2

Chapter 2: Approaches to System Development

Systems Analysis and Design in a Changing World, 3rd Edition

2

Learning Objectives

- Explain the purpose and various phases of the systems development life cycle (SDLC)
- Explain the differences between a model, a tool, a technique, and a methodology
- Describe the two overall approaches used to develop information systems: the traditional method and the object-oriented method

2

Learning Objectives (continued)

- Describe some of the variations of the system development life cycle (SDLC)
- Describe the key features of current trends in system development: the spiral model, eXtreme Programming (XP), the Unified Process (UP), and Agile Modeling
- Explain how automated tools are used in system development

2

Overview

- Systems development life cycle (SDLC)
  - Provides overall framework for managing system development process
- Two main approaches to SDLC
  - Traditional approach: structured systems development and information engineering
  - Object-oriented approach: object technologies requires different approach to analysis, design, and programming
- All projects use some variation of SDLC

2

Systems Development Life Cycle (SDLC)

- Systems development project
  - Planned undertaking with fixed beginning and end
  - Produces desired result or product
  - Can be a large job of thousands of hours of effort or a small one month project
- Successful development project:
  - Provides a detailed plan to follow
  - Organized, methodical sequence of tasks and activities
  - Produces reliable, robust, and efficient system

2

Phases of the Systems Development Lifecycle (SDLC)

- Project planning: initiate, ensure feasibility, plan schedule, obtain approval for project
- Analysis: understand business needs and processing requirements
- Design: define solution system based on requirements and analysis decisions
- Implementation: construction, testing, user training, and installation of new system
- Support: keep system running and improve

Information System Development Phases

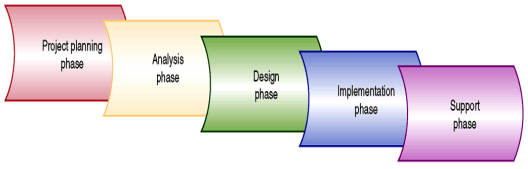


FIGURE 2-1
Information system development phases.

SDLC and problem-solving

- Similar to problem-solving approach
  - Organization recognizes problem (Project Planning)
  - Project team investigates, understands problem and solution requirements (Analysis)
  - Solution is specified in detail (Design)
  - System that solves problem built and installed (Implementation)
  - System used, maintained, and enhanced to continue to provide intended benefits (Support)

Planning Phase of SDLC

- Define business problem and scope
- Produce detailed project schedule
- Confirm project feasibility
  - Economic, organizational, technical, resource, and schedule
- Staff the project (resource management)
- Launch project → official announcement

Analysis Phase of SDLC

- Gather information to learn problem domain
- Define system requirements
- Build prototypes for discovery of requirements
- Prioritize requirements
- Generate and evaluate alternatives
- Review recommendations with management

Design Phase of SDLC

- Design and integrate the network
- Design the application architecture
- Design the user interfaces
- Design the system interfaces
- Design and integrate the database
- Prototype for design details
- Design and integrate system controls

Implementation Phase of SDLC

- Construct software components
- Verify and test
- Convert data
- Train users and document the system
- Install the system

Support Phase of SDLC

- Maintain system
  - Small patches, repairs, and updates
- Enhance system
  - Small upgrades or enhancements to expand system capabilities
  - Larger enhancements may require separate development project
- Support users
  - Help desk and/or support team

Scheduling Project Phases

- Waterfall approach – each phase falls into next phase
  - Freeze planning specifications before analysis
  - Freeze analysis specifications before design
  - Once go over the waterfall for each phase, do not go back
- Overlapping (or concurrent) phases
  - Waterfall is not realistic, we are not perfect
  - Overlaps can be more efficient than waterfall

Scheduling Project Phases (continued)

- Iteration - Work activities are repeated
  - Each iteration refines previous result
  - Approach assumes no one gets it right the first time
  - There are a series of mini projects for each iteration
- Example: Outline, rough draft, edited result
- Example: Blueprint, frame, completed house

