Chapter 13

Building Information Systems

LEARNING OBJECTIVES

- Demonstrate how building new systems produces organizational change.
- Identify and describe the core activities in the systems development process.
- Describe the principal methodologies for modeling and designing systems.
- Compare alternative methodologies for building information systems.
- Identify and describe new approaches for system building in the digital firm era.

PC Connection Makes New System Connections

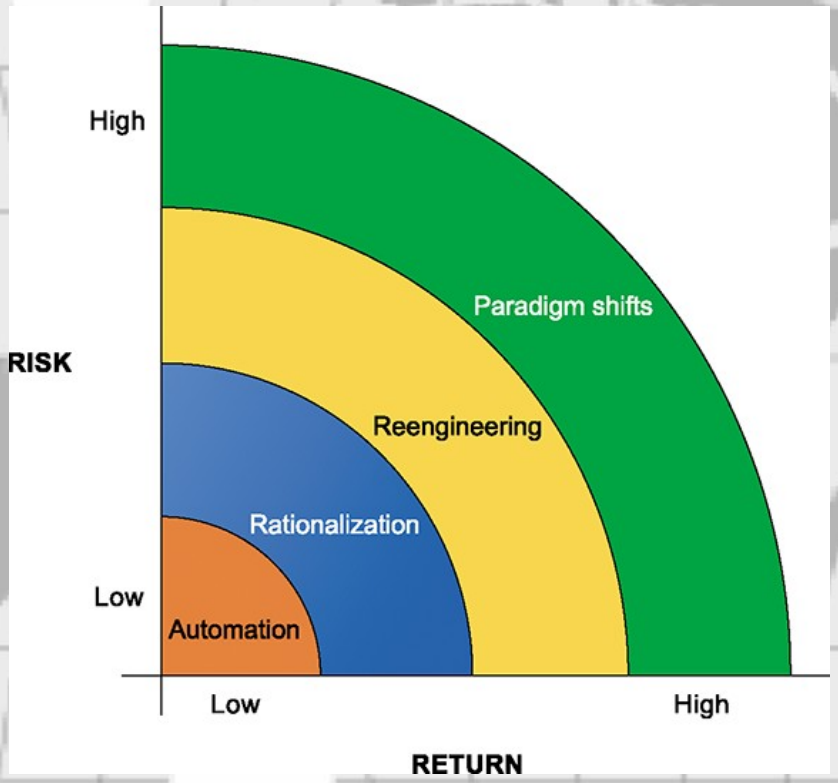
- **Problem:** Out-of-date, overly manual fulfillment system; not designed to handle multi-tiered fulfillment system
- **Solutions: Develop in-house new front-end software based on Web services,** including business process changes, workflow design, user interface changes
- Eliminated 90% of manual work for purchase orders; optimizes order fulfillment, links disparate company facilities
- Illustrates steps needed required to build new systems, from analyzing problems and requirements to managing change

Systems as Planned Organizational Change

- **Four kinds of structural organizational change enabled by IT**

1. **Automation**
   - Increase efficiency, replace manual tasks
1. **Rationalization**
   - Streamline standard operating procedures
1. **Business process reengineering (BPR)**
   - Analyze, simplify, and redesign business processes
1. **Paradigm shifts**
   - Rethink nature of business, define new business model, change nature of organization

Organizational Change Carries Risks and Rewards



The most common forms of organizational change are automation and rationalization. These relatively slow-moving and slow-changing strategies present modest returns but little risk. Faster and more comprehensive change—such as reengineering and paradigm shifts—carries high rewards but offers substantial chances of failure.

