IMS1001/ CSE1204
Information Systems

Week 5:
Systems Development and the
Systems Development Life
Cycle (SDLC)

Lecture Objectives

• to understand the information
  systems problem-solving process
• to be aware of the main phases of the
  systems development lifecycle

Business Information Processing Problems

• Information problems occur when an organisation’s
  current systems fail to meet its information and
  processing needs adequately.
  Some causes:
  • changing information needs
  • business expansion
  • cost pressures
  • competitive pressures
  • new business activities
  • inefficiencies
• Information processing problems can occur at any
  stage of the information processing cycle
Characteristics
- complex - the number and variety of components and their interactions
- non-standard - many organisations have similar needs, but rarely are these identical solutions must be customised for specific circumstances
- unstructured - problems cannot easily be broken down into clearly-defined components with easily-identified connections between them

How do you solve problems?
The ‘Intuitive’ Approach versus The ‘Scientific’ Approach
(not quite true, but we’ll carry on for the moment)

‘Intuitive’ Problem-solving
- “I don’t have to think about how to solve the problem; I just do it”
- No conscious reasoning process or planning involved
- Use accumulated knowledge/understanding/judgement/heuristics (i.e. ‘rules of thumb’), but do not make them explicit
- The most commonly used approach for everyday problem-solving
- Agile Methods?
'Scientific' Problem-solving

- Identify that there is a problem
- Describe it
- Analyse the problem and specify what needs to be done to fix it
- Identify the potential courses of action which may fix the problem - evaluate them and choose the best one
- Describe in detail the chosen course of action and put the chosen course of action into effect
- Evaluate the outcome of the chosen course of action and check that it has fixed the problem

Which Approach to Use?

- Depends on:
  - nature of the problem
  - complexity of the problem
  - degree of standardisation of the problem
  - experience in solving this kind of problem
  - extent to which the problem can be structured into small self-contained parts

Approaches to information systems development

- early computer information systems development focused on technology, programming and technical skills
- systems developers were technically trained and skilled, and used rule-of-thumb and personal experience as the basis for developing systems
- as computer use became more widespread, a backlog of computer application requests developed, existing applications increasingly required changes, and changes made tended to have unexpected and undesirable effects
- these problems led to awareness of the need for an overall accepted, standardised approach to system development
The Process of System Development

- There is no 'universal' problem-solving process which can meet the needs of all system development situations
- Approaches to developing information systems to solve business information processing problems must be tailored to meet the needs of the situation
- Some elements of the system development process can be 'standardised' to some degree

The Systems Development Lifecycle (SDLC)

- The concept of the systems development life cycle (SDLC) is an attempt at achieving this standardisation.
- It provides:
  - a systematic and orderly approach to solving business information and processing problems
  - a means of managing, directing, monitoring and controlling the process of system building, including:
    - a description of the process - steps to be followed
    - deliverables - reports/programs/documentation/etc
    - milestones - dates of completion of steps or deliverables

The Systems Development Lifecycle

- It has several phases that define the progress of the development process
- It is often adapted to suit the organisational, human and technical needs of organisations and system development projects
- There are many variants of the SDLC: traditional ‘waterfall’ or linear model, iterative model, spiral model etc.
- We will consider the traditional waterfall model first
Benefits of SDLC

- breaks the problem-solving process into manageable steps
- identifies and defines everything which needs to be done, and how it should be done
- identifies the resources needed in each step
- identifies who will do each activity and when they will do it
- provides a basis for project planning

Principles of System Development

- get the owners and users involved
- use a problem-solving approach
- establish phases and activities
- establish standards for consistent development and documentation
- justify systems as capital investments
- don’t be afraid to cancel or revise project scope
- divide and conquer
- design systems for growth and change

Systems Development Phases
Initiation (Why?)

- Is this project worth doing?

  - Planned development project
  - Survey Project Feasibility
  - Unplanned development project

  - Feasibility Report (scope defined)
  - Constraints
  - Problem/opportunity details

  ANALYSIS

Initiation

- a preliminary investigation of the problems, opportunities, constraints and available resources in order to decide on a course of action
  - enhance existing system?
  - develop a new information system?
  - do nothing – add it to the backlog?

- define the system scope: the functions/activities which are to be developed/redeveloped: poor scope management often results in unsuccessful systems

Initiation

- Defining the project scope includes identifying:
  - key stakeholder groups
  - perceived problems and opportunities
  - constraints
  - possible solutions & client expectations

- Key deliverable is a feasibility report:
  - Includes overview of proposed solutions with cost/benefit analyses for each solution
Analysis (What?)

- Define the client’s requirements
- Analyse the problem and define requirements

Analysis

- “Don’t try to fix it unless you understand it”
- Study the existing system to thoroughly understand the problems and opportunities
- Review findings with clients and revise scope if necessary
- Clearly define WHAT the new system must do
- Agree on acceptance criteria for the new system (sign-off on the system specification)
  - should the system specification be “frozen”?  
- Assess feasibility again

Design (How?)

