Chapter 13

CIMB Group Redesigns Its Account Opening Process

- ✧ **Problem:**Financial services provider CIMB group wanted to improve efficiency in business processes, specifically process of opening accounts at branch
- ✧ **Solutions**: **ARIS BPM tool** used to identify 25 areas for improving efficiency. Utilized Malaysia’s government ID smart card to automate inputting customer data, reducing time spent by 50%
- ✧ Demonstrates the use of information systems to streamline and redesign business processes
- ✧ Illustrates first key step in building new system – analysis

Systems as Planned Organizational Change

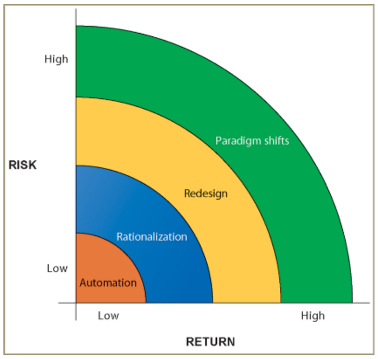
✧ Structural organizational changes enabled by IT

1. **Automation**
   - ▪ Increases efficiency
   - ▪ Replaces manual tasks
2. **Rationalization of procedures**
   - ▪ Streamlines standard operating procedures
   - ▪ Often found in programs for making continuous quality improvements
     - Total quality management (TQM)
     - Six sigma
3. **Business process redesign**
   - ▪ Analyze, simplify, and redesign business processes
   - ▪ Reorganize workflow, combine steps, eliminate repetition
4. **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 redesign and paradigm shifts—carries high rewards but offers substantial chances of failure.

FIGURE 13-1



- **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

AS-IS BUSINESS PROCESS FOR PURCHASING A BOOK FROM A PHYSICAL BOOKSTORE

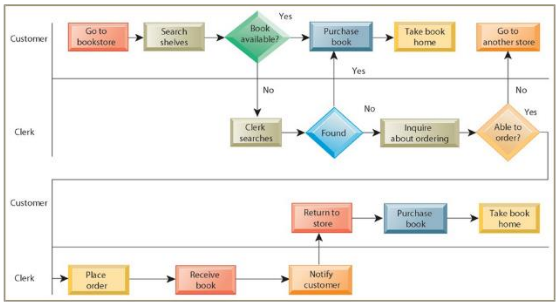


FIGURE 13-2 Purchasing a book from a physical bookstore requires many steps to be performed by both the seller and the customer.

REDESIGNED PROCESS FOR PURCHASING A BOOK ONLINE

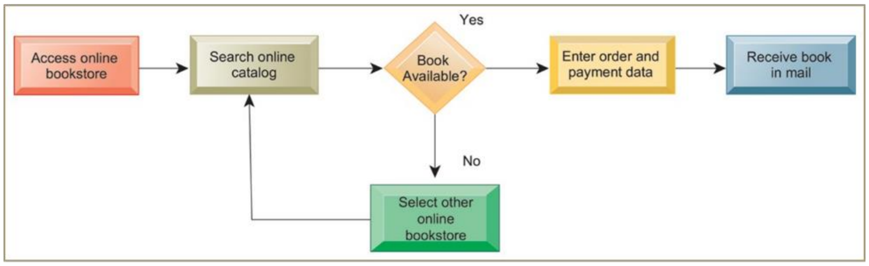


FIGURE 13-3 Using Internet technology makes it possible to redesign the process for purchasing a book so that it requires fewer steps and consumes fewer resources.

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

CAN BUSINESS PROCESS MANAGEMENT MAKE A DIFFERENCE?

Read the Interactive Session and discuss the following questions

- Why are large companies such as AmerisourceBergen and Diebold good candidates for business process management?
- What were the business benefits for each company from redesigning and managing their business processes?
- How did BPM change the way these companies ran their businesses?
- What might be some of the problems with extending BPM software across a large number of business processes?
- What companies stand to gain the most by implementing BPM?