Figure 13-1

- **Business process reengineering (BPR)**
  - Large payoffs can result from redesigning business processes
  - Home mortgage industry used IT to redesign mortgage application process
    - BEFORE: 6- to 8-week process costing $3000
    - AFTER: 1-week process costing $1000
    - Replaced sequential tasks with “work cell” or team approach
  - **Work flow management:** Process of streamlining business procedures so documents can be moved easily and efficiently

Redesigning Mortgage Processing in the United States

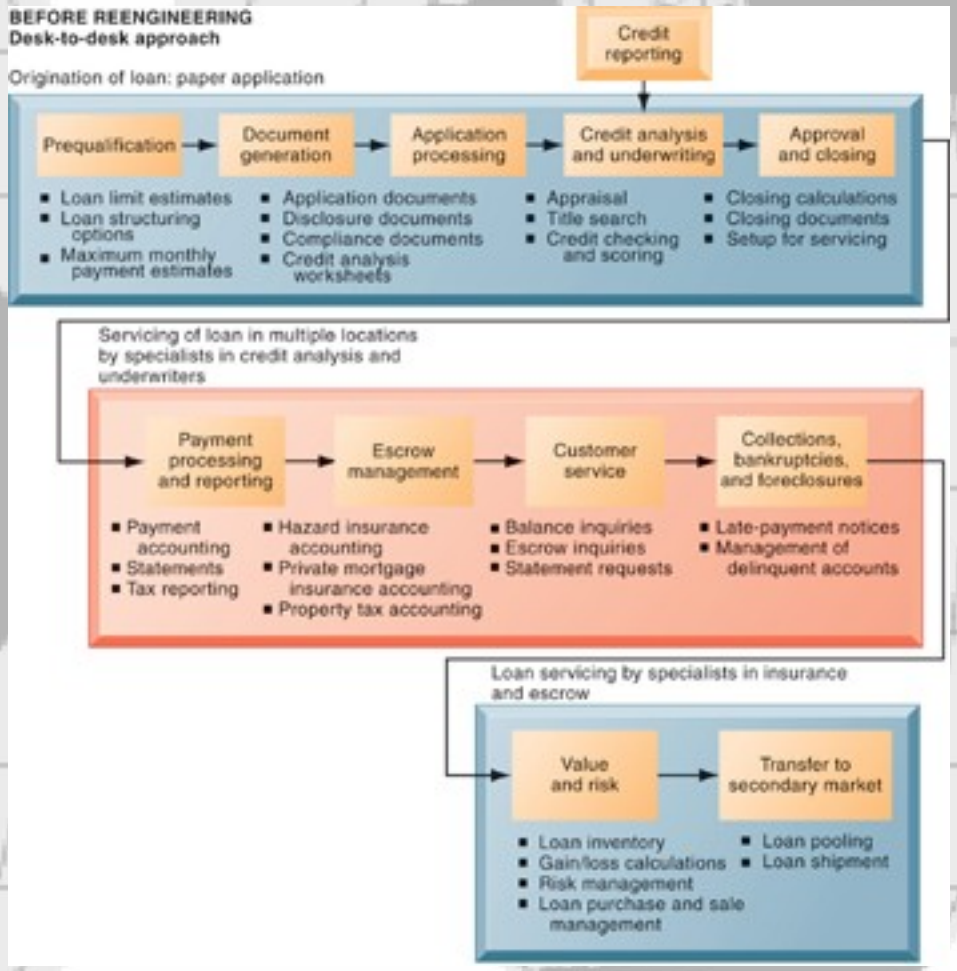


Figure 13-2A

By redesigning their mortgage processing systems and the mortgage application process, mortgage banks have been able to reduce the costs of processing the average mortgage from $3,000 to $1,000 and reduce the time of approval from six weeks to one week or less. Some banks are even preapproving mortgages and locking interest rates on the same day the customer applies.

