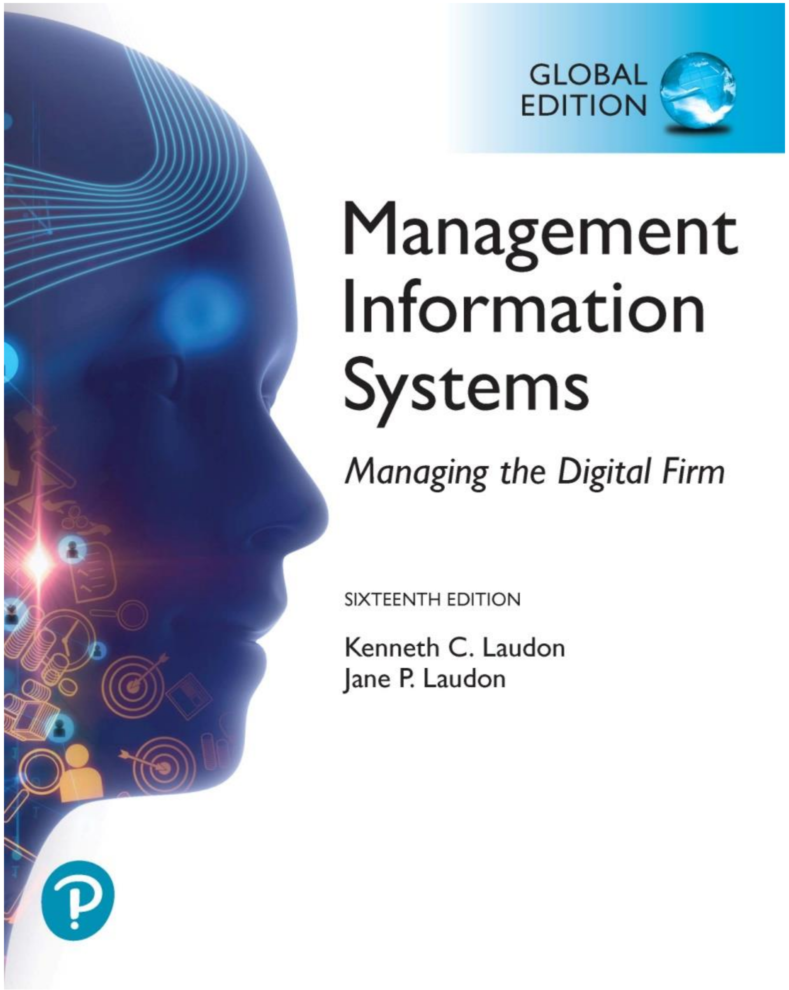
Management Information Systems: Managing the Digital Firm

Sixteenth Edition • Global Edition



Chapter 13

Building Information Systems

Pearson

Learning Objectives

13.1 How does building new systems produce organizational change?

13.2 What are the core activities in the systems development process?

13.3 What are the principal methodologies for modeling and designing systems?

13.4 What are alternative methods for building information systems?

13.5 What are new approaches for system building in the digital firm era?

13.6 How will MIS help my career?

Systems Development and Organizational Change (1 of 2)

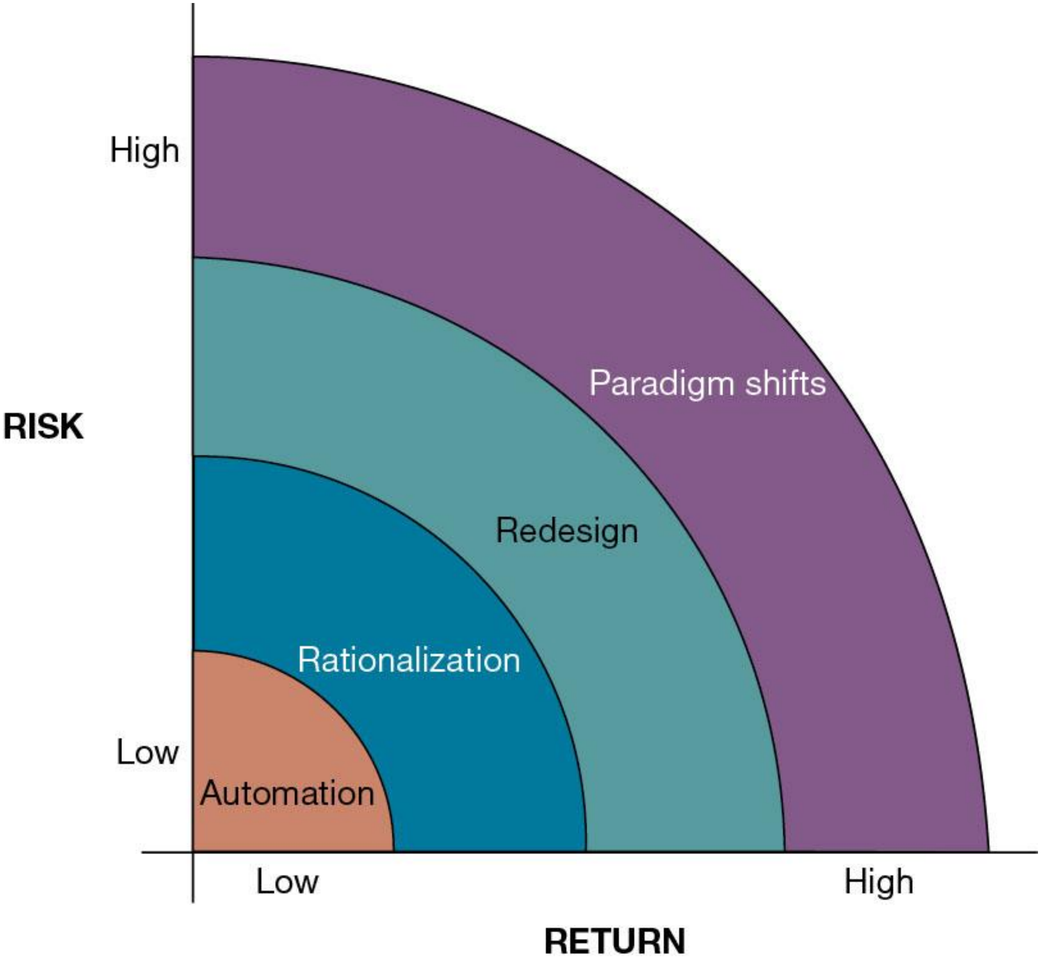
- IT-enabled organizational change
- Automation
  - Increases efficiency
  - Replaces manual tasks
- Rationalization of procedures
  - Streamlines standard operating procedures
  - Often found in programs for making continuous quality improvements
    - Total quality management (TQM)
    - Six sigma

