Lecture 4 - STRADIS, JSD, SSADM, SSM

Project Failure Types

- Overrun - cost or time targets not met
- Scrapping - project not completed
- Neglect - project completed but doesn't meet needs, doesn't deliver ROI

STRADIS (Structured Analysis Design and Implementation of IS)

Philosophy: Top down functional decomposition. Fundamentally about studying a predefined problem, and modelling it in such a way that a system can be created.
Systems analysis > Systems design

Solving a predefined problem without needing to further define it
Relies on DFDs

Used when there is a backlog of systems waiting to be developed or insufficient resources to develop all potential systems

1. Initial study
    - Justify the new system: Cost and benefits of each proposal
    - Systems viewed as contributing toward increasing revenues, reducing costs and improving services
    - Construct high level DFD of existing system and interfaces
    - Estimate costs for investigation and new system
    - Report for management

2. Detailed study
    - If initial study is approved
    - Examine existing system in detail
    - Identify users
    - Create logical model of new system
    - Different levels of DFDs
    - Refine estimates with effects and costs
    - Present to management
    - Mark system boundary with a dashed line on DFD

3. Defining and designing alternative solutions
    - Organisational objectives → System objectives
    - New DFD of current system meeting system objectives

- Identify alternatives, statements covering
    - Parts of DFD to implement
    - UI
    - Estimated cost and benefits
    - Risks

Report with DFD of current and new system, limitations of current system

4. Physical design
Refine best alternative into a specific physical design

- Data dictionary
- Files/Database designed based on data stores in DFD, converted and loaded
- Firm estimate of cost of developing and operating new system is prepared
- Implementation plan prepared
- Testing
- Commit system for live operation
- Compare performance to original objectives

JSD (Jackson System Development)
Its a later version of STRADIS. *Logical methodology* which aims to simplify system development by breaking down the problem into three independent stages/domains. As such, it aims to simplify software development by simply providing a framework through which development occurs, from the ground up all the way to a working system.

Developing software systems is the main goal
Process scheduling and real-world modelling

1. Modeling Phase - not just system, but the entire problem domain modelled
Events and entities are identified
Entity structures, behaviours and life cycles formed
Entity structure diagram created

2. Network Phase
Highly distributed network of programs created
Inputs and outputs added to model
i. Initial Model Step
Model created by developer
Contains sequential process for each instance of an entity type
System specification diagram - shows system as a network of processes, communication with processes

ii. Function Step
Functions added to model for required outputs

iii. System Timing Step
Consider allowable delays between receipt of inputs and production of outputs
Time constraints from user requirements or technical considerations

3. Implementation Phase
Details of design and coding
Network model of solution is converted into a physical system
SID (System Implementation Diagram)

SSADM (Structured Systems Analysis and Design Method)
Systematic approach to design and implementation of large-scale IS projects
Each stage has pans, timescales, controls and monitoring procedures, links to next module
Each activity has deliverables to facilitate management of project
Doesn't cover programming or testing
Logical data modeling - identifying, modeling and documenting data requirements of system being design, separate data into entities and relationships
Data flow modeling - aka DFD
Entity behaviour modeling - identifying, modeling and documenting events that affect each entity and sequence in which these events occur

Module 1 - Feasibility study
Stage 1 - Feasibility
Make sure project is viable and identify obstacles to a successful project
Prepare for study - assess scope
Define problem
Select feasibility option
Feasibility report produced

Module 2 - Requirements analysis
Gain full understanding of requirements and establish programme for the rest of the project
Stage 2 - Investigation of current environment
Data models used to look at data in current system
DFDs analyse physical and logical data flow

Data dictionaries represent detailed structure of information in current system
Requirements catalogue is produced with desirable features of new system

Stage 3 - Business system options
Agree upon functionality of new system, scope the project
Different options that meet all/some requirements from stage 2 are considered
One selected based on cost and review of best meeting requirements

Module 3 - Requirements specification
Stage 4 - Definition of requirements
Specification and logical design
Update data models and requirements catalogue
Logical entity model extended and normalised to 3NF
Requirements added and removed

Module 4 - Logical systems specification
Ensures smooth transition from requirements to working solution
Stage 5 - Technical system options
Identify constraints e.g. hardware, time, cost, security, etc.

Stage 6 - Logical design
Logical statement of what the system is required to do
How system will control operations + rules for validating data

Module 5 - Physical design
Stage 7 - Physical design
Develop all of the programs that are necessary to give the software its required functionality
Convert logical data model into an appropriate design and map onto a physical platform
Methodology stops, programming and testing start

SSM (Soft System Methodology)
Emphasis on human involvement
Model behaviour as part of systems analysis
Identify requirements from a system and anticipate reactions to change
Complements analysis and design in system development process

1. Determine problem situation - how does current process work, purpose, key players
2. Define problem situation through rich pictures - for non-technical staff

3. Define problem situation through root definitions - highlight change management issues, create root definitions using CATWOE
4. Build conceptual models - produce for each root definitions, define how the system works
5. Compare conceptual models with real world
6. Identify feasible and desirable changes - approaches needed to improve current situation are defined
7. Recommendations - how changes from 6 are implemented

Root definition - concise, tightly constructed description of human activity system which states what the system is
What human activity system is to be dealt with?
What problem is to be tackled?
Who, what, assumptions, environment

CATWOE
Client - affected by system and its activities
Actor - carry out activities within the system
Transformation - changes that takes place within or because of system
Weltanschauung (Worldview) - assumptions made about system
Owner - person to whom the system is answerable
Environment - wider system of which problem situation is a part