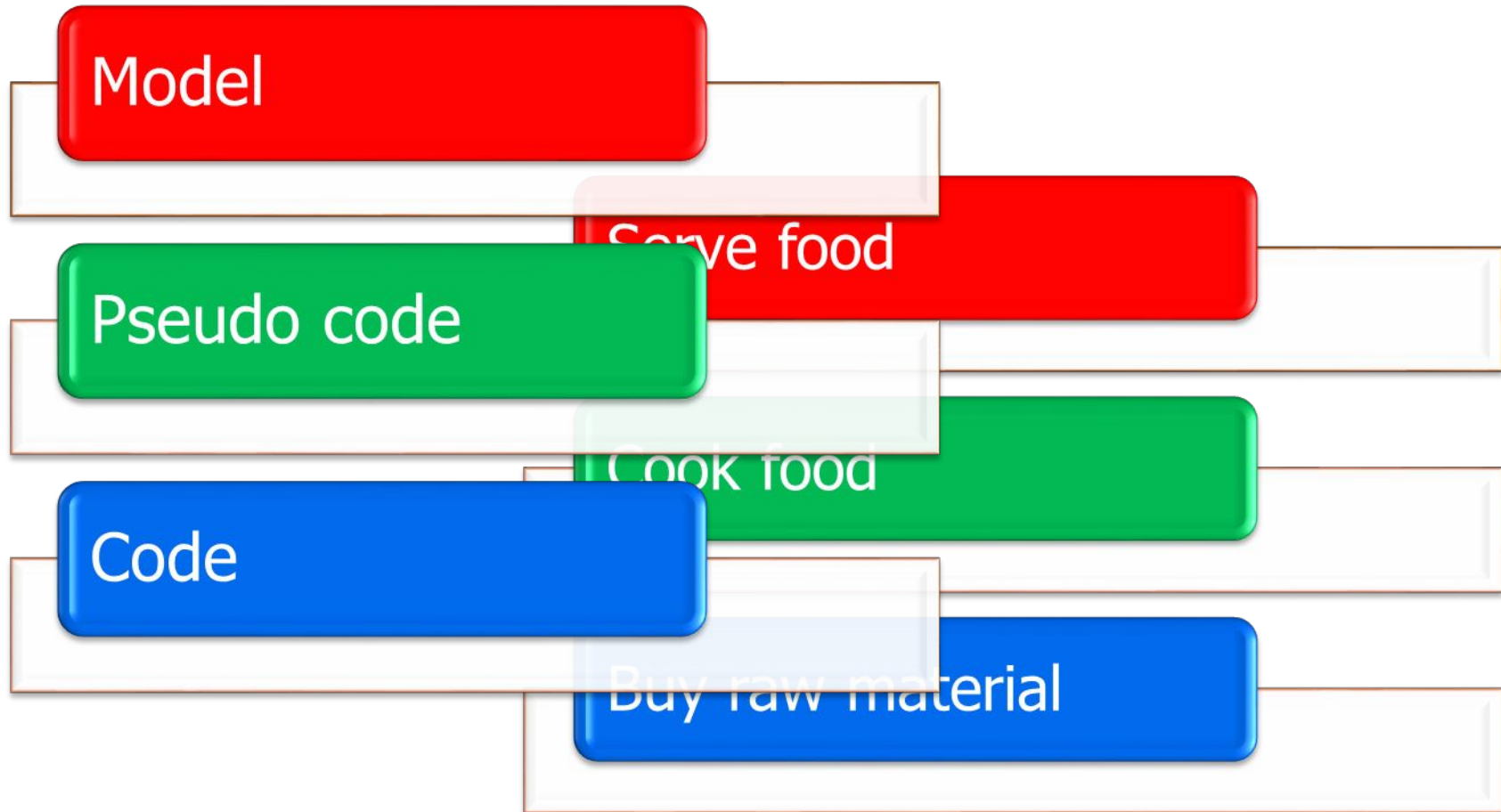
- *Information systems* are **software** applications which **manage large amounts of data**.
- Most of the software out there is information systems software,
 - written in languages such as Java, C++, .NET and the like.



Type of Systems

- Information systems fall into one of the following eight categories:
 - Transaction Processing System (TPS)
 - Office Automation Systems (OAS)
 - Knowledge Work Systems (KWS)
 - Management Information Systems
 - Decision Support Systems (DSS)
 - Expert Systems (ES)
 - Group Decision Support Systems (GDSS)
 - Executive Support Systems (ESS)

Why study System Analysis Design



Learning Objectives

- Understand the need for systems analysis and design in organizations.
- Realize roles of the systems analyst.
- Comprehend fundamentals of three development methodologies:
 - SDLC
 - The agile approach
 - Object-oriented systems analysis and design

Major Topics

- Fundamentals of different kinds of information systems
- Roles of systems analysts
- Phases in the systems development life cycle as they relate to Human-Computer Interaction (**HCI**) factors
- CASE tools
- Open Source Software

Need for Systems Analysis and Design

Remember, finding and fixing a fault after software delivery is 100x more expensive than finding and fixing it during systems analysis or early design phases

- Installing a system without proper planning
 - Leads to great user dissatisfaction and
 - Frequently causes the system to fall into disuse
- User involvement throughout the systems demand for system analysis
- New technologies are also driving the need for systems analysis

What is Systems Analysis?