Pearson

Systems Development and Organizational Change (2 of 2)

- Business process redesign
  - Analyze, simplify, and redesign business processes
  - Reorganize workflow, combine steps, eliminate repetition
- Paradigm shifts
  - Rethink nature of business
  - Define new business model
  - Change nature of organization

Figure 13.1 Organizational Change Carries Risks and Rewards



Business Process Redesign

- Business process management (BPM)
  - Variety of tools, methodologies to analyze, design, optimize processes
  - Used by firms to manage business process redesign

- Steps in BPM
  1. Identify processes for change
  2. Analyze existing processes
  3. Design the new process
  4. Implement the new process
  5. Continuous measurement

Figure 13.2 As-is Business Process for Purchasing a Book from a Physical Bookstore

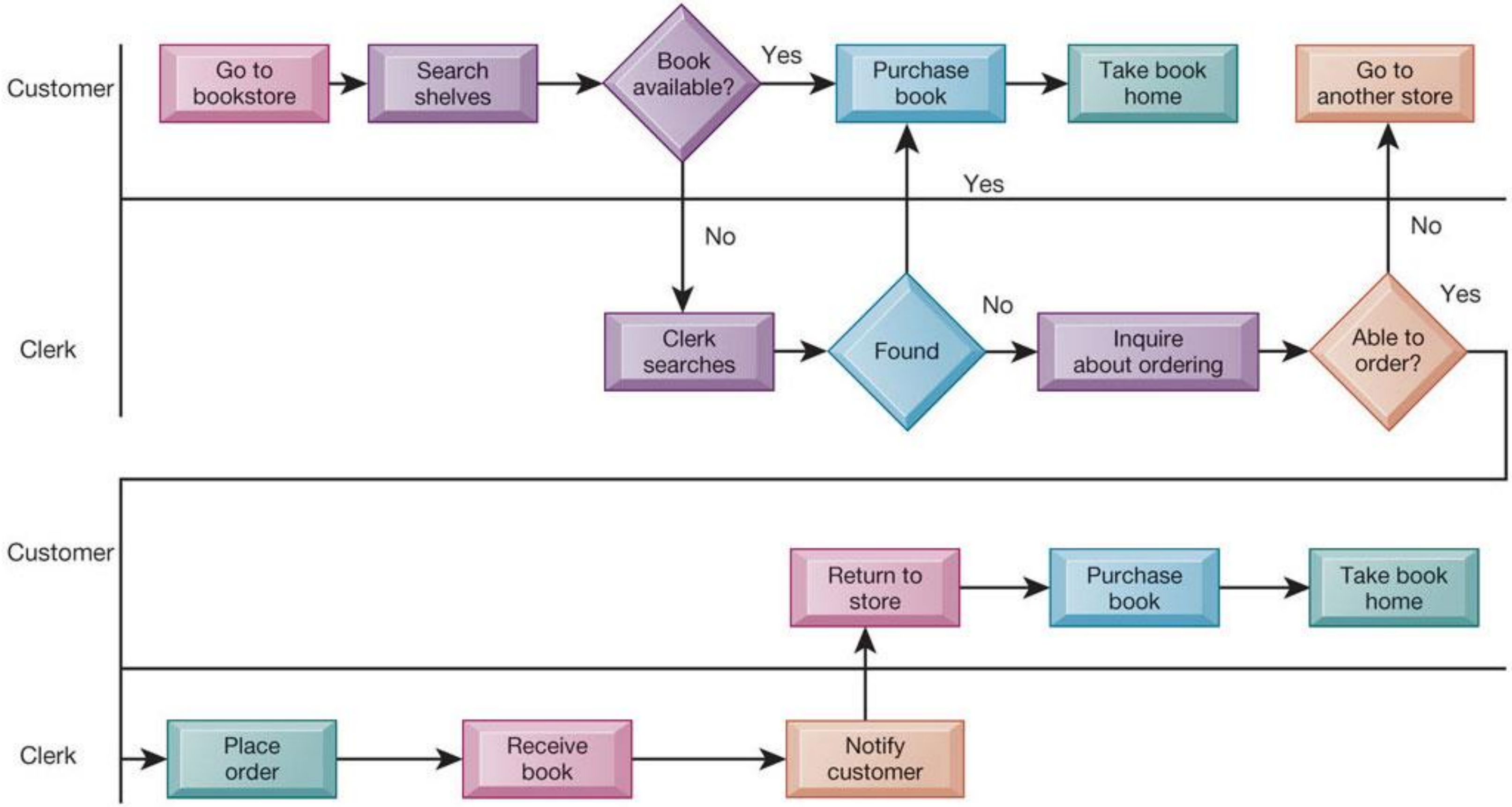
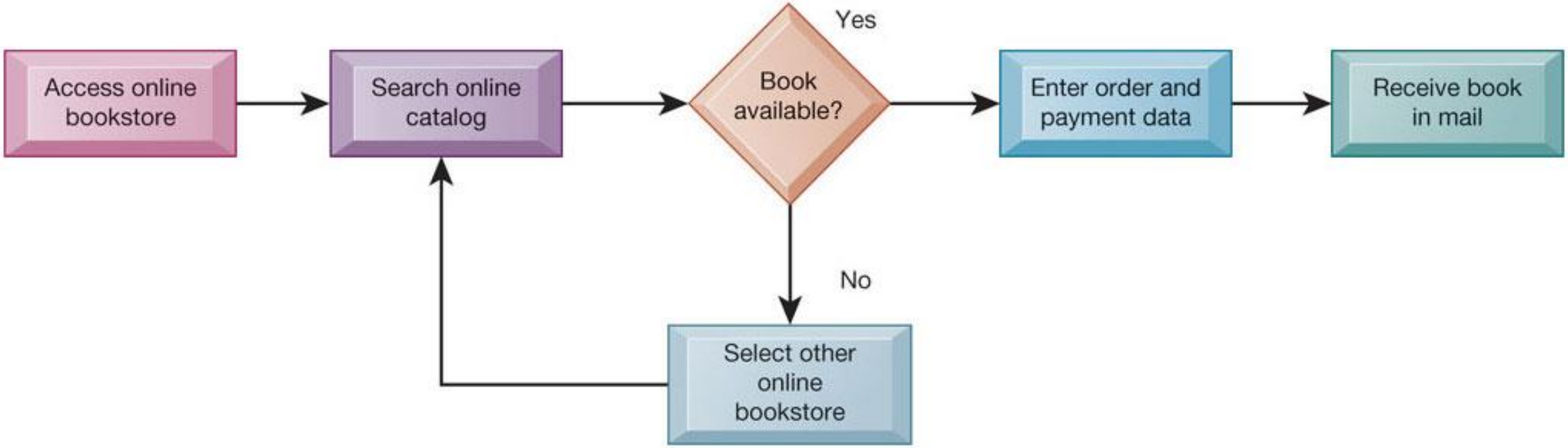