Overview of Systems Development

- **Systems development:**

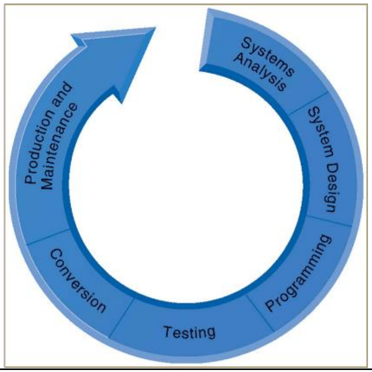
  – Activities that go into producing an information system solution to an organizational problem or opportunity

  1. Systems analysis
  2. Systems design
  3. Programming
  4. Testing
  5. Conversion
  6. Production and maintenance

THE SYSTEMS DEVELOPMENT PROCESS

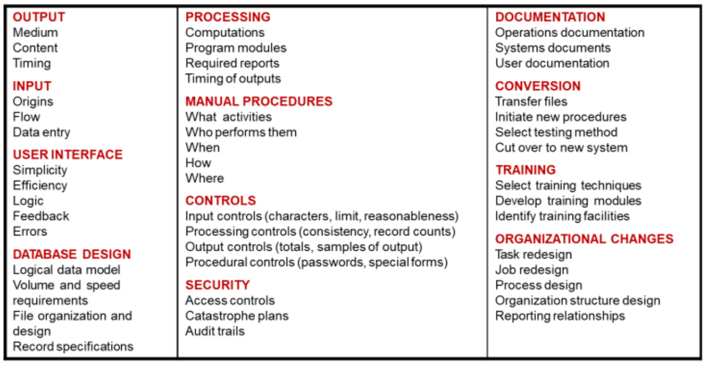
Building a system can be broken down into six core activities.

FIGURE 13-4



- **Systems analysis**
  - **Analysis of problem to be solved by new 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 and good investment?
    - Is required technology, skill available?
  - **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**
  - **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



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

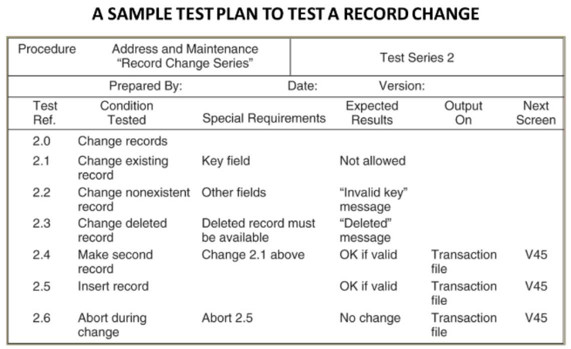
DOCUMENTATION
Operations documentation
Systems documents
User documentation

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

- **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

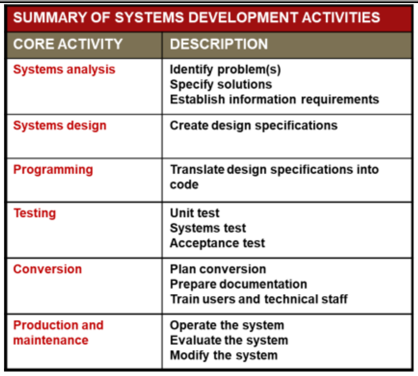


A SAMPLE TEST PLAN TO TEST A RECORD CHANGE

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

FIGURE 13-5 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.