- The collection of
 - **Notations**
 - **Methodologies** and
 - **Tools**
- Used to gather details and
 - Analyze a problem situation
 - prior to information system design and implementation

What is Systems Analysis?

- *Systems analysis (or, requirements analysis)* must ensure that the proposed information system meets
 - User needs,
 - Can be delivered on time, and
 - Can be updated inexpensively.
- Problems in "getting the systems analysis right",
- such as **ill-defined situations, ambiguities, inconsistencies**, mixing requirements with design

Roles of the Systems Analyst

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

Roles of the Systems Analyst

System analyst as a consultant

- **Advantage**

- can bring with them a fresh perspective that other people in an organization do not possess.

- **Disadvantage**

- true organizational structure can never be known to an outsider.

Roles of the Systems Analyst

Supporting expert:

- play as professional expertise concerning *computer hardware* and *software* and their uses in the business.
- Serves as a resource for those who are working on and managing other projects

Roles of the Systems Analyst

- **Agent of change:**
 - A person who serves as a catalyst for
 - Change,
 - Develops a plan for change, and
 - Works with others in facilitating that change.

Systems Analysts

a specialist who studies the

- ***problems* and *needs*** of an organization to determine
 - how people, data, processes, and information technology
 - can best accomplish improvements for the business.

Qualities of the Systems Analyst

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

Qualities of the Systems Analyst

- **Problem solver**

- views the analysis of problems as a challenge and enjoys devising workable solutions

- **Communicator**

- Capable of relating meaningfully to other people over extended periods over time.
- Need enough computer experience to program, to understand the capabilities of computers,
- glean information requirements from users, and
- communicate what is needed to programmers.