Figure 13.3
Redesigned Process for Purchasing a Book Online



Tools for Business Process Management

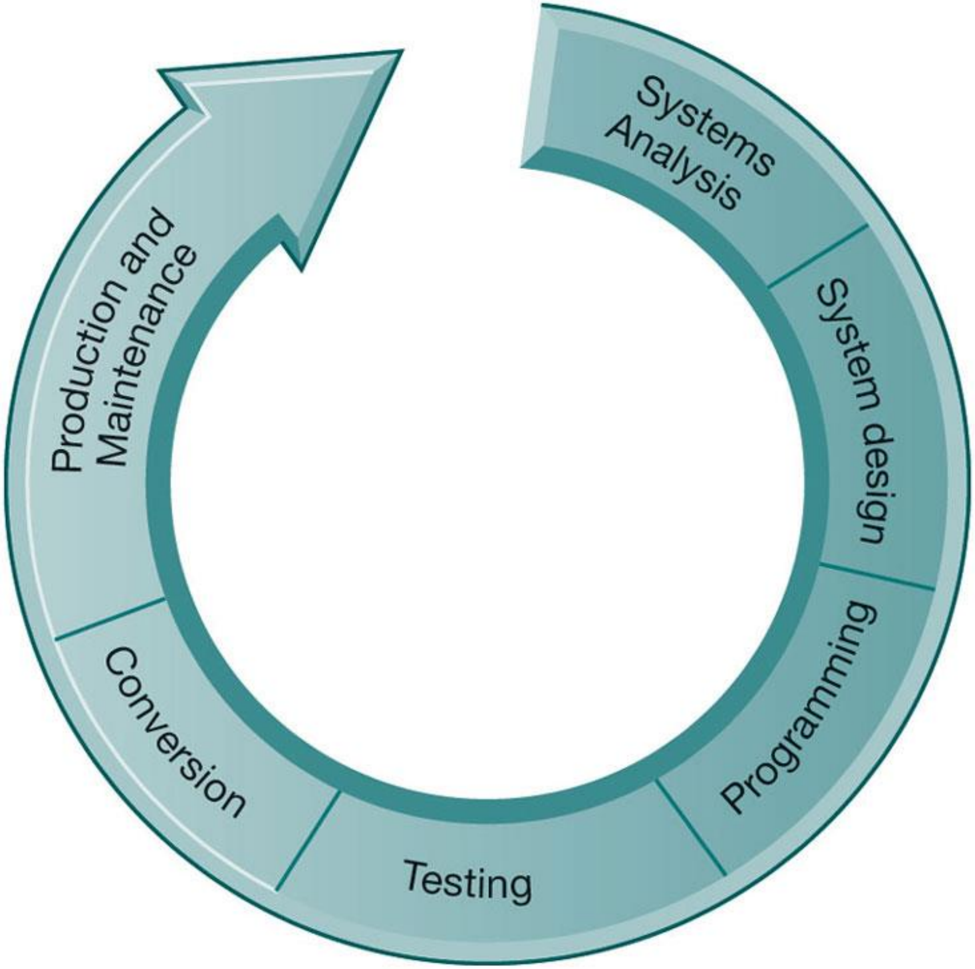
- Identify and document existing processes
  - Identify inefficiencies
- Create models of improved processes
- Capture and enforce business rules for performing, automating processes
- Integrate existing systems to support process improvements
- Verify that new processes have improved
- Measure impact of process changes on key business performance indicators

Systems Development

- Activities that go into producing an information system solution to an organizational problem or opportunity
  - Systems analysis
  - Systems design
  - Programming
  - Testing
  - Conversion
  - Production and maintenance

Pearson

Figure 13.4 The Systems Development Process



Systems Analysis

- Analysis of problem to be solved by new system
  - Defining the problem
  - Identifying causes
  - Specifying solutions
  - Identifying information requirements
- Feasibility study
- Systems proposal report
- Information requirements
  - Faulty requirements analysis is a leading cause of systems failure and high systems development costs

Systems Design

- Describes system specifications that will deliver functions identified during systems analysis
- Should address all managerial, organizational, and technological components of system solution
- Role of end users
  - User information requirements drive system building
  - Users must have sufficient control over design process to ensure system reflects their business priorities and information needs
  - Insufficient user involvement in design effort is major cause of system failure

Table 13.1 System Design Specifications (1 of 2)

Category	Specifications
Output	Medium, Content, Timing
Input	Origins, Flow, Data entry
User Interface	Simplicity, Efficiency, Logic, Feedback, Errors
Database Design	Logical data model, Volume and speed requirements, File organization and design, Record specifications
Processing	Computations, Program modules, Required reports, Timing of outputs
Manual Procedures	What activities, Who performs them, When, How, Where
Controls	Input controls (characters, limit, reasonableness), Processing controls (consistency, record counts), Output controls (totals, samples of output), Procedural controls (passwords, special forms)

