Seminar 4:
The Systems Development Life Cycle
System Development Alternatives

Seminar Objectives

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

Why is it important for you?
- Overview of what you are going to be involved in for the next 2.5 years of your BIS degree
- Provides you with a foundation for methodically developing an Information Systems

IMS1501: Unit framework

Business Information Problems and Opportunities

- The need to build new information systems or change existing ones comes about because either:
  - there are problems in the way in which existing systems operate; or
  - changes in circumstances create opportunities to improve things by doing them differently

Business Information Problems

- Information problems occur when the organisation’s systems fail to meet its information needs adequately. Some causes:
  - changing information needs
  - business expansion
  - cost pressure
  - competitive pressure
  - staff dependency
- Information problems can occur at any stage of the information processing cycle

How do you solve problems?

The ‘Intuitive’ Approach
versus
The ‘Scientific’ Approach
Historical Approach

- Historically, people building information systems have tended to use an intuitive approach
- This is partly a reflection of the origins of computing and partly a reflection of the apparent conceptual simplicity of IS development. It looks easy, so who needs to follow a structured, 'scientific' approach? Just Do IT!!
- The need for structured approaches grew out of the disasters which followed from people 'just doing it' without following any organised sequence of activities

The Process of System Development

- There is no such thing as a 'universal' problem-solving process which meets all situations; our approach to problem-solving must be tailored to meet the needs of the situation
- However, some elements of the problem-solving process can be 'standardised' to some extent

The Systems Development Process.1

The systems development life cycle (SDLC) is an attempt at this standardisation. It provides:
- a systematic and orderly approach to solving business problems - automated option - CASE tools
- 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 Process.2

- It features several phases to mark the progress of the development process (many variations ..)
- It is often adapted to suit the business, social and political needs of organisations and projects
- We will use the generic waterfall model as a springboard for exploring system development issues

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

The Systems Development Life Cycle (SDLC)
**Initiation (Why?)**

- Is this project worth doing?

- Planned development project
- Unplanned development project

**Analysis (What?)**

- Define the clients requirements (What?)

**Design (How?)**

- Define how the system will be implemented

**Initiation**

- A preliminary investigation of the problems, opportunities, constraints and available resources to decide on a course of action
  - Enhance existing system
  - Develop a new information system
  - Do nothing .. add it to the backlog
- Define the scope .. poor scope management often results in unsuccessful systems
  - Scope = identifying .. key client groups, perceived problems and opportunities, constraints, possible solutions & client expectations

**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 (signed systems specification) – should the system spec. be frozen?
- Assess feasibility again

**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
  - User acceptance testing, user documentation, user training, maintenance procedures
  - Finalise system and technical documentation
  - Install the system

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?
  - How well were the system objectives met?
    - Clients 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

Review

- How well were the system objectives met?
  - Clients 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
  - 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 package, re-engineer
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 professional societies

Alternative Approaches to IS Development

There are many different approaches to developing systems depending on the nature of the systems and the users needs. YOUR JOB:
To investigate the usefulness and limitation of these alternatives WILL BE AN EXAM QUESTION

Alternative Approaches to IS Development

There are many different approaches to developing systems depending on the nature of the systems and the users needs. Some of these are:
- SDLC (System Development Life Cycle)
  - A detailed, well-planned development process
  - Provides a comprehensive formal framework, but time consuming
  - Works best on projects in which users have a clear idea about their requirements
- SSADM (Structured Systems Analysis & Design Methodology)
  - Originated from the UK (1981), mostly used by government agencies/department & its contractors
  - Use of multiple methods (which are cross-referenced & sometimes overlaps) to provide a thorough, comprehensive, and complete view of the system
  - Potentially tedious + long development lifecycle + expensive

Alternative Approaches to IS Development

Protootyping (evolutionary development)
- Developers produce a prototype based on general indication of the user’s needs
- The prototype contains portions of the system of most interest to the users or small scale working model of the system
- Continual process of review-refine-extend until the final specifications specified by the user are reached

Application Packages
- Purchase commercially available software

Rapid Application Development (RAD)
- Cursory attempt at conceptual data modeling
- Define database during development of initial prototype
- Uses GUI development environment (e.g., ability to 'drag & drop' S/W components)
- Uses reusable components
- Uses code-generator
- Iterative process of implementation and maintenance with each new version of the module

X-Programming (XP)
- Based on values of simplicity, communication, and feedback
- Developers produce the S/W in a series of small, fully integrated releases
- Normally associated with RAD

Joint Application Development (JAD)
- Group based method for collecting user requirements & creating system design
- Analysts, a facilitator, and all the users meet simultaneously to define the requirements & reach agreement of the needs & requirements for the system
- Might use a combination of different IS development methods
- Commonly conducted within the Integrated Environment (IE) scope/concept

Object Oriented (OO) Development
- The proponents of OO dev claims that OO dev provides: faster development, greater reuse potential, increased quality, easier maintenance, enhanced modifiability
- May be expensive in the earlier/starter years during the creation of the OO libraries
- Training of the OO concept is needed
Summary

There is no such thing as a 'correct' 'standard' development life-cycle, because all information problems are different and need different system development approaches.

References