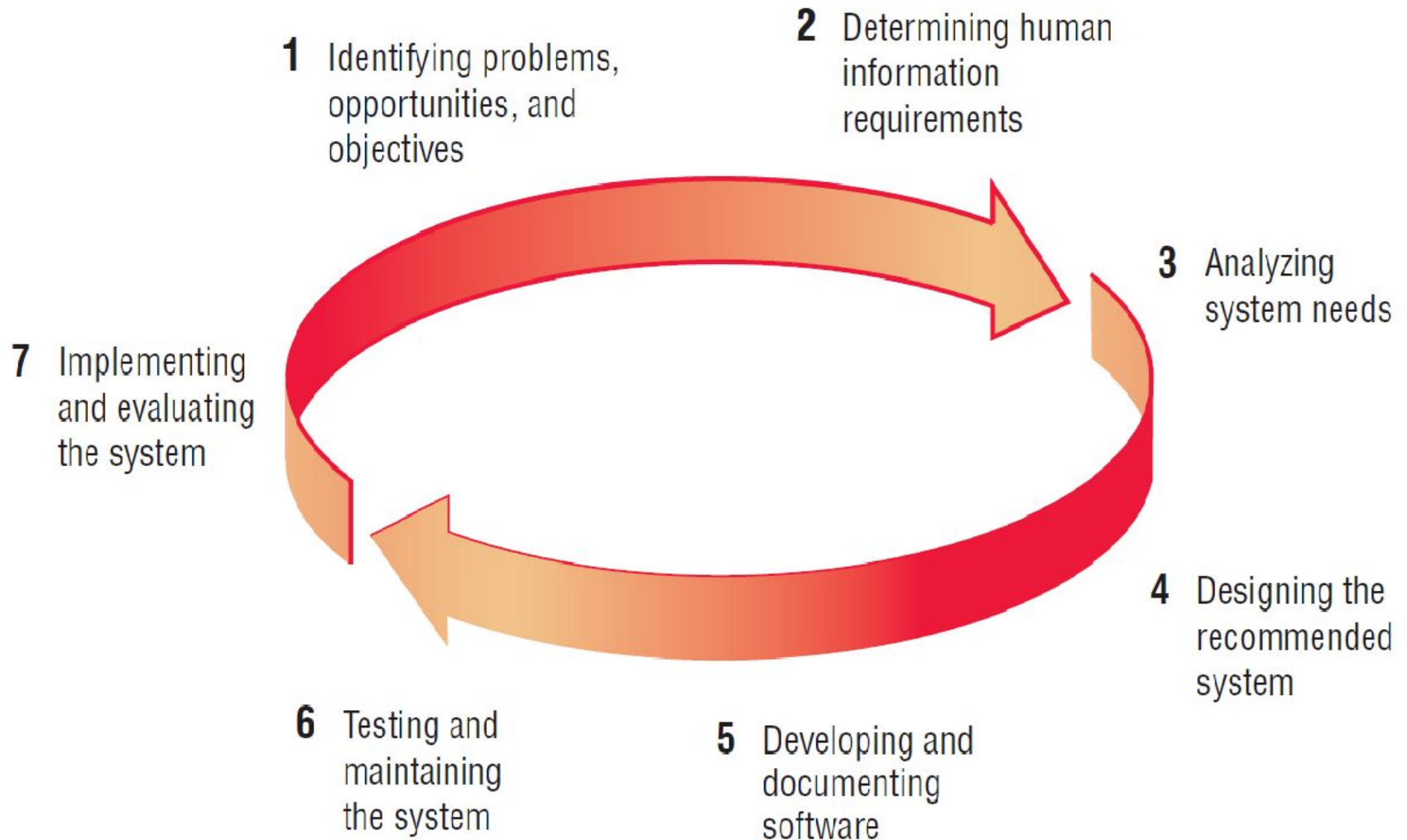
Design Methodology

- Several design methodology are used to communicate between technical people and the business people
- There are three approaches to designing new system.
 - **SDLC**
 - **Agile**
 - **UML**

Systems Development Life Cycle (SDLC)

- is a systematic and orderly approach
 - to solving business problems, and
 - developing and supporting resulting information systems.
- Developed through the use of a ***specific cycle of analyst and user activities***
- Each phase has unique user activities and certain ***deliverable***

The Seven Phases of the Systems Development Life Cycle (Figure 1.1)



Incorporating Human-Computer Interaction (HCI) Considerations

- The demand for analysts who are capable of incorporating HCI into the systems development process keeps increasing,
- as companies begin to realize that the *quality of systems* and the *quality of work* life can be improved by taking a human-centered approach at the outset of a project

1) Identifying Problems, Opportunities, and Objectives

- **Problems:** -Recognize the Problem, Define the problem
- **Opportunities** : situations that the analyst believes can be improved through the use of computerized information systems.
- **Objectives** : how can the business reach its objectives by addressing specific problems or opportunities.

Identifying Problems, Opportunities, and Objectives

● **Activity:**

- Interviewing user management
- Summarizing the knowledge obtained
- Estimating the scope of the project
- Documenting the results

Identifying Problems, Opportunities, and Objectives

Output:

- **Feasibility report containing**

- Problem definition and

- Objective summaries from which

Management can make a decision on whether to proceed with the proposed project

2) Determining Human Information Requirements (HCI)

The primary purpose of Phase II is to understand what users need to perform their jobs.

Formal process of using

- **Research,**
- **Interview,**
- **Questionnaires,**
- Sampling And Other Techniques To Collect Information About
 - **System,**
 - **Requirements And**
 - **Preferences**

Determining Human Information Requirements (HCI)

Activity:

- Interviewing
- Sampling and investigating 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

Determining Human Information Requirements (HCI)

There are several questions an analyst must answer in this phase:

- **Who**—the people who are involved
- **What**—the business activity
- **Where**—the environment in which the work takes place
- **When**—the timing
- **How**—how the current procedures are performed
- **Why**—why the system uses the current system

Determining Human Information Requirements (HCI)

OUTPUT:

- The analyst understands how users accomplish their work when interacting with a computer.
 - Begin to know how to make the new system more useful and usable
 - Know the business functions
- Have complete information on the:
 - People
 - Goals
 - Data
 - Procedure involved

3) Analyzing System Needs

The primary purpose of Phase III is to determine what **information** and **processes** are needed to resolve a problem or take advantage of an opportunity.

- Typical tools used in this phase are:
- Data flow diagrams
- Data dictionaries
- Analyzing structured decisions using structured English, decision tables, and decision trees
- Analyzing semi-structured decisions using the tradeoff process and the use of weighting methods

3) Analyzing System Needs

Deliverables:

- Data Flow Diagram and optionally a Data Dictionary.
- Presentation of a Systems Proposal.
- Updated Schedule
- Updated Risk Management Document

4) Designing the Recommended System

- The primary purpose of Phase IV is to design the logical and physical components necessary to implement the recommended system.
- This includes
 - database design,
 - input/output layouts,
 - logical data flow,
 - structured & semi-structured decision designs, and the architectural design.
 - In addition, controls and backup procedures are documented.

4) Designing the Recommended System

Typical tools used in this phase are:

- Prototyping Tools
- ERD Tool
- Data Dictionary Tools

4) 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

Deliverables:

- Presentation of Detail Design Document to developers which includes:
 - Data Flow Diagrams, ERD's, & Data Dictionary
 - Controls & Backup Procedure
 - Input/Output Layouts
 - Logical & Physical Designs
 - Architectural Design
- Updated Project Schedule
- Updated Vision/Scope Document
- Updated Risk Management Document

5) Developing and Documenting Software

- Purpose of Phase V is to build the recommended system.
- This includes purchasing hardware & software, building custom software components, and preparing for full system testing.