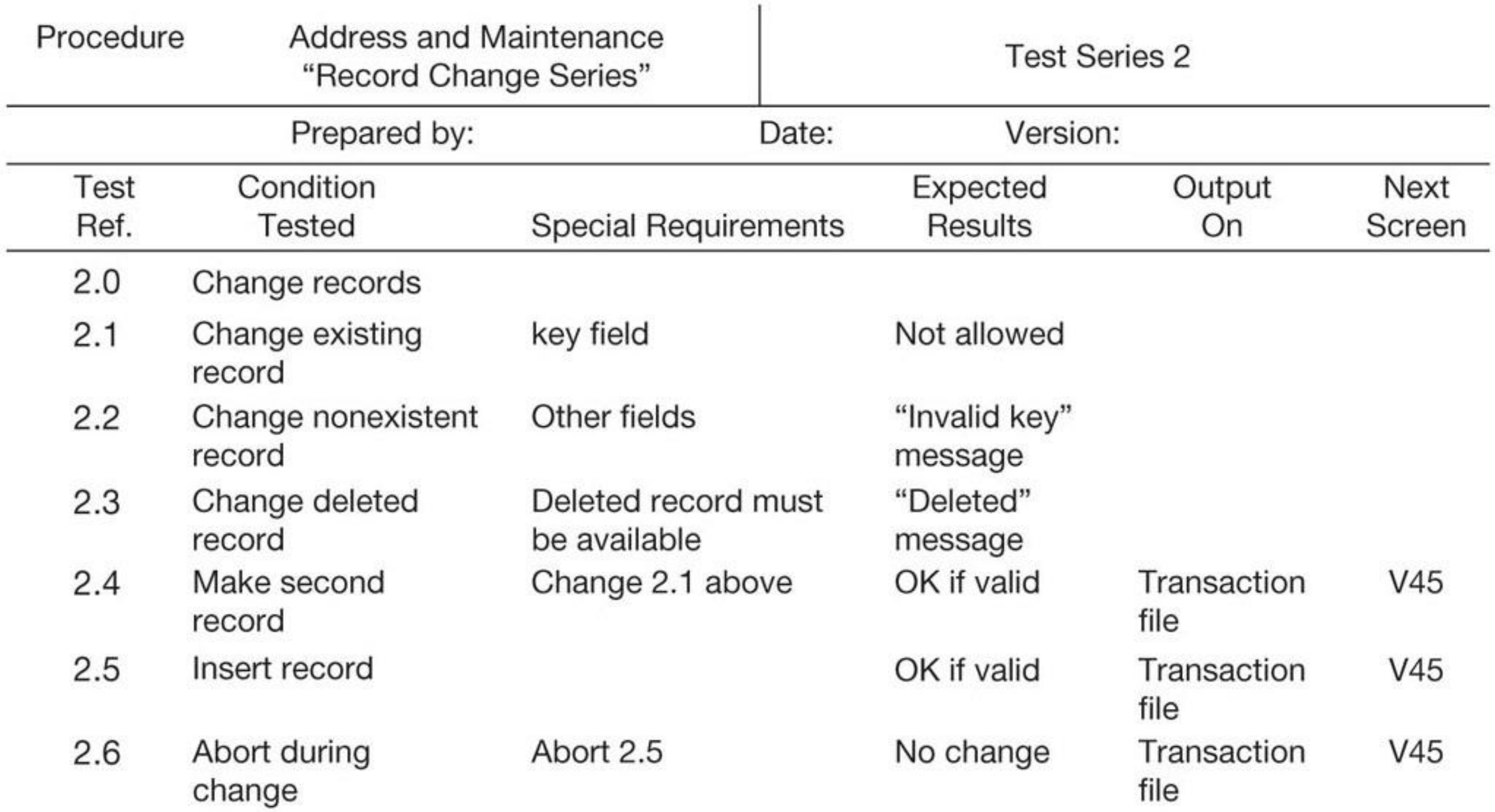
Table 13.1 System Design Specifications (2 of 2)

Category	Specifications
Security	Access controls, Catastrophe plans, Audit trails
Documentation	Operations documentation, Systems documents, User documentation
Conversion	Transfer files, Initiate new procedures, Select testing method Cut over to new system
Training	Select training techniques, Develop training modules, Identify training facilities
Organizational Changes	Task redesign, Job redesign, Process design, Organization structure design, Reporting relationships

Completing the Systems Development Process (1 of 3)

- Programming
  - System specifications from design stage are translated into software program code
- Testing
  - Ensures system produces right results
  - Unit testing: Tests each program in system separately
  - System testing: Test functioning of system as a whole
  - Acceptance testing: Makes sure system is ready to be used in production setting
  - Test plan: All preparations for series of tests

Figure 13.5
A Sample Test Plan to Test a Record Change



Procedure Address and Maintenance “Record Change Series” | Test Series 2

Prepared by: Date: Version:

Test Ref.	Condition Tested	Special Requirements	Expected Results	Output On	Next Screen
2.0	Change records				
2.1	Change existing record	key field	Not allowed		
2.2	Change nonexistent record	Other fields	“Invalid key” message		
2.3	Change deleted record	Deleted record must be available	“Deleted” message		
2.4	Make second record	Change 2.1 above	OK if valid	Transaction file	V45
2.5	Insert record		OK if valid	Transaction file	V45
2.6	Abort during change	Abort 2.5	No change	Transaction file	V45

Completing the Systems Development Process (2 of 3)

- Conversion
  - Process of changing from old system to new system
  - Four main strategies
    - Parallel strategy
    - Direct cutover
    - Pilot study
    - Phased approach
  - Requires end-user training
  - Finalization of detailed documentation showing how system works from technical and end-user standpoint

Completing the Systems Development Process (3 of 3)

- Production and maintenance
  - System reviewed to determine if revisions needed
  - May include post-implementation audit document
  - Maintenance
    - Changes in hardware, software, documentation, or procedures to a production system to correct errors, meet new requirements, or improve processing efficiency
      - 20 percent debugging, emergency work
      - 20 percent changes to hardware, software, data, reporting
      - 60 percent of work: user enhancements, improving documentation, recoding for greater processing efficiency

Table 13.2 Systems Development

Core Activity	Core Activity
Systems analysis	Identify problem(s), Specify solutions, Establish information requirements
Systems design	Create design specifications
Programming	Translate design specifications into program code
Testing	Perform unit testing, Perform systems testing, Perform acceptance testing
Conversion	Plan conversion, Prepare documentation, Train users and technical staff
Production and maintenance	Operate the system, Evaluate the system, Modify the system

Structured Methodologies (1 of 2)

- Structured: Techniques are step-by-step, progressive
- Process-oriented: Focusing on modeling processes or actions that manipulate data
- Separate data from processes
- Data flow diagram (DFD)
  - Represents system’s component processes and flow of data between them
  - Logical graphic model of information flow

Structured Methodologies (2 of 2)

- Data dictionary
  - Defines contents of data flows and data stores
- Process specifications
  - Describe transformation occurring within lowest level of data flow diagrams
- Structure chart
  - Top-down chart, showing each level of design, relationship to other levels, and place in overall design structure