Redesigning Mortgage Processing in the United States

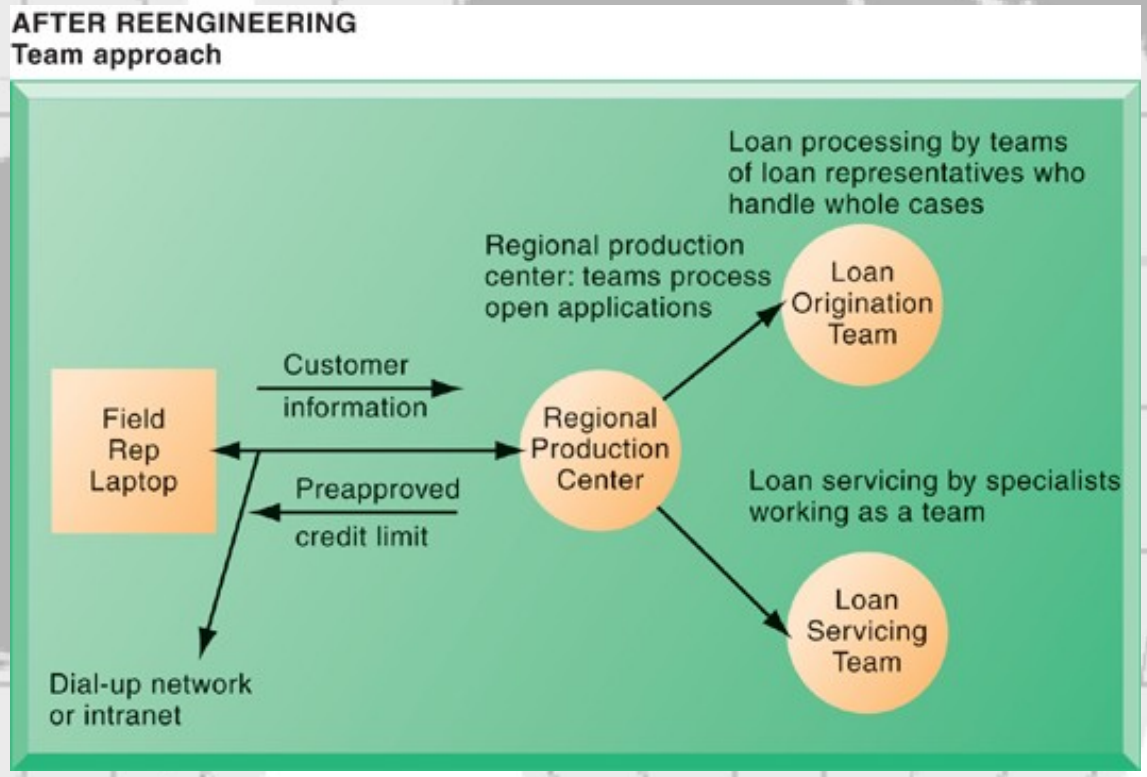


Figure 13-2B

- **Steps in effective reengineering**
  - Determine which business processes should be improved
    - Must avoid becoming good at the wrong process
  - Understand how improving the right processes will help the firm execute its business strategy
  - Understand and measure performance of existing processes as a baseline
- Even with effective BPR, majority of reengineering projects do not achieve breakthrough gains because of inadequate change management

- **Business process management (BPM)**
  - Helps firms manage incremental process changes
  - Uses process-mapping tools to:
    - Identify and document existing processes
    - Create models of improved processes that can be translated into software systems
    - Measure impact of process changes on key business performance indicators

- **Business process management (cont.)**
  - Includes:
    - Work flow management
    - Business process modeling notation
    - Quality measurement and management
    - Change management
    - Tools for standardizing business processes so they can be continually manipulated
    - Process monitoring and analytics
      - To verify process performance has improved and measure impact of process changes on key business performance indicators