5) Developing and Documenting Software

Typical tools used in this phase are:

- Software Development Tools
- Development Environment
- Documentation Tools

5) 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

(5) Developing and Documenting Software

- Output:
 - Computer programs
 - System documentation

(6) Testing and Maintaining the System

Activity:

- Test the information system
- System maintenance
- Maintenance documentation

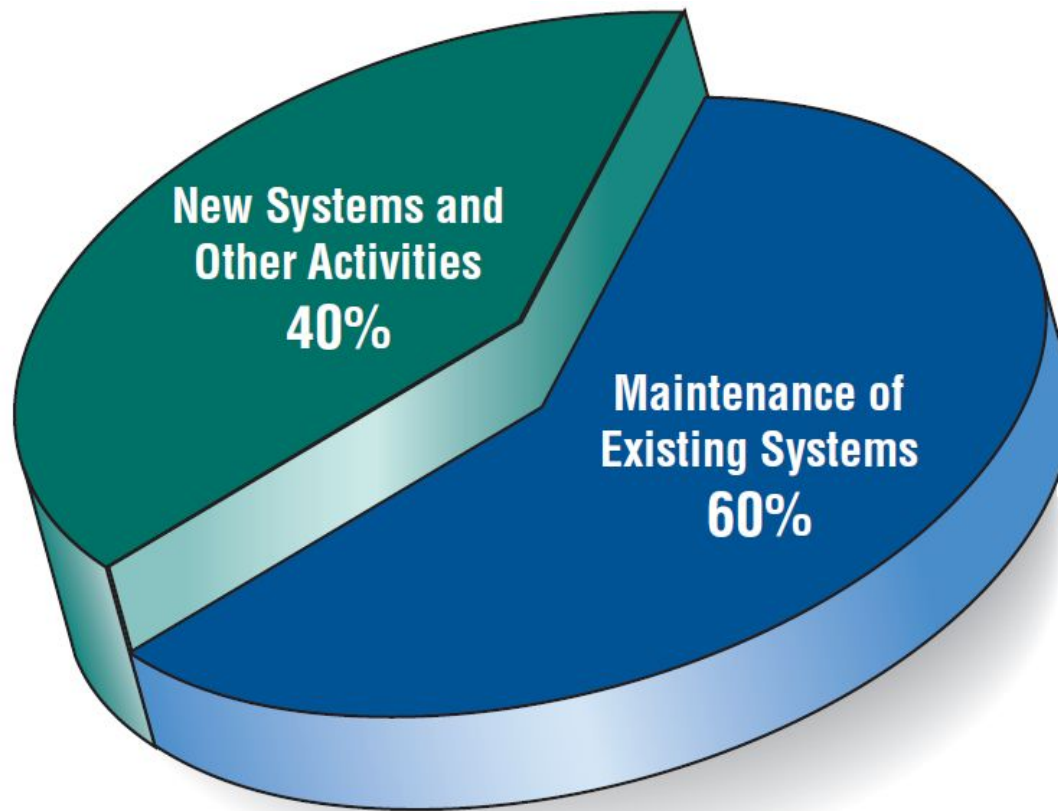
Output:

- Problems, if any
- Updated programs
- Documentation

(7) 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

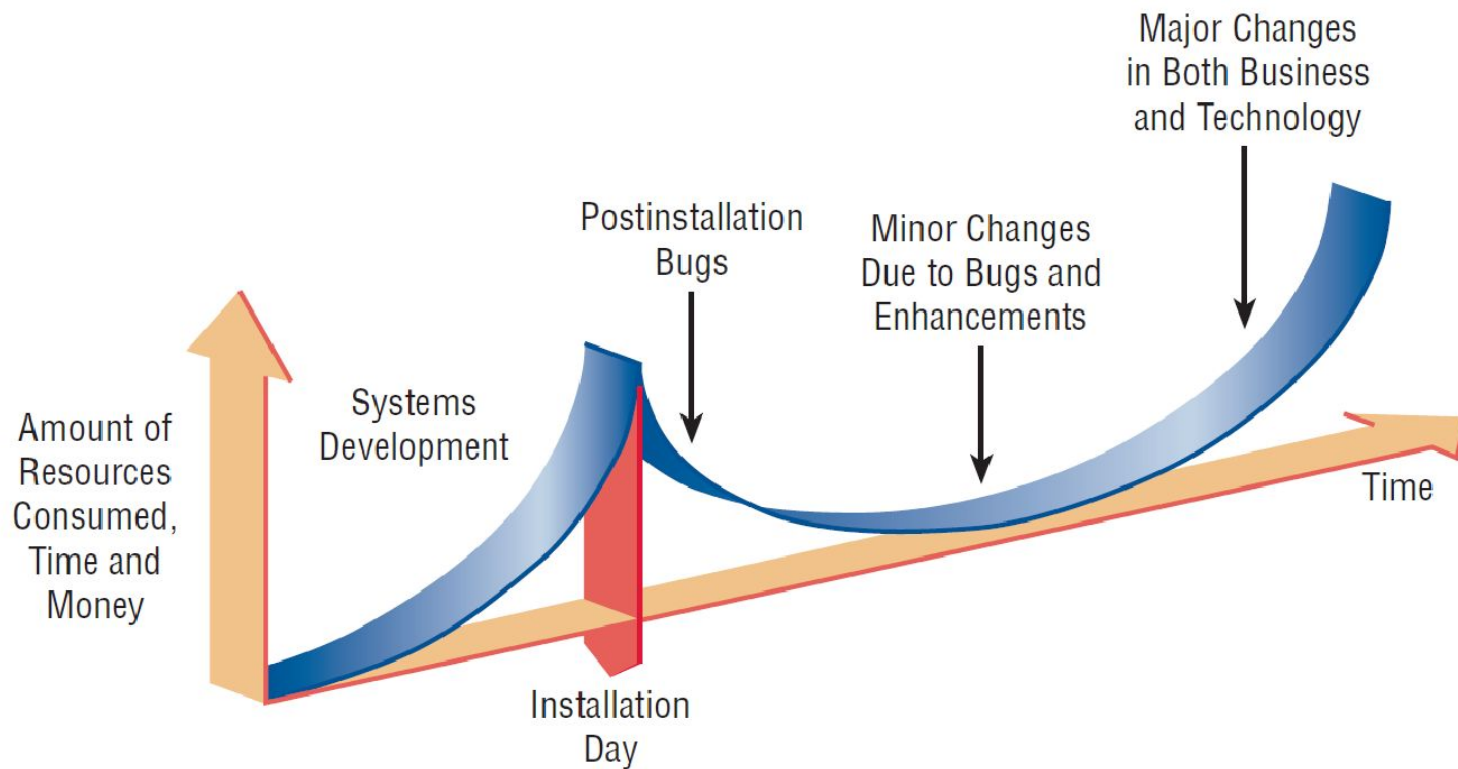
Some Researchers Estimate that the Amount of Time Spent on Systems Maintenance May Be as Much as 60 Percent of the Total Time Spent on Systems Projects (Figure 1.2)