Figure 13.6
Data Flow Diagram for Mail-in University Registration System

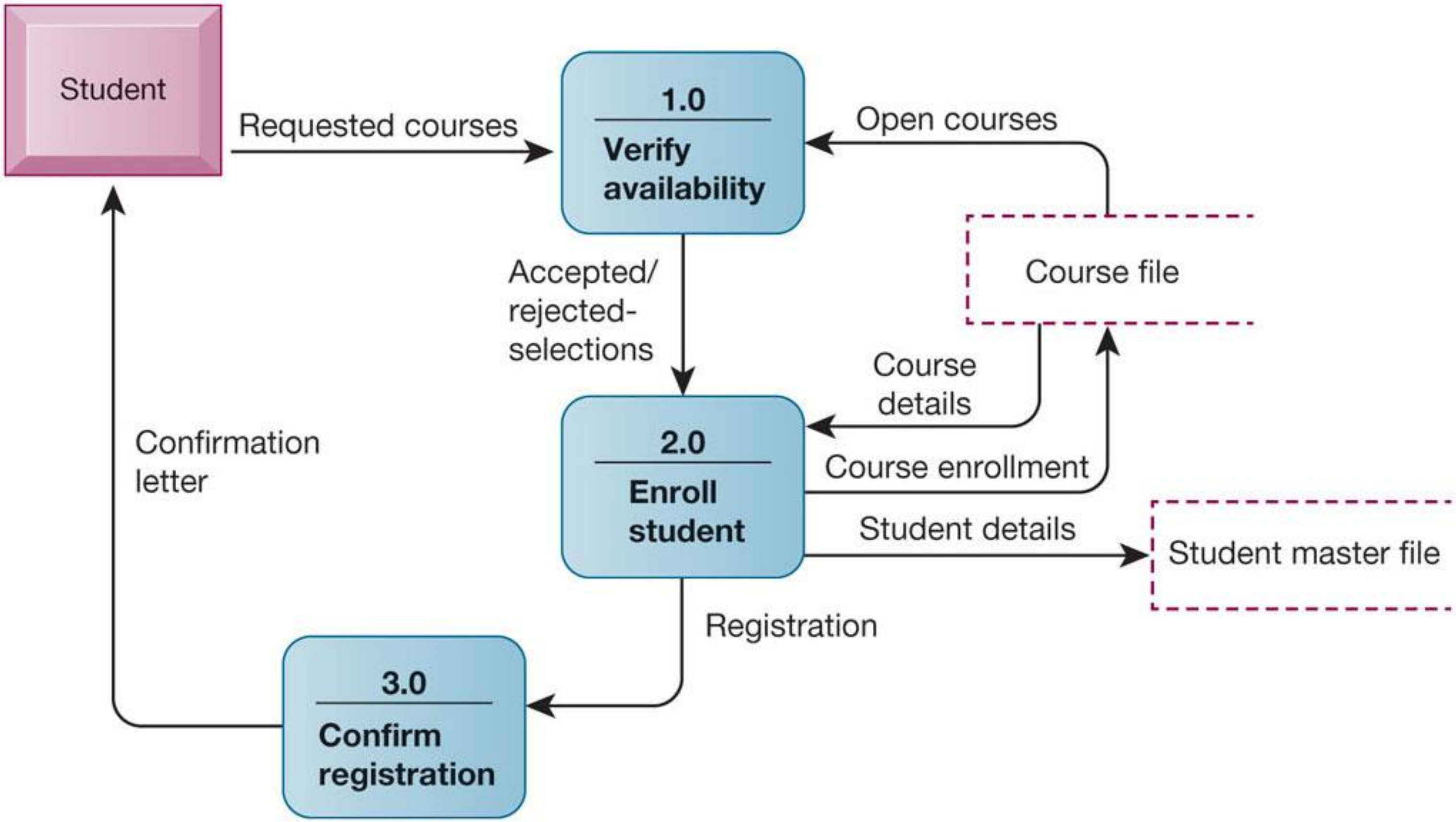
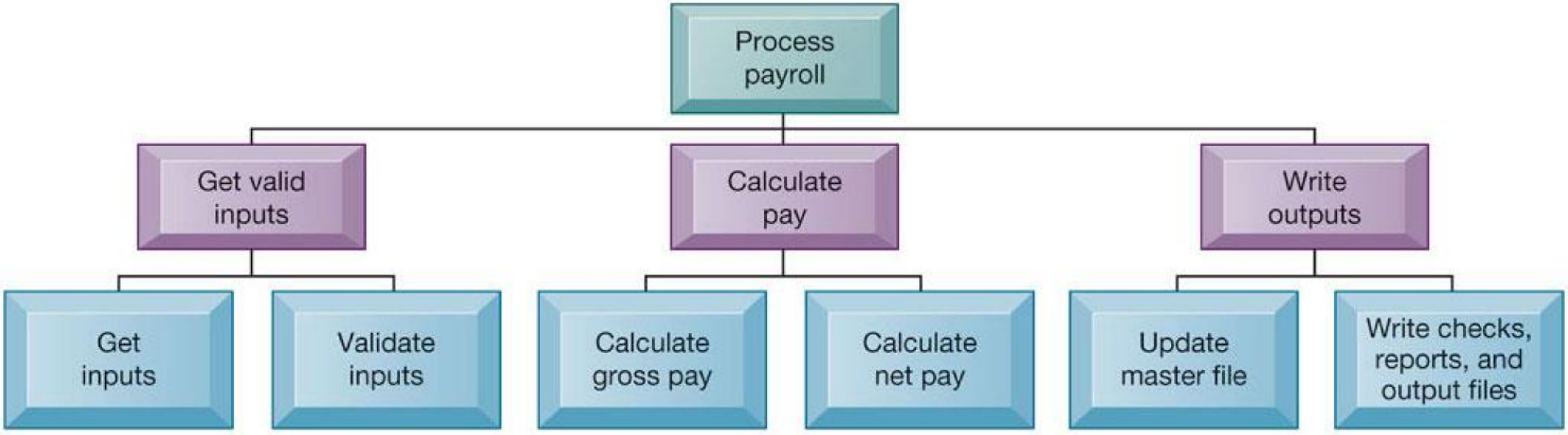