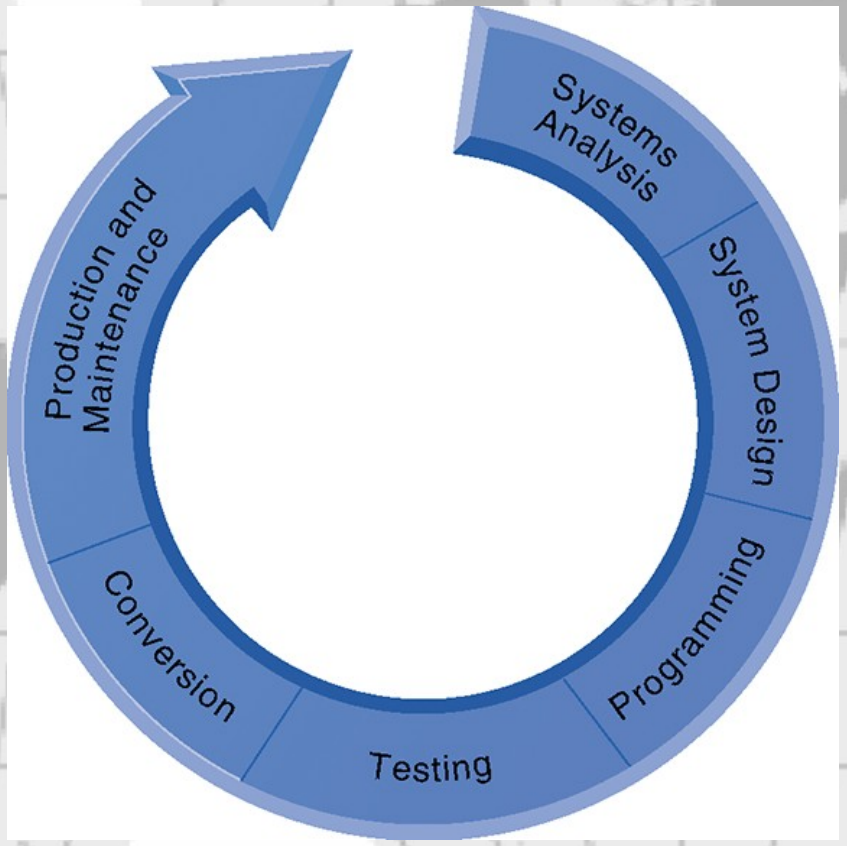
- **Quality management:**
  - Fine-tuning business processes to improve quality in their products, services, and operations
  - The earlier in the business cycle a problem is eliminated, the less it costs the company
  - Quality improvements raise level of product and service quality as well as lower costs

- **Total Quality Management (TQM):**
  - Achievement of quality control is end in itself
  - Everyone is expected to contribute to improvement of quality
  - Focuses on continuous improvements rather than dramatic bursts of change
- **Six sigma:**
  - Specific measure of quality
  - 3.4 defects per million opportunities
  - Uses statistical analysis tools to detect flaws in the execution of an existing process and make minor adjustments

- **Information systems support quality improvements by helping firms:**
  - Simplify products or processes
  - Make improvements based on customer demands
  - Reduce cycle time
  - Improve quality and precision of design and production
  - Meet **benchmarking** standards
    - **Benchmarking**: Setting strict standards for products, services, and other activities, and then measuring performance against those standards

- **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

The Systems Development Process



Building a system can be broken down into six core activities.

Figure 13-3

- **Systems analysis**
  - Analysis of problem that will be solved by system
    - Defining the problem and identifying causes
    - Specifying solutions
      - Systems proposal report identifies and examines alternative solutions
    - Identifying **information requirements**
  - Includes **feasibility study**
    - Is solution feasible from financial, technical, organizational standpoint
    - Is solution a good investment?
    - Is required technology, skill available?

- **System analysis (cont.)**
  - Establishing information requirements
    - Who needs what information, where, when, and how
    - Define objectives of new/modified system
    - Detail the functions new system must perform
  - Faulty requirements analysis is leading cause of systems failure and high systems development cost

- **Systems design**
  - Describe 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 that system reflects their business priorities and information needs
    - Insufficient user involvement in design effort is major cause of system failure

Design 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)

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

Dorfman Pacific Rolls Out a New Wireless Warehouse

- Read the Interactive Session: Organizations, and then discuss the following questions:
  - Compare Dorfman Pacific's old and new order-picking processes. Diagram the processes.
  - What role did end users play in developing Dorfman's wireless warehouse system? What would have happened to the project if users hadn't been so involved? Explain your answer.
  - What types of system building methods and tools did Dorfman use for building its wireless warehouse system?
  - How did the new system change the way Dorfman ran its business?
  - What problems did the new system solve? Was it successful?

- **Programming:**
  - System specifications from design stage are translated into software program code
  - Software may be purchased, leased, or outsourced instead
- **Testing**
  - To ensure system produces right results
  - **Unit testing:** Tests each program in system separately
  - **System testing:** Tests 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

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

When developing a test plan, it is imperative to include the various conditions to be tested, the requirements for each condition tested, and the expected results. Test plans require input from both end users and information systems specialists.