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.

Resource Consumption over the System Life (Figure 1.3)



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

- Traditional systems development life cycle
- CASE(Computer- Aided Software Engineering) systems development life cycle
- Object-oriented systems analysis and design

Case Tools

- CASE tools are set of software application programs, which are used to automate SDLC activities.
- These tools simplify various stages of SDLC
- Use of CASE tools accelerates the development of project to produce desired result and
- helps to uncover flaws before moving ahead with next stage in software development.
- Analysis tools, Design tools, Project management tools, Database Management tools, Documentation tools

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

- Reasons for using CASE tools
 - Increasing analyst **productivity**
 - Improving analyst-user **communication**
 - Integrating life cycle activities

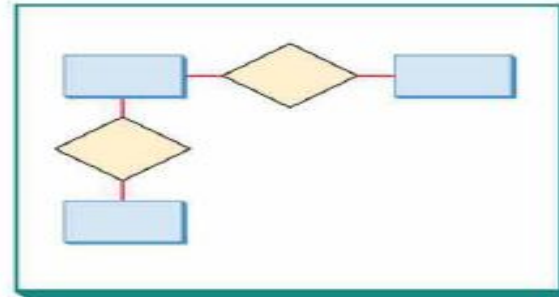
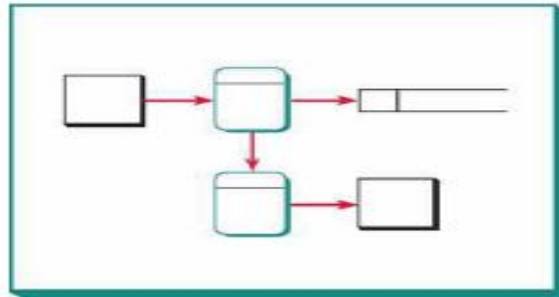
CASE tool

- Visible Analyst (VA) is one example of a CASE tool that enables systems analysts to do
- **graphical planning, analysis, and design** in order to build complex client/server applications and database
- Visible Analyst and Microsoft Visio allow users to
- draw and modify diagrams easily.

ADD CUSTOMER	
NUMBER	XXXXXX
NAME	XXXXXXXXXXXXXX
STREET	XXXXXXXXXXXXXX
CITY	XXXXXXXXXXXXXX
STATE	XX
ZIP	XXXXX-XXXX

SALES ANALYSIS REPORT	
ITEM DESCRIPTION	TOTAL SALES
XXXXXXXXXXXXXXXXXX	ZZ,ZZ9
XXXXXXXXXXXXXXXXXX	ZZ,ZZ9
XXXXXXXXXXXXXXXXXX	ZZ,ZZ9
XXXXXXXXXXXXXXXXXX	ZZ,ZZ9
XXXXXXXXXXXXXXXXXX	ZZ,ZZ9
XXXXXXXXXXXXXXXXXX	ZZ,ZZ9

Screen and Report Design



System Diagrams and Models

Item = Number +
Description +
Cost +
Price +
Quantity on hand +
Quantity on order +
Reorder point +
Monthly sales +
Year to date sales

```

DO WHILE NOT End of file
  Read Item record
  IF Item is low in stock
    Print Purchase Order
    Update Item record
  ENDIF
ENDDO
  
```

Data Dictionary and Process Logic

System Requirements
<ul style="list-style-type: none"> • Add new customers • Identify fast- and slow-selling items • Enter customer orders • Look up customer credit balance • Maintain adequate inventory

Deliverables
<ul style="list-style-type: none"> • Add customer screen • Item Analysis Report • Customer order entry screen • Customer inquiry screen • Vendor purchase order program • Seasonal forecasting

Project Management

The Agile Approach

- Agile is a software development approach.
- *Agile* is a term used to describe *approaches* to software development emphasizing incremental delivery, collaboration and continual learning
- Based on:
 - Values
 - Principles
 - Core practices

Agile Values

The four values are

- Communication
- Simplicity
- Feedback
- Courage

We recommend that systems analysts adopt these values in all projects they undertake, not just when adopting the agile approach.