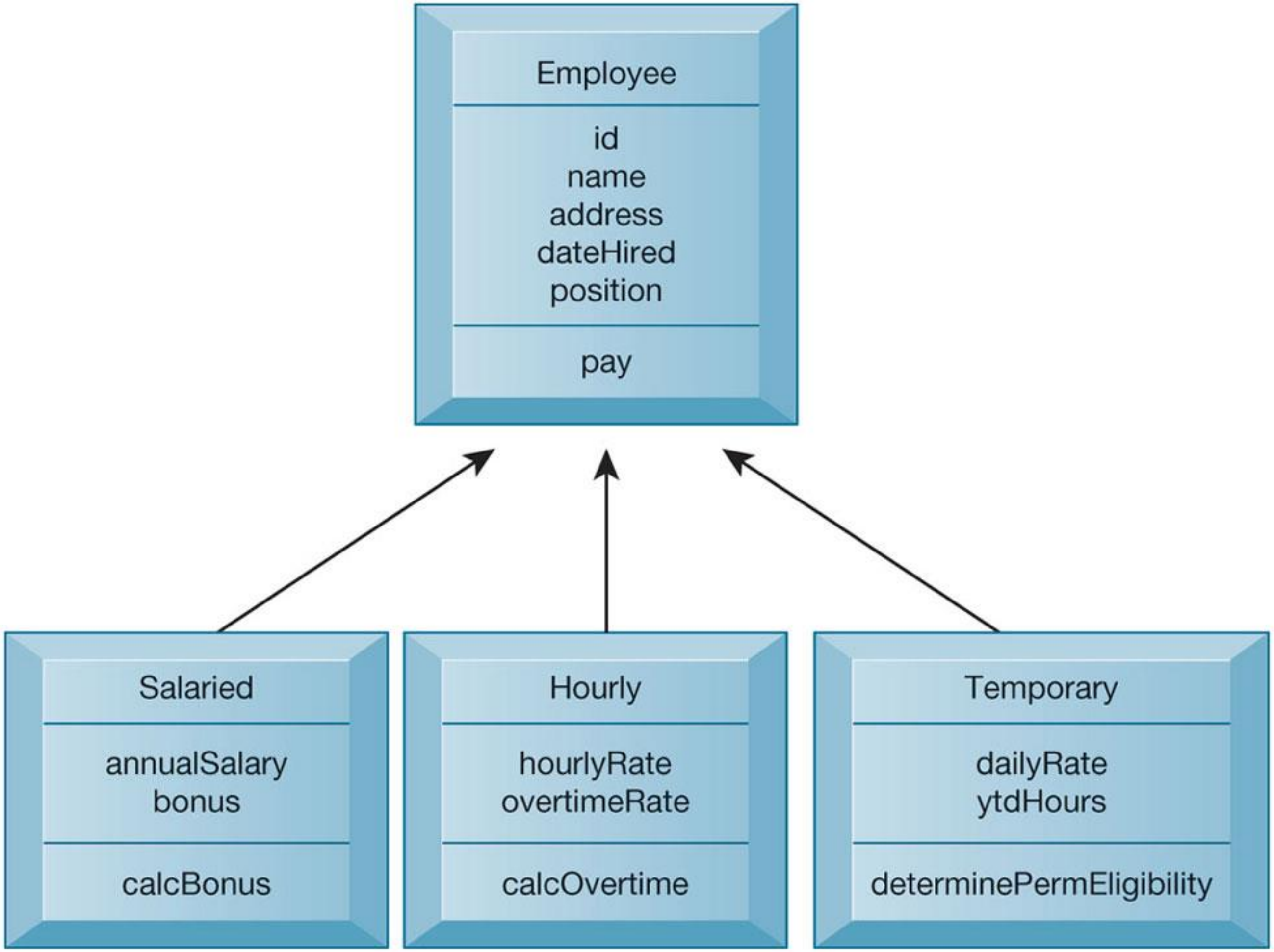
Figure 13.7
High-level Structure Chart for a Payroll System



Object-Oriented Development (1 of 2)

- Object
  - Basic unit of systems analysis and design
  - Combines data and the processes that operate on those data
  - Data in object can be accessed only by operations associated with that object
- Object-oriented modeling
  - Based on concepts of class and inheritance
  - Objects belong to a certain class and have features of that class
  - May inherit structures and behaviors of a more general, ancestor class

Figure 13.8 Class and Inheritance



Object-Oriented Development (2 of 2)

- More iterative and incremental than traditional structured development
  - Systems analysis: Interactions between system and users analyzed to identify objects
  - Design phase: Describes how objects will behave and interact; grouped into classes, subclasses, and hierarchies
  - Implementation: Some classes may be reused from existing library of classes, others created or inherited
- Objects are reusable
  - Object-oriented development can potentially reduce time and cost of development