The waterfall approach to the SDLC

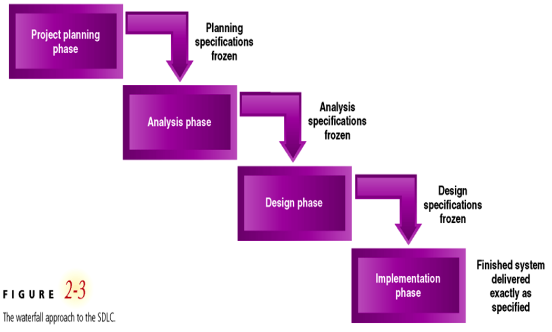


FIGURE 2-3
The waterfall approach to the SDLC

Overlap of Systems Development Activities

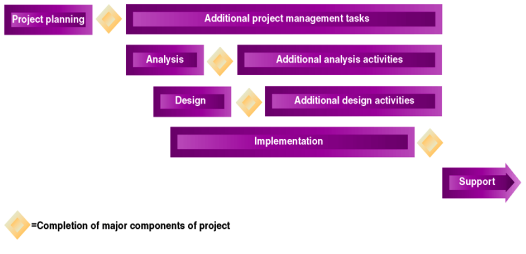
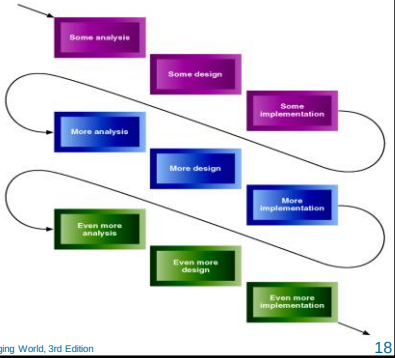


FIGURE 2-4
The overlap of system development phases.

Iterations across life cycle phases

FIGURE 2-5
Iterations across life cycle phases.



Methodologies and Models

- Methodologies
  - Comprehensive guidelines to follow for completing every SDLC activity
  - Collection of models, tools, and techniques
- Models
  - Representation of an important aspect of real world, but not same as real thing
  - Abstraction used to separate out aspect
  - Diagrams and charts
  - Project planning and budgeting aids

Some Models Used in System Development

FIGURE 2-6
Some models used in system development.

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

- PERT chart
- Gantt chart
- Organizational hierarchy chart
- Financial analysis models – NPV, ROI

Tools and Techniques

- Tools
  - Software support that helps create models or other required project components
  - Range from simple drawing programs to complex CASE tools
- Techniques
  - Collection of guidelines that help analyst complete system development activity or task
  - Can be step-by-step instructions or just general advice

Some Tools Used in System Development

FIGURE 2-7
Some tools used in system development.

- Project management application
- Drawing/graphics application
- Word processor/text editor
- Computer-aided system engineering (CASE) tools
- Integrated development environment (IDE)
- Database management application
- Reverse-engineering tool
- Code generator tool

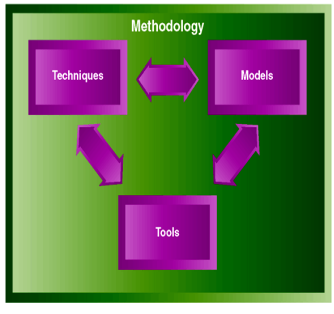
Some Techniques Used in System Development

FIGURE 2-8
Some techniques used in system development.

- Strategic planning techniques
- Project management techniques
- User interviewing techniques
- Data-modeling techniques
- Relational database design techniques
- Structured analysis technique
- Structured design technique
- Structured programming technique
- Software-testing techniques
- Object-oriented analysis and design techniques

Relationships Among Components of a Methodology

FIGURE 2-9
Relationships among components of a methodology.