Four Agile Resources

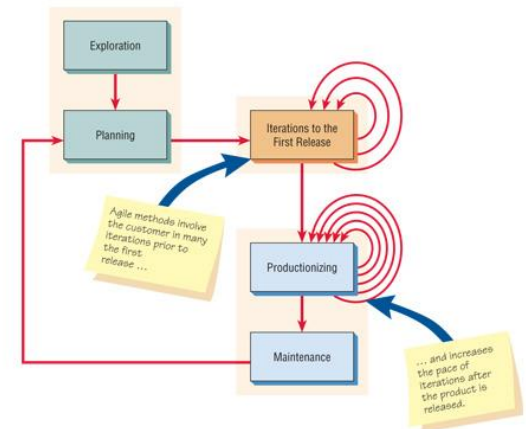
Agile methods can ensure successful completion of a project by adjusting the important resources of

- Time
 - Cost
 - Quality
 - Scope
- **When these four control variables are properly included in the planning, there is a state of balance between the resources and the activities needed to complete the project**

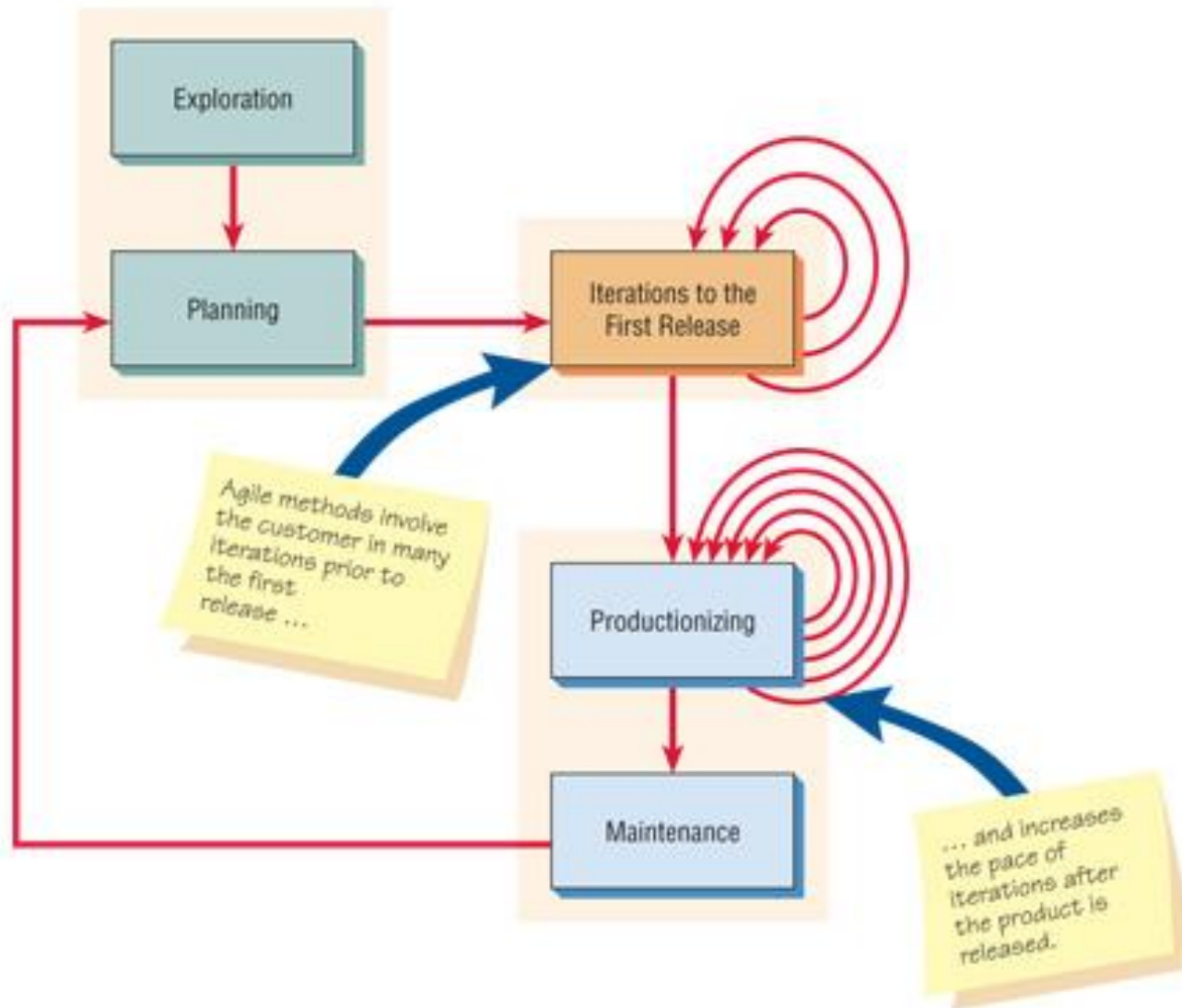
Developmental Process for an Agile Project