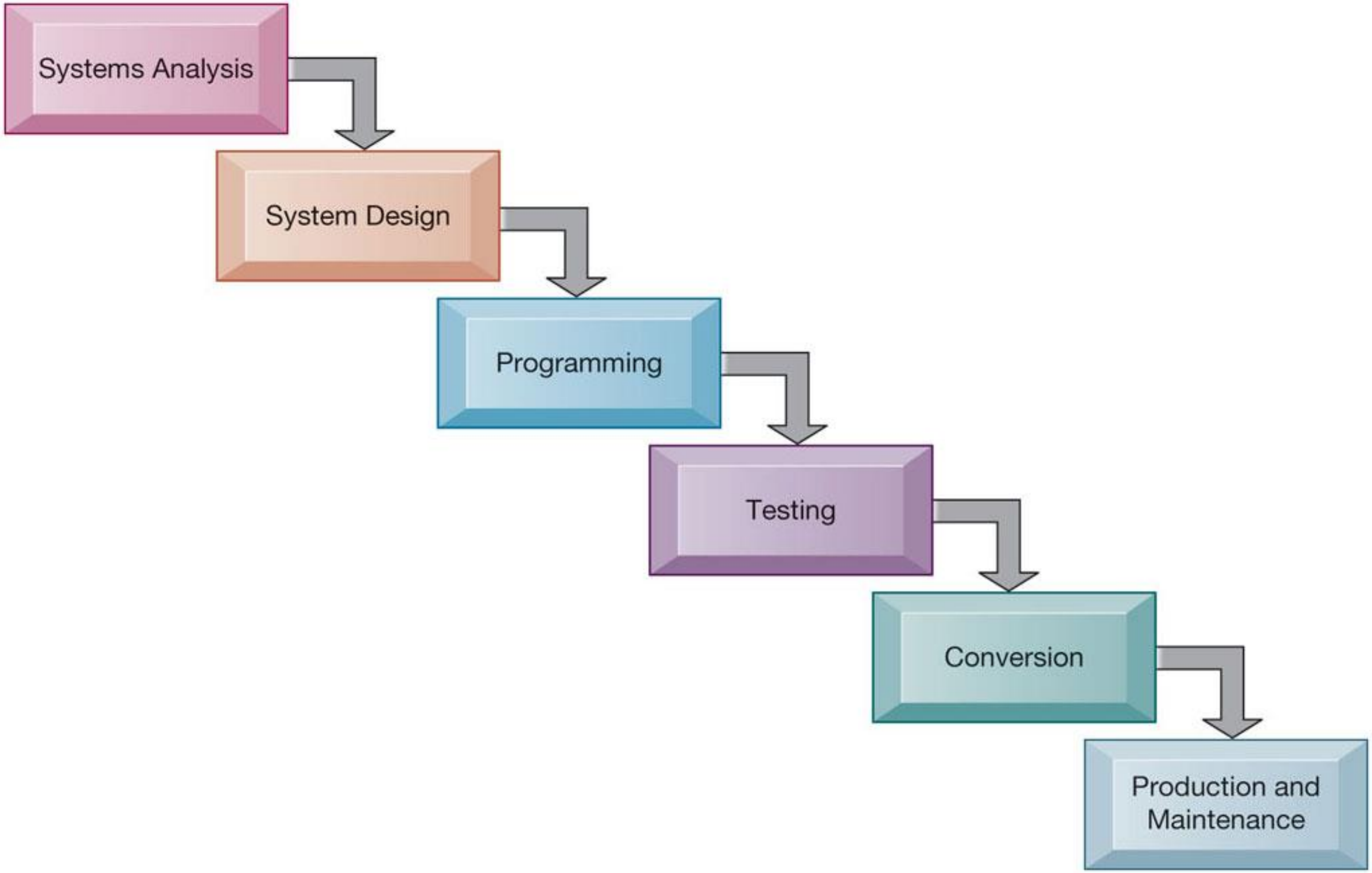
Computer-Aided Software Engineering

- Software tools to automate development and reduce repetitive work, including:
  - Graphics facilities for producing charts and diagrams
  - Screen and report generators, reporting facilities
  - Analysis and checking tools
  - Data dictionaries
  - Code and documentation generators
- Support iterative design by automating revisions and changes and providing prototyping facilities
- Require organizational discipline to be used effectively

Traditional Systems Life Cycle

- Oldest method for building information systems
- Phased approach
  - Development divided into formal stages
  - “Waterfall” approach: One stage finishes before next stage begins
- Formal division of labor between end users and information systems specialists
- Emphasizes formal specifications and paperwork
- Still used for building large complex systems
- Can be costly, time-consuming, and inflexible

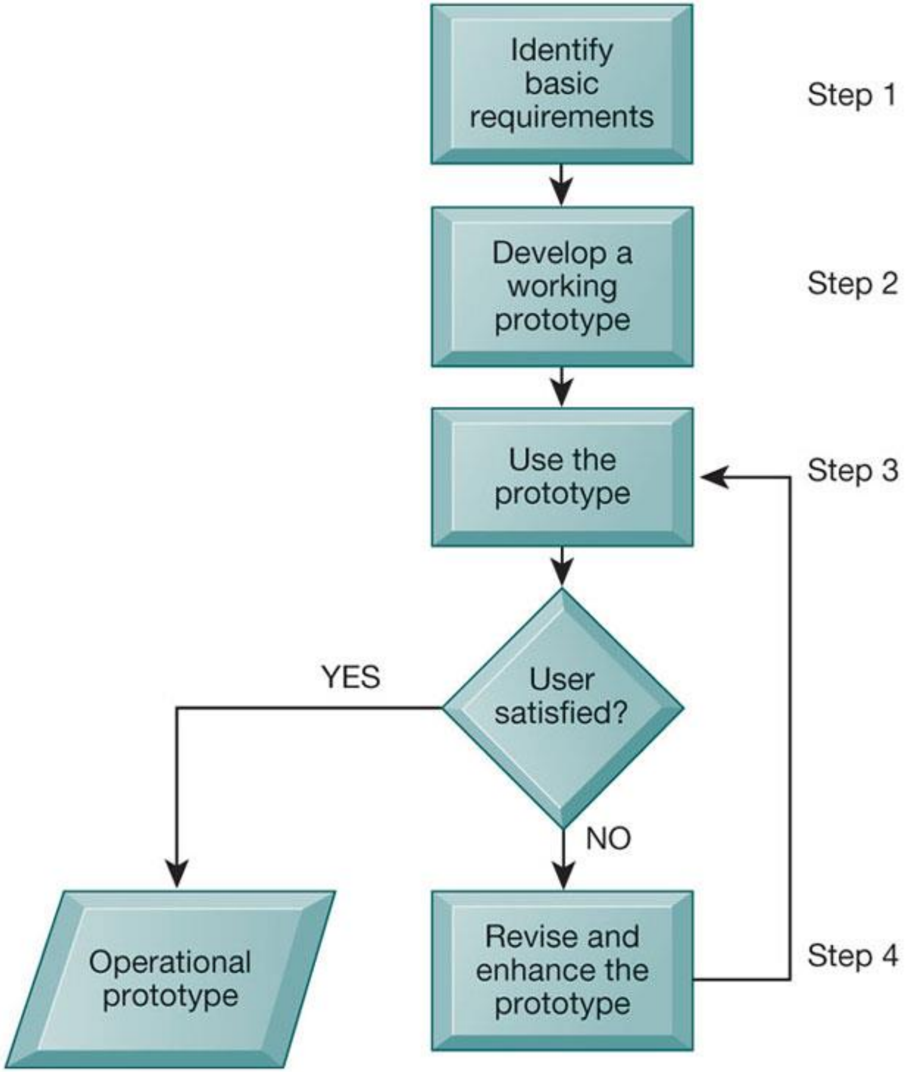
Figure 13.9 The Traditional Systems Development Life Cycle



Prototyping (1 of 2)

- Building experimental system rapidly and inexpensively for end users to evaluate
- Prototype: Working but preliminary version of information system
  - Approved prototype serves as template for final system
- Steps in prototyping
  - Identify user requirements
  - Develop initial prototype
  - Use prototype
  - Revise and enhance prototype

Figure 13.10 The Prototyping Process


Identify basic requirements
Step 1
Develop a working prototype
Step 2
Use the prototype
Step 3
YES
User satisfied?
NO
Operational prototype
Revise and enhance the prototype
Step 4

Prototyping (2 of 2)

- Advantages of prototyping
  - Useful if some uncertainty in requirements or design solutions
  - Often used for end-user interface design
  - More likely to fulfill end-user requirements
- Disadvantages
  - May gloss over essential steps
  - May not accommodate large quantities of data or large number of users
    - May not undergo full testing or documentation

End-User Development (1 of 2)

- Allows end users to develop simple information systems with little or no help from technical specialists
- Reduces time and steps required to produce finished application
- Tools include
  - User friendly query languages and reporting
  - PC software tools

End-User Development (2 of 2)

- Advantages
  - More rapid completion of projects
  - High level of user involvement and satisfaction
- Disadvantages
  - Not designed for processing-intensive applications
  - Inadequate management and control, testing, documentation
  - Loss of control over data
- Managing end-user development
  - Require cost-justification of end-user system projects
  - Establish hardware, software, and quality standards

Application Software Packages and Cloud Software Services

- Application software packages and cloud software services
  - Save time and money
  - Many packages offer customization features
- Evaluation criteria for systems analysis include:
  - Functions provided, flexibility, user friendliness, required resources, database requirements, installation and maintenance efforts, documentation, vendor quality, and cost
- Request for Proposal (RFP)
  - Detailed list of questions submitted to packaged-software vendors
  - Used to evaluate alternative software packages