- **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 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% 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:**
    1. **Structured methodologies**
    2. **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

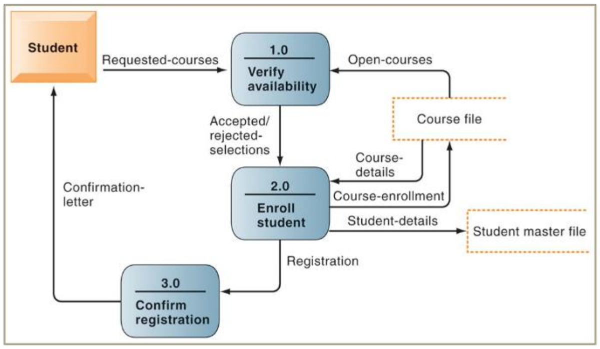


FIGURE 13-6 The system has three processes: Verify availability (1.0), Enroll student (2.0), and Confirm registration (3.0). The name and content of each of the data flows appear adjacent to each arrow. There is one external entity in this system: the student. There are two data stores: the student master file and the course file.

HIGH-LEVEL STRUCTURE CHART FOR A PAYROLL SYSTEM

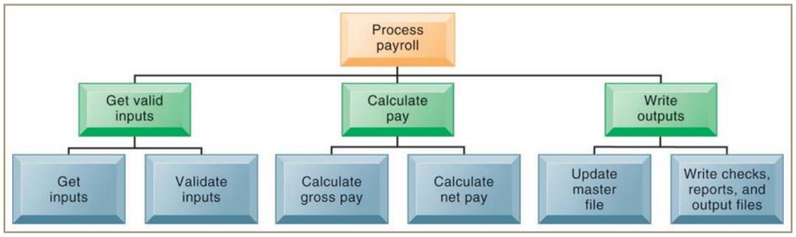
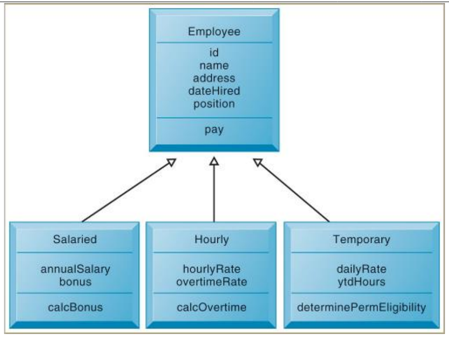


FIGURE 13-7 This structure chart shows the highest or most abstract level of design for a payroll system, providing an overview of the entire system.

- **Object-oriented development**
    - **Object is basic unit of systems analysis and design**
        - **Object:**
            - Combines data and the 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 13-8

- **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 are 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 Approaches

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

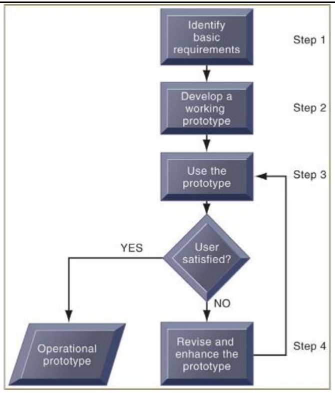
- Traditional systems lifecycle:
  - **Oldest method for building information systems**
  - **Phased approach divides development into formal stages**
    - Follows “waterfall” approach: Tasks in one stage finish before another stage begins
  - **Maintains formal division of labor between end users and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

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.

FIGURE 13-9


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

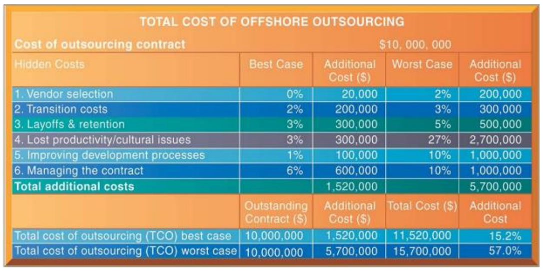
- **Advantages of prototyping**
  - **Useful if some uncertainty in requirements or design solutions**
  - **Often used for end-user interface design**
  - **More likely to fulfill end-user requirements**
- **Disadvantages**
  - **May gloss over essential steps**
  - **May not accommodate large quantities 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
  - **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 offer customization features:**
    - Software can 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
  - **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%

FIGURE 13-10 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.

Application Development for the Digital Firm

- **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