Five Stages of Agile Development

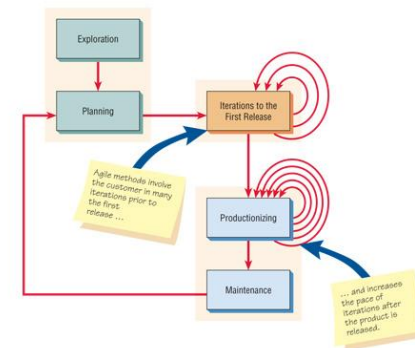
- Exploration
- Planning
- Iterations to the first release
- Productionizing
- Maintenance



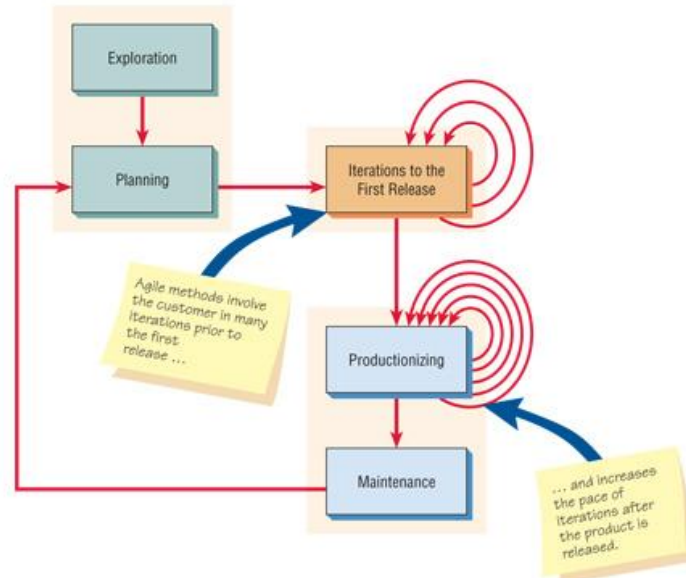
Agile Project Development Process



- three **red arrows** that loop back into the “Iterations” box symbolize
 - Incremental changes created through Repeated Testing and Feedback that eventually lead to a
 - Stable but Evolving system.
- Multiple Looping Arrows that feed back into the productionizing phase.
 - These symbolize that the pace of iterations is increased after a product is released.



- The **red arrow** is shown leaving the maintenance stage and returning to the planning stage,
- so that there is a continuous feedback loop involving customers and the development team as they agree to alter the evolving system



EXPLORATION:

- explore Environment, assemble the team, and assess team member skills
- take from a few weeks to few month
- examining potential technologies needed to build the new system
- Estimating the time
- playful and curious attitude toward the work environment, its problems, technologies, and people

PLANNING

- planning may take a few days to accomplish
- Story cards in the planning game briefly describe the task, provide notes, and provide an area for task tracking.
- Customers decide what the development team should tackle first.
- Their decisions will set priorities and check functionalities throughout the process.

ITERATIONS TO THE FIRST RELEASE

- iterations are cycles of testing, feedback, and change
- Run customer-written functional tests at the end of each iteration
- should question whether the schedule needs to be altered
- celebrate your progress

PRODUCTIONIZING

- feedback cycle speeds up
- daily briefings so everyone knows what everyone else is doing.
- product released may be improved by adding other features

MAINTENANCE

- New features may be added,
- riskier customer suggestions may be considered,
- team members may be rotated on or off the team

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

- *Object-oriented (O-O) analysis and design* is an approach that is intended to facilitate the development of systems that **must change rapidly** in *response to dynamic business environments*.