Pearson

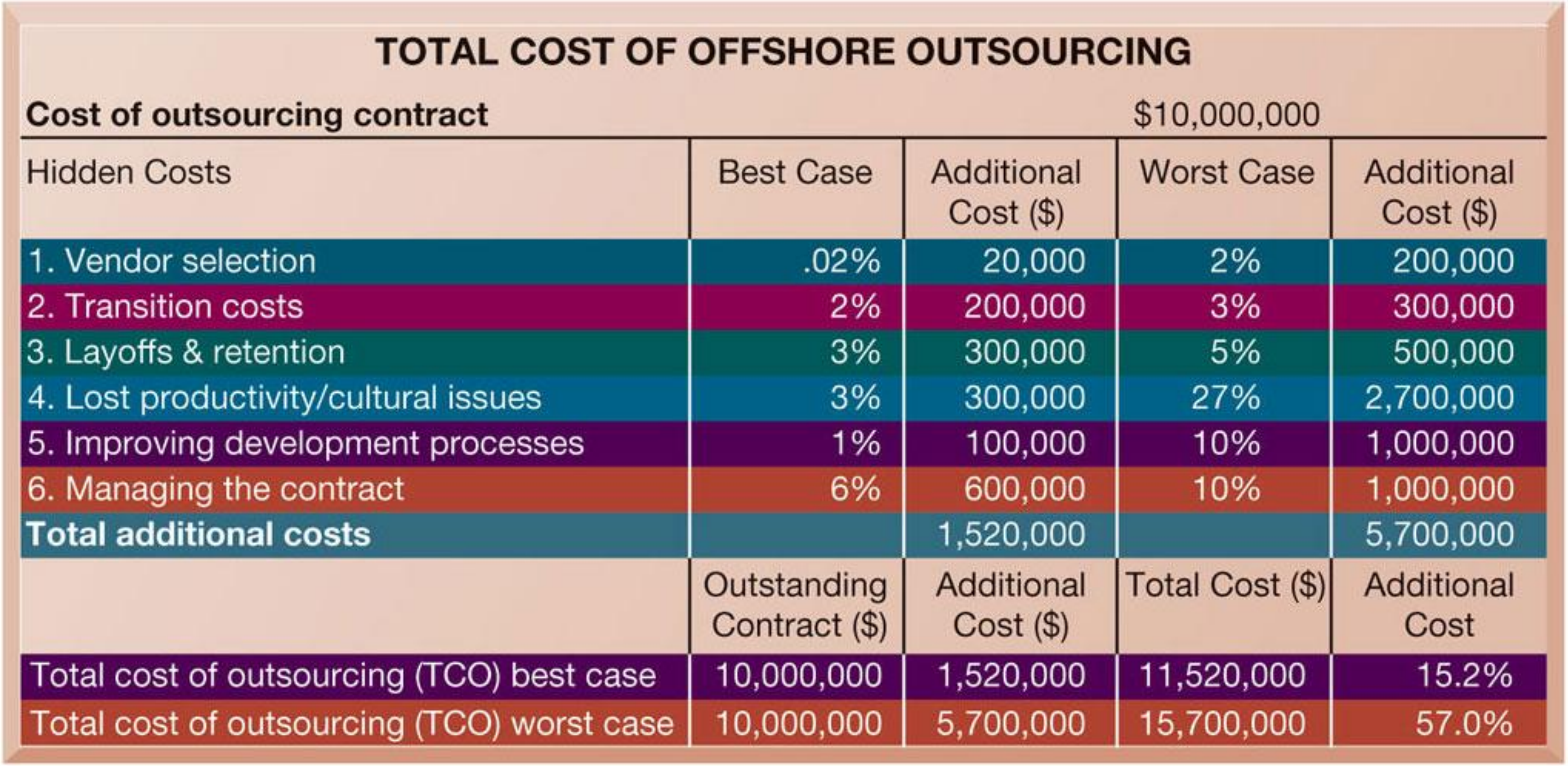
Outsourcing (1 of 2)

- Several types:
  - Cloud and SaaS providers
    - Subscribing companies use software and computer hardware provided by vendors
  - External vendors
    - Hired to design, create software
    - Domestic outsourcing
      - Driven by firm’s need for additional skills, resources, assets
    - Offshore outsourcing
      - Driven by cost-savings

Outsourcing (2 of 2)

- Advantages
  - Allows organization flexibility in IT needs
- Disadvantages
  - Hidden costs, for example:
    - Identifying and selecting vendor
    - Transitioning to vendor
  - Opening up proprietary business processes to third party

Figure 13.11 Total Cost of Offshore Outsourcing



TOTAL COST OF OFFSHORE OUTSOURCING

Cost of outsourcing contract			$10,000,000	
Hidden Costs	Best Case	Additional Cost ($)	Worst Case	Additional Cost ($)
1. Vendor selection	.02%	20,000	2%	200,000
2. Transition costs	2%	200,000	3%	300,000
3. Layoffs & retention	3%	300,000	5%	500,000
4. Lost productivity/cultural issues	3%	300,000	27%	2,700,000
5. Improving development processes	1%	100,000	10%	1,000,000
6. Managing the contract	6%	600,000	10%	1,000,000
Total additional costs		1,520,000		5,700,000
	Outstanding Contract ($)	Additional Cost ($)	Total Cost ($)	Additional Cost
Total cost of outsourcing (TCO) best case	10,000,000	1,520,000	11,520,000	15.2%
Total cost of outsourcing (TCO) worst case	10,000,000	5,700,000	15,700,000	57.0%

Rapid Application Development (RAD), Agile Development, and DevOps

- Rapid application development (RAD)
  - Process of creating workable systems in a very short period of time
- Joint application design (JAD)
  - Used to accelerate generation of information requirements and to develop initial systems design
- Agile development
  - Focuses on rapid delivery of working software by breaking large project into several small subprojects
- DevOps
  - Builds on Agile development principles as an organizational strategy

Pearson

Component-Based Development and Web Services

- Component-based development
  - Groups of objects that provide software for common functions (e.g., online ordering) and can be combined to create large-scale business applications

- Web services
  - Reusable software components that use XML and open Internet standards (platform independent)
  - Enable applications to communicate with no custom programming required to share data and services
  - Can engage other web services for more complex transactions

Mobile Application Development

- Mobile websites
- Mobile web apps
- Native apps
- Special requirements for mobile platform
  - Smaller screens, keyboards, multitouch gestures, saving resources (memory, processing)
- Responsive web design
  - Websites programmed so that layouts change automatically according to user's computing device

Copyright