Two Approaches to System Development

- Traditional Approach
  - Also called structured system development
  - Structured analysis and design technique (SADT)
- Structured programming
  - Improves computer program quality
  - Allows other programmers to easily read and modify code
  - Each program module has one beginning and one ending
  - Three programming constructs (sequence, decision, repetition)

Three Structured Programming Constructs

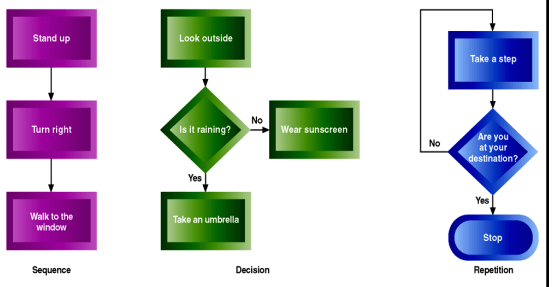


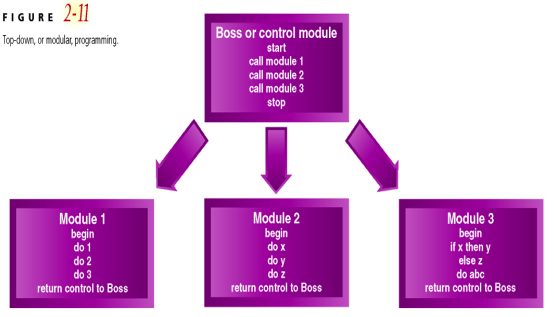
FIGURE 2-10
Three structured programming constructs.

Top-Down Programming

- Divides complex programs into hierarchy of modules
- The module at top controls execution by "calling" lower level modules
- Modular programming
  - Similar to top-down programming
- One program calls other programs to work together as single system

Top-Down or Modular Programming

FIGURE 2-11
Top-down, or modular, programming.

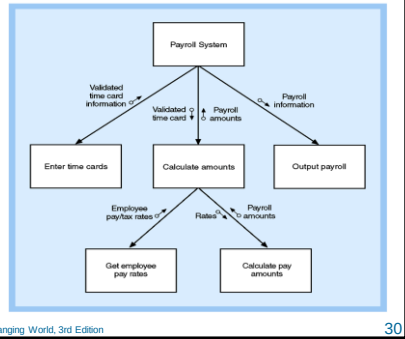


Structured Design

- Technique developed to provide design guidelines
  - What set of programs should be
  - What program should accomplish
  - How programs should be organized into a hierarchy
- Modules are shown with structure chart
- Main principle of program modules
  - Loosely coupled – module is independent of other modules
  - Highly cohesive – module has one clear task

Structure Chart Created Using Structured Design Technique

FIGURE 2-12
A structure chart created using the structured design technique.



Structured Analysis

- Define what system needs to do (processing requirements)
- Define data system needs to store and use (data requirements)
- Define inputs and outputs
- Define how functions work together to accomplish tasks
- Data flow diagrams and entity relationship diagrams show results of structured analysis

Data Flow Diagram (DFD) created using Structured Analysis Technique

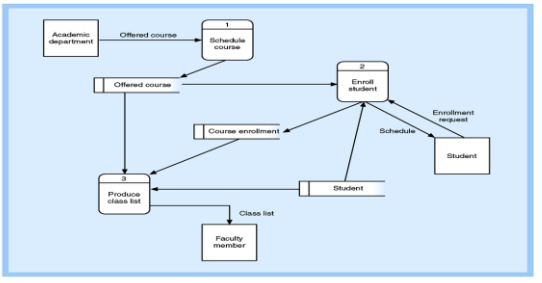


FIGURE 2-13
A data flow diagram (DFD) created using the structured analysis technique.