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

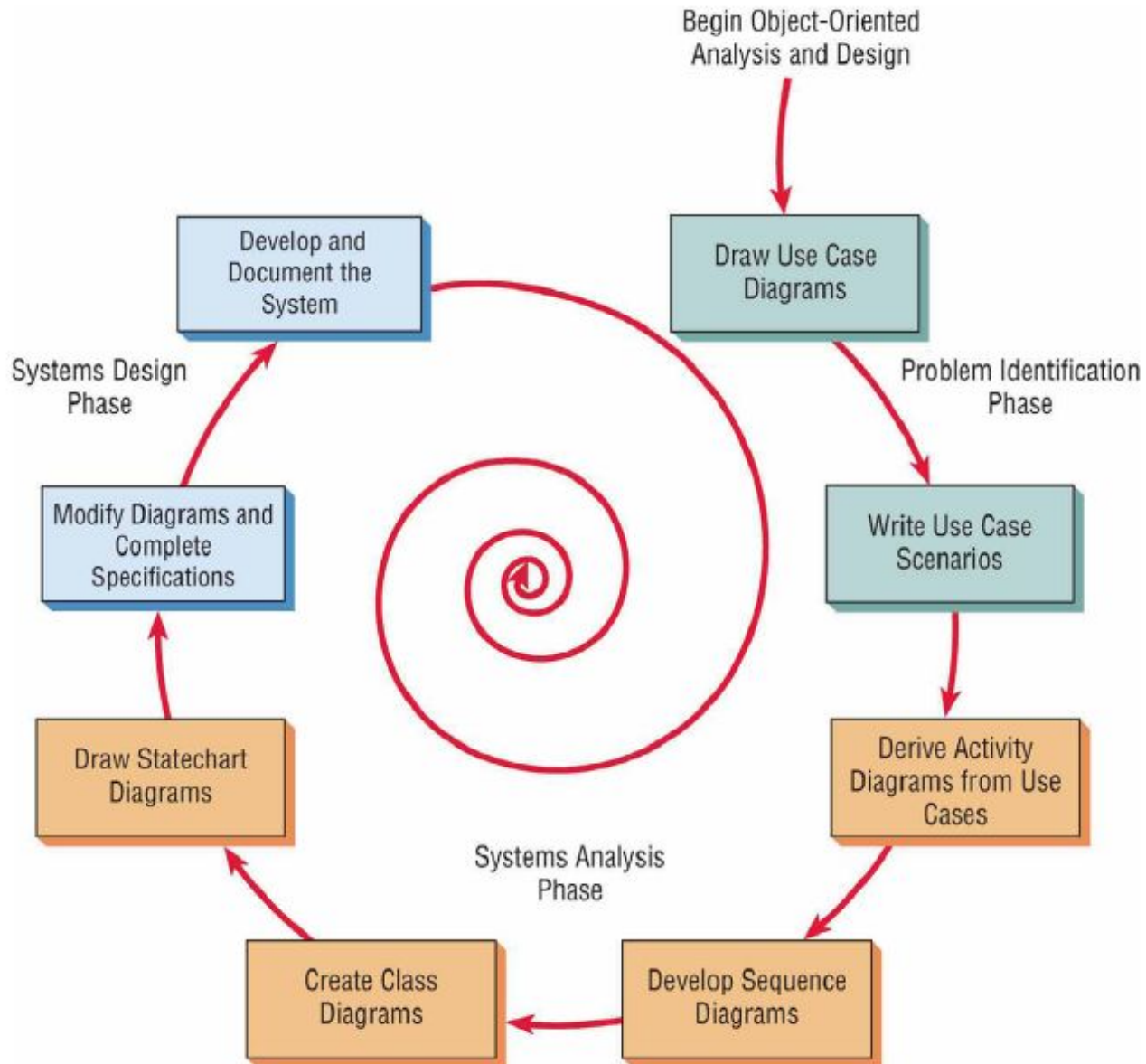
- It works well in situations in which complicated information systems are undergoing **continuous maintenance, adaptation, and redesign**.
- Object-oriented approaches use the industry standard for **modeling** object-oriented systems, called the unified modeling language (**UML**), to break down a system into a use case model.

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

- 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

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
- Examples: Linux Operating System, Apache Web Server, Mozilla Firefox

Four Types of Open Source Communities:

- Ad hoc
- Standardized
- Organized
- Commercial

Six Key Dimensions that Differentiate Open Source Communities

- General structure
- Environment
- Goals
- Methods
- User community
- Licensing

Reasons for Participating in Open Source Communities

- Rapidity with which new software can be developed and tested
- Faster to have a committed group of experts develop, test, and debug code
- This fosters creativity
- Have many good minds work with innovative applications

Reasons for Participating in Open Source Communities

- Potential to reduce development costs
- Bolster their self-image
- Contribute something worthwhile to the software development community

Open Source Contribution and Differentiation

- Contributions to the open community and differentiation from the open community are for the following reasons:
 - Cost
 - Managing resources
 - Time it takes to bring a new product to the market

Reasons for Analyst Participation in the Open Source Community

- Curiosity about software benefits
- Achieve collective design
 - Incorporate open source software design into:
 - Proprietary products
 - Processes
 - Knowledge
 - IT artifacts

Collective Design

- Through a process of collective design the IT artifact is imbued with
 - Community and organizational structures
 - Knowledge
 - Practices

Summary

- Information is a key resource
- 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
- Open source systems



This work is protected by United States copyright laws and is provided solely for the use of instructors in teaching their courses and assessing student learning. Dissemination or sale of any part of this work (including on the World Wide Web) will destroy the integrity of the work and is not permitted. The work and materials from it should never be made available to students except by instructors using the accompanying text in their classes. All recipients of this work are expected to abide by these restrictions and to honor the intended pedagogical purposes and the needs of other instructors who rely on these materials.

Copyright © 2014 Pearson Education, Inc.
Publishing as Prentice Hall