- Define how the system will be implemented
- Select a design strategy and specify details
Design

• Generate a number of design options based on technical, operational, economic, scheduling and tendering constraints (HOW)
• The client selects the best option for their needs (assess feasibility again)
• Acquire the necessary hardware and software
• Design interfaces, databases, networks as required
• Specify integration requirements and software requirements (programs)

Implementation (Build)

• Build and deliver the system

  Build, test, install and deliver the new system

  User acceptance testing
  User Documentation
  User Training
  Production System
  Project Report

  System Owners
  System Users

  Technical Design Report
  Hardware/Software
  System and Technical Documentation

  MAINTENANCE

Implementation

• Build/modify databases and networks as required
• Build and test programs
• Prepare users for new system
  • acceptance testing, user documentation, user training, maintenance procedures
• Finalise system and technical documentation
• Install the system
Review

• What went wrong/right? Why?

- Review the system and the project
- System
- Audit Report
- Problems/New ideas
- System
- Users
- Project issues and system bugs
- Steering Committee
- Monash University, SIMS, Semester One, 2005

Review

• How well were the system and project objectives met?
  - Client’s requirements met, within budget, on time?
  - Can further benefits be realised?
  - Are major changes required?
  - How successful was the development process .. what can we learn?
  - Review the maintenance effort

Maintenance

• Fix it / Make it better

- Maintain the new system
- System Users
- Problems/New ideas
- Additional training and documentation
- Modifications
- Technical problems and new technology
- Escalating maintenance
- Fix and enhancements
- Project staff
- Monash University, SIMS, Semester One, 2005
Maintenance

- Corrective - fix errors
- Adaptive - satisfy changing needs
- Perfective - enhance performance
- Preventative - fix potential problems

- If the cost of maintenance is too high consider other options:
  - new development, purchase a software package, re-engineer/modify

Systems Development:
Cross Life Cycle Activities

Cross Life Cycle Activities are those which overlap many or all of the life cycle phases. Some of these are:
- Quality - must be embedded in the process of systems development to achieve a quality outcome
- Project Management - to monitor and control the project and ensure it stays on track
- Documentation - essential at every stage to help ensure project and system viability
- Ethics - voluntary compliance with guidelines of IS/IT professional societies

Quality

- Quality is defined as fitness for purpose and concerns both process and product.
- Error detection and correction in analysis and design is much cheaper than after the system is implemented.
- Achieving quality requires that organisational structures, responsibilities, procedures, processes and resources for implementing quality management are in place.
Project management

- Select systems development methodology
- Plan the project tasks
- Estimate the resources and time required to complete individual phases of the project
- Staff the project team
- Organise and schedule the project effort (tasks/time/people/technical resources) and therefore cost
- Control the project development: direct the team, monitor progress, replan, restaff, reallocate resources

Documentation

- Various types of documentation must be produced throughout the SDLC
- The *data dictionary* plays an important role during and after systems development:
  - A repository for information about and definitions of all "objects" identified during development
  - It supports and is maintained throughout the system lifecycle
  - It provides an important source for system documentation

Professional ethics

- Australian Computer Society (ACS) Code of Ethics for IT professionals
- Your reputation
- Your client's interests
- Confidentiality
  - The client's own and their competitors'
- Impartiality
- Honesty
Ethical issues in IT

- Designing a controversial computer system
- Software piracy
- Morality v. ethics

Systems Development: The systems developer’s skills

Systems developers require many different skills during the SDLC. Some of these are:
- Interpersonal skills - to communicate effectively, facilitate groups, work in teams, manage expectations and change, deal with organisational politics
- Analytical skills - to identify problems and determine solutions
- Business knowledge - understanding of business systems
- Technical skills and knowledge - to use the technology, and understand its potential and limitations
- Management skills - to manage resources, projects, risk, and organisational change

Some Approaches to Systems Development

There are many different approaches to developing systems depending on the nature of the systems and the users’ needs. Some of these are:
- Traditional Waterfall SDLC - formal approach which partitions development into distinct phases
- Prototyping - an iterative process of building an experimental system rapidly
- Application Packages - purchase commercially available software
### Some Approaches to Systems Development

- **Joint Application Development (JAD)** - a workshop approach in which a facilitator, users, managers, and developers work intensively together over a short period (days) to specify requirements and design a system.
- **Participatory Design (PD)** - where the central focus is the users participating actively in system development.
- **RAD** - rapid application development using techniques to build systems quickly where appropriate.

### Who does Systems Development?

- The organisation’s information technology department (in-house development).
- End-user computing - development of systems by end-users with minimal assistance.
- Outsourcing - contracting development to external providers.
- IT consultants.
- Often a combination of the above.

### Summary

There is no such thing as a 'correct', 'standard' development lifecycle approach to systems development, because all information and processing problems are different.
### References

   5th ed., *Systems Analysis and Design Methods*,  
   Irwin/ McGraw- Hill, New York, NY. Chapter 3
   *Modern Systems Analysis and Design*,  
   Benjamin/ Cummings, Massachusetts. Chapter 1