Figure 13-4

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

- **Production and maintenance**
  - System reviewed to determine if any revisions needed
  - May prepare formal **postimplementation 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% debugging, emergency work
    - 20% changes to hardware, software, data, reporting
    - 60% of work: User enhancements, improving documentation, recoding for greater processing efficiency

Summary of Systems Development Activities

CORE ACTIVITY	DESCRIPTION
Systems analysis	Identify problem(s) Specify solutions Establish information requirements
Systems design	Create design specifications
Programming	Translate design specifications into code
Testing	Unit test Systems test Acceptance test
Conversion	Plan conversion Prepare documentation Train users and technical staff
Production and maintenance	Operate the system Evaluate the system Modify the system

- **Most prominent methodologies for modeling and designing systems:**
  - Structured methodologies
  - Object-oriented development
- **Structured methodologies**
  - **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:**
  - Primary tool for representing system's component processes and flow of data between them
  - Offers logical graphic model of information flow
  - High-level and lower-level diagrams can be used to break processes down into successive layers of detail
- **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

Data Flow Diagram for Mail-In University Registration System

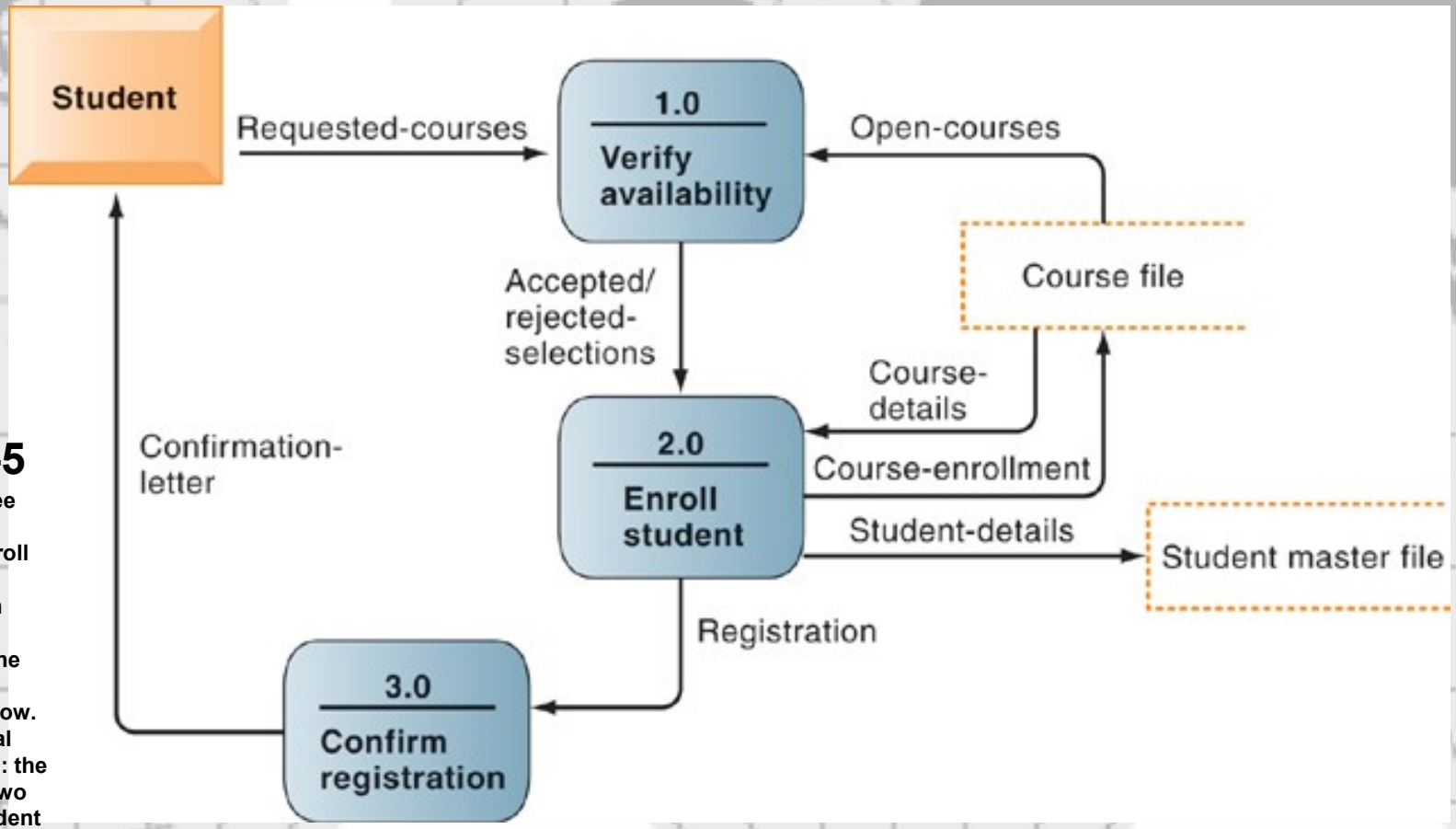
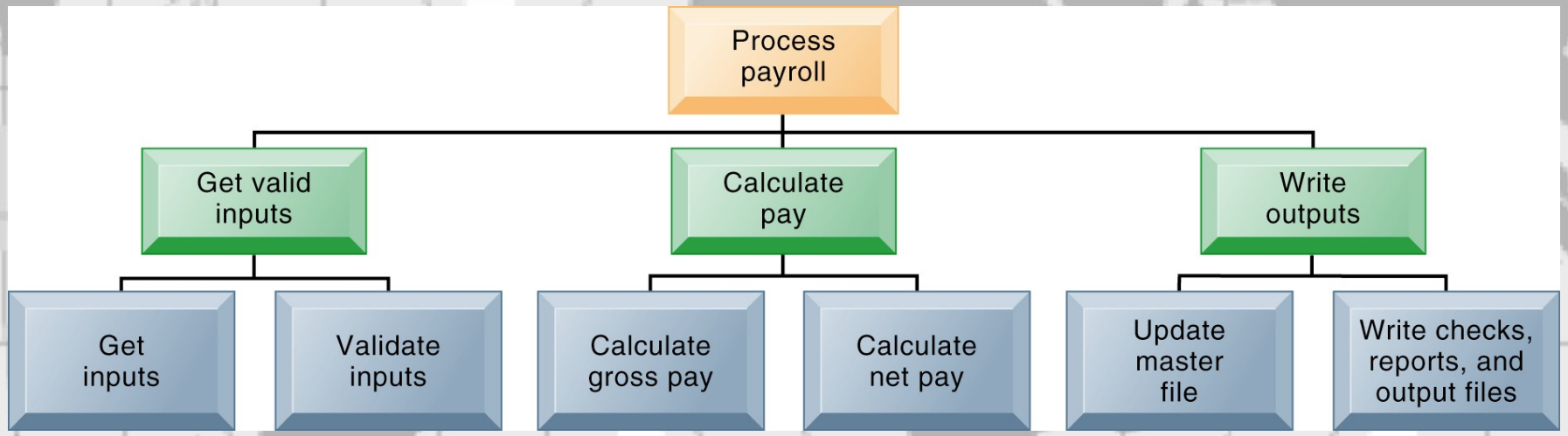