Entity-Relationship Diagram (ERD) created using the Structured Analysis technique

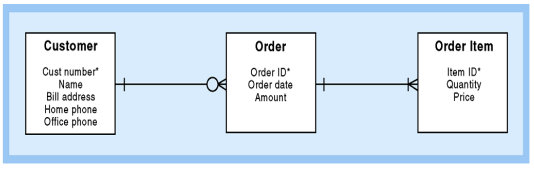
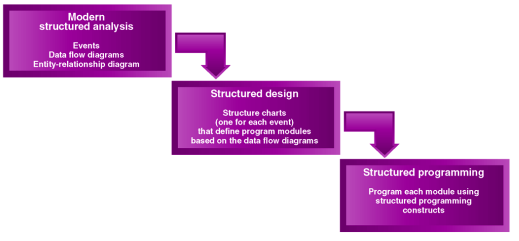


FIGURE 2-14
An entity-relationship diagram (ERD) created using the structured analysis technique.

Structured Analysis Leads to Structured Design and Structured Programming

FIGURE 2-15
How structured analysis leads to structured design and to structured programming.



Information Engineering (IE)

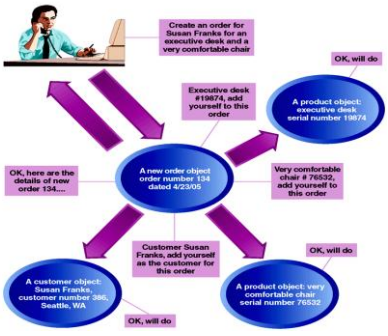
- Refinement to structured development
- Methodology with strategic planning, data modeling, automated tools focus
- More rigorous and complete than SADT
- Uses process dependency diagram
- Industry merged key concepts from structured development and information engineering approaches into traditional approach

Object-Oriented Approach

- Views information system as collection of interacting objects that work together to accomplish tasks
  - Objects - things in computer system that can respond to messages
  - No processes, programs, data entities, or files are defined – just objects
- Object-oriented analysis (OOA)
  - Defines types of objects that do work of system
  - Shows how objects interact with users to complete tasks

Object-Oriented Approach to Systems

FIGURE 2-16
The object-oriented approach to systems (read clockwise starting with user).



Object-Oriented Approach (continued)

- Object-oriented design (OOD)
  - Defines object types needed to communicate with people and devices in system
  - Shows how objects interact to complete tasks
  - Refines each type of object for implementation with specific language of environment
- Object-oriented programming (OOP)
  - Writing statements in programming language to define what each type of object does
- Benefits of OOA include naturalness and reuse

Class Diagram Created During OO Analysis

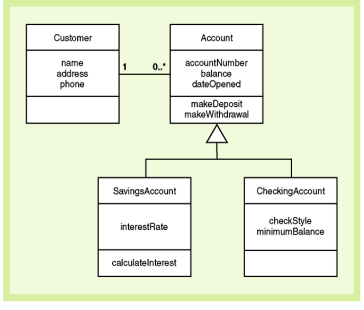


FIGURE 2-17
A class diagram created during object-oriented analysis.

SDLC Variations

- Many variations of SDLC in practice
  - No matter which one, tasks are similar
- Based on variation of names for phases
  - SDLC compared to IE compared to UP
- Based on emphasis on people
  - User-centered design, participatory design
- Based on speed of development
  - Rapid application development (RAD)
  - Prototyping

Life Cycles with Different Names for Phases

FIGURE 2-18
Life cycles with different names for phases.