Figure 13-5

The system has three processes: Verify availability (1.0), Enroll student (2.0), and Confirm registration (3.0). The name and content of each of the data flows appear adjacent to each arrow. There is one external entity in this system: the student. There are two data stores: the student master file and the course file.

High-Level Structure Chart for a Payroll System

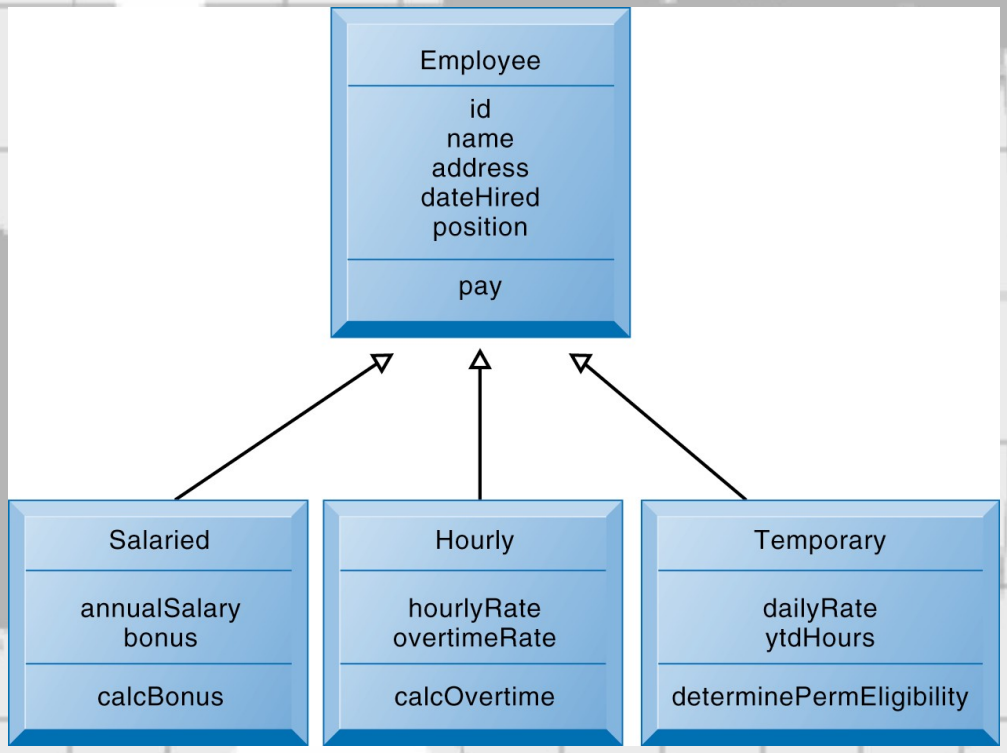


This structure chart shows the highest or most abstract level of design for a payroll system, providing an overview of the entire system.

Figure 13-6

- **Object-oriented development**
  - Uses **object** as basic unit of systems analysis and design
    - **Object:**
      - Combines data and the specific processes that operate on those data
      - Data encapsulated in object can be accessed and modified only by operations, or methods, 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

Class and Inheritance



This figure illustrates how classes inherit the common features of their superclass.

Figure 13-7

- **Object-oriented development**
  - 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
  - Because objects reusable, object-oriented development can potentially reduce time and cost of development

- **Computer-aided software engineering (CASE)**
  - 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

- Alternative Systems-Building Methods
  - Traditional systems life-cycle
  - Prototyping
  - End-user development
  - Application software packages
  - Outsourcing

- **Traditional systems lifecycle:**
  - Oldest method for building information systems
  - Phased approach - divides development into formal stages
    - Follows “waterfall” approach: Tasks in one stage finish before another stage begins
  - Maintains 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

- **Prototyping**
  - 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**
    1. Identify user requirements
    2. Develop initial prototype
    3. Use prototype
    4. Revise and enhance prototype