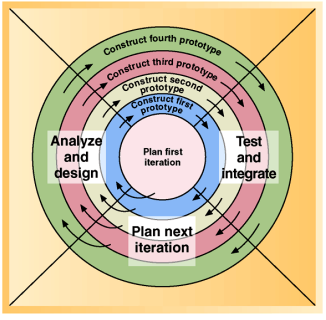
	Early Example of an SDLC	Information Engineering	Unified Process (UP)	SDLC with Activity Names for Phases
Planning Phase	Feasibility study	Information strategy planning	Inception phase	Organize the project and study feasibility
				Study and analyze the current system
Analysis Phase	System investigation	Business area analysis		Model and prioritize the functional requirements
	Systems analysis		Elaboration phase	Generate alternatives and propose the best solution
Design Phase	Systems design	Business system design		Design the system
		Technical design	Construction phase	Obtain needed hardware and software
Implementation Phase	Implementation	Construction		Build and test the new system
		Transition	Transition phase	Install and operate the new system
Support Phase	Review and maintenance	Production		

Current Trends in Development

- Spiral Model
  - Highly iterative approach
  - Works around the phases (analysis, design, construction, testing, integration with previous prototype component) in a spiral until project is complete
  - Initial planning is to do just enough analysis to build initial prototype
  - Each iteration in the spiral addresses greatest risk

The Spiral Life Cycle Model

FIGURE 2-20
The spiral life cycle model.



Extreme Programming (XP)

- Recent, lightweight, development approach to keep process simple and efficient
- Describes system support needed and required system functionality through informal user stories
- Has users describe acceptance tests to demonstrate defined outcomes
- Relies on continuous testing and integration, heavy user involvement, programming done by small teams

The Unified Process (UP)

- Object-oriented development approach
- Offered by IBM / Rational
  - Booch, Rumbaugh, Jacobson
- Unified Modeling Language (UML) used primarily for modeling
- UML can be used with any OO methodology
- UP defines 4 life cycle phases
  - Inception, elaboration, construction, transition

The Unified Process (UP) (continued)

- Reinforces six best practices
  - Develop iteratively
  - Define and manage system requirements
  - Use component architectures
  - Create visual models
  - Verify quality
  - Control changes

Agile Modeling

- Hybrid of XP and UP (Scott Ambler) has more models than XP, less documents than UP
- Interactive and Incremental Modeling:
  - Apply right models
  - Create several models in parallel
  - Model in small increments
- Teamwork:
  - Get active stakeholder participation
  - Encourage collective ownership
  - Model with others and display models publicly

Agile Modeling (continued)

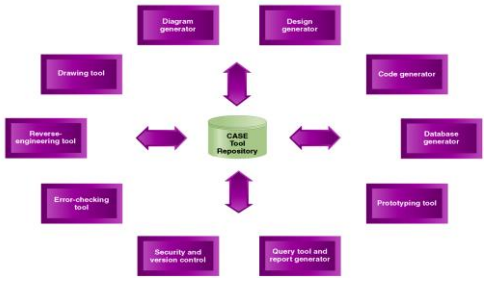
- Simplicity:
  - Use simple content
  - Depict models simply
  - Use simplest modeling tools
- Validation
  - Consider testability
  - Prove model is right with code

Tools to Support System Development

- Computer-Aided System Engineering (CASE)
  - Automated tools to improve the speed and quality of system development work
  - Contains database of information about system called repository
- Upper CASE - support for analysis and design
- Lower CASE - support for implementation
- ICASE - integrated CASE tools

CASE Tool Repository Contains all System Information

FIGURE 2-21
A CASE tool repository contains all information about the system.



Summary

- Systems development projects are organized around the SDLC
- SDLC Phases include project planning, analysis, design, implementation, and support to be completed for each project
- Systems developers learn SDLC based on the sequential waterfall approach
- In practice, phases overlap and projects contain many iterations of analysis, design, and implementation activities

Summary (continued)

- All development approaches use a SDLC to manage the project.
- Models, techniques, and tools make up a systems development methodology
- System development methodologies are based on traditional approach or object-oriented approach
- System development methodology provides guidelines to complete every activity in the SDLC

Summary (continued)

- Original SDLC was waterfall approach
- Most SDLC use iteration across phases
- Rapid application development (RAD) goal is to speed up development
- Current trends include: spiral model, eXtreme Programming (XP), Unified Process (UP) and Agile Modeling
- CASE tools are designed to help analysts complete tasks