The Prototyping Process

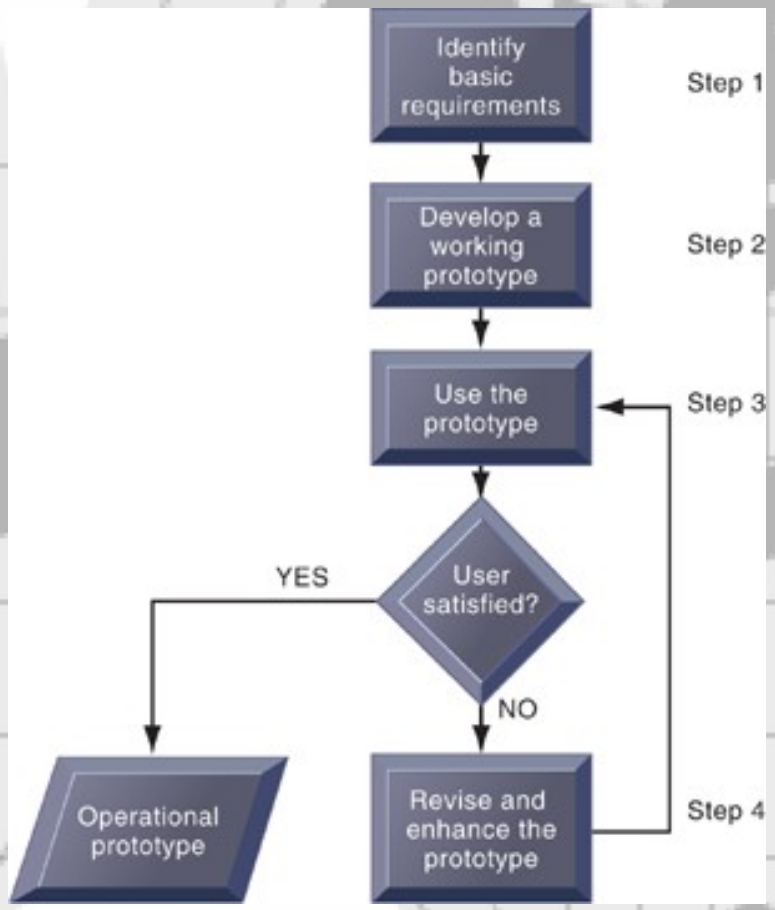


Figure 13-8

The process of developing a prototype can be broken down into four steps. Because a prototype can be developed quickly and inexpensively, systems builders can go through several iterations, repeating steps 3 and 4, to refine and enhance the prototype before arriving at the final operational one.

- **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:**
  - Uses **fourth-generation languages** to allow end-users to develop systems with little or no help from technical specialists
  - **Fourth generation languages**: Less procedural than conventional programming languages
    - PC software tools
    - Query languages
    - Report generators
    - Graphics languages
    - Application generators
    - Application software packages
    - Very high-level programming languages

- **End-user development (cont.):**
  - **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**
  - Save time and money
  - Many packages offer customization features:
    - Allow software package to be modified to meet unique requirements without destroying integrity of package software
  - Evaluation criteria for systems analysis include:
    - Functions provided by the package, flexibility, user friendliness, hardware and software 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

- **Outsourcing**
  - **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 firms need for additional skills, resources, assets
      - Offshore outsourcing
        - Driven by cost-savings

- **Outsourcing (cont.)**
  - **Advantages**
    - Allows organization flexibility in IT needs
  - **Disadvantages**
    - Hidden costs, e.g.
      - Identifying and selecting vendor
      - Transitioning to vendor
    - Opening up proprietary business processes to third party

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 ($)
1. Vendor selection	0%	20,000	2%	200,000
2. Transition costs	2%	200,000	3%	300,000
3. Layoffs & retention	3%	300,000	5%	500,000
4. Lost productivity/cultural issues	3%	300,000	27%	2,700,000
5. Improving development processes	1%	100,000	10%	1,000,000
6. Managing the contract	6%	600,000	10%	1,000,000
Total additional costs		1,520,000		5,700,000
	Outstanding Contract ($)	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%

If a firm spends $10 million on offshore outsourcing contracts, that company will actually spend 15.2 percent in extra costs even under the best-case scenario. In the worst-case scenario, where there is a dramatic drop in productivity along with exceptionally high transition and layoff costs, a firm can expect to pay up to 57 percent in extra costs on top of the $10 million outlay for an offshore contract.

Figure 13-9

Did Chrysler Make the Right Outsourcing Decision?

- Read the Interactive Session: Management, and then discuss the following questions:
  - What management, organization, and technology issues should Chrysler have explored when deciding whether to outsource to TCS?
  - What points should Chrysler have addressed in its outsourcing contract with TCS?
  - Was Tata Consultancy Services a good outsourcing choice for Chrysler? Why or why not? Explain your answer.

- **Rapid application development (RAD)**
  - Process of creating workable systems in a very short period of time
  - Utilizes techniques such as:
    - Visual programming and other tools for building graphical user interfaces
    - Iterative prototyping of key system elements
    - Automation of program code generation
    - Close teamwork among end users and information systems specialists

- **Joint application design (JAD)**
  - Used to accelerate generation of information requirements and to develop initial systems design
  - Brings end users and information systems specialists together in interactive session to discuss system's design
  - Can significantly speed up design phase and involve users at intense level

- **Agile development**
  - Focuses on rapid delivery of working software by breaking large project into several small sub-projects
  - Subprojects
    - Treated as separate, complete projects
    - Completed in short periods of time using iteration and continuous feedback
  - Emphasizes face-to-face communication over written documents, allowing collaboration and faster decision making

- **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
    - Using platform and device-independent standards can result in significant cost-savings and opportunities for collaboration with other companies