SYSTEM ANALYSIS AND DESIGN

ARYA S V

Lecturer in Computer Science School of Distance Education University of Kerala

OVERVIEW

- Introduction
- System Development Life Cycle
- Feasibility Study
- Cost Benefit Analysis
- Data Flow Diagram
- System Design
- Audit Trails
- Testing

Quality Assurance System Control Training Conversion Operating plan Hardware Selection Bench marking Vendor Selection Software Selection

INTRODUCTION

- Systems Analysis and Design is an active field in which analysts learn new approaches and different techniques for building the system more effectively and efficiently.
- The primary objective of systems analysis and design is to improve organizational systems.
- Systems development is systematic process which includes phases such as
 - Planning
 - Analysis
 - Design
 - Deployment
 - Maintenance

Characteristics of a System Elements of a System



Arya S. V, Lecturer in Computer Science, School of Distance Education, University of Kerala

System Development Life Cycle

System Development Life Cycle (SDLC) is a conceptual model which includes policies and procedures for developing or altering systems throughout their life cycles.

An effective System Development Life Cycle (SDLC) should result in a high quality system that meets customer expectations, reaches completion within time and cost evaluations, and works effectively and efficiently in the current and planned Information Technology infrastructure.



Project Review

- The decision to accept or reject a request can be made in different ways and by various members of the organization.
- Selecting projects for development is by committee.
- Steering committee method-It consists of key managers from various departments of the organization, as well as members of the information systems group.
- Information System Committee Method-The responsibility for reviewing project requests is assigned to a committee of managers, analysts in the information systems department.
- User-group committee method-In some departments form their own selection committees user ,group, committees controlling what is developed and when it is implemented.

Feasibility Considerations

- Economic Feasibility
- Determine the benefits and savings that are expected from a candidate system and compare them with costs.
- Technical Feasibility
- Technical feasibility centers around the existing computer system (hardware, software, etc.) and to what extent it can support the proposed addition.
- Behavioral Feasibility
- It evaluates and estimates the user attitude or behavior towards the development of new system.

Steps in Feasibility Analysis

- 1. Form a project team and appoint a project leader.
- 2. Prepare system flowcharts.
- 3. Enumerate potential candidate systems.
- 4. Describe and identify characteristics of candidate systems.
- 5. Determine and evaluate performance and cost effectiveness of each candidate system.
- 6. Weight system performance and cost data.
- 7. Select the best candidate system.
- 8. Prepare and report final project directive to management.

Cost Benefit Analysis

- Cost-Benefit Analysis is also known as CBA and Benefit-Cost Analysis
- Cost-Benefit Analysis involves adding up the benefits of a course of action, and then comparing these with the costs associated with it.
- CBA is a quick and simple technique that you can use for non-critical financial decisions.
- The decisions are mission-critical, or large sums of money are involved, other approaches – such as use of Net present values and Internal rates of return – are often more appropriate.

Costs

- Hardware costs relate to the actual purchase or lease of the computer and peripherals.
- **Personnel costs** include EDP staff salaries and benefits as well as pay for those involved in developing the system costs.
- Facility costs are expenses incurred in the preparation of the physical site where the application or the computer will be in operation.
- **Operating costs** include all costs associated with the day-today operation of the system.
- **Supply costs** are variable costs that increase with increased use of paper, ribbons, disks, and the like. They should be estimated and included in the overall cost of the system.

Steps for Cost/Benefit Determination

- Identify the costs and benefits pertaining to given project.
- 2. Categorize the various costs and benefits for analysis.
- 3. Select a method of evaluation.
- 4. Interpret the results of the analysis.
- 5. Take an action.



Evaluation Methods

Net benefit analysis: Net Present value analysis:

 $F = P (1 + i)^n$

Where,

- F= Future value of an investment
- P= Present value of the investment.
- i= Interest rate per compounding period.
 n= Number of years.

 $NPV = C/(1 + r)^n$

Where,

C=Cash flows in the time period

Image: Comparison of the second second

n=No of periods.

Fact Finding Techniques

- □ Interview: This method is used to collect the information from groups or individuals. Analyst selects the people who are related with the system for the interview.
- Questionnaire: It is the technique used to extract information from number of people. The Questionnaire consists of series of questions framed together in logical manner.
- Record Review: The information related to the system is published in the sources like newspapers, magazines, journals, documents etc. This record review helps the analyst to get valuable information about the system and the organization.
- Observation: Unlike the other fact finding techniques, in this method the analyst himself visits the organization and observes and understand the flow of documents, working of the existing system, the users of the system etc.

Data Flow Diagram (DFD)

- A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system.
- A DFD also known as a "bubble chart.
- A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both.
- □ It shows how information enters and leaves the system, what changes the information and where information is stored.
- The purpose of a DFD is to show the scope and boundaries of a system as a whole.
- It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

DFD Symbols

External Entity: An external entity can indicates a human, system or subsystem. External Entity represented by using square.

Process: It is a business activity or function where the manipulation and transformation of data take place. Process represented by using square a circle or a "bubble".



 Data Flow: A data flow represents the flow of information, with its direction represented by an arrowhead that shows at the end(s) of flow connector.

 Data Store: A data store represents the storage of persistent data required and produced by the process. An open rectangle represented as a data store- data.

Different levels of DFD

There are three main types of data-flow diagram:

- Context diagrams context diagram DFDs are diagrams that represent an overview of the system and its interaction with the rest of the "world".
- Level 1 data-flow diagrams Level 1 DFDs present a more detailed view of the system than context diagrams, by showing the main sub-processes and stores of data that make up the system as a whole.
- Level 2 (and lower) data-flow diagrams a major advantage of the data-flow modeling technique is that, through a technique called "leveling", the detailed complexity of real world systems can be managed and modeled in a hierarchy of abstractions.

System Design

System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. *"how to implement?"*

Types of System Design

- Logical Design: Logical design pertains to an abstract representation of the data flow, inputs, and outputs of the system.
- Physical Design: Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output.

Audit trails

- An audit trail (also called audit log) is a routine designed to allow the analyst, user or auditor to verify a process or an area in the new system.
- It is a security-relevant chronological record, set of records, and/or destination and source of records that provide documentary evidence of the sequence of activities that have affected at any time a specific operation, procedure, or event.
- The study of data as processed from step to step, an auditor may then trace all transactions that affect an account.
- In a manual system, the audit trail includes journals, ledgers and other documents used by auditor to trace transactions.
- In a computerized system, record content and format frequently make it difficult to trace a transaction completely.



Modularization

- Modularization is the division of a system or product into physically and functionally distinct units to allow removal and replacement.
- Each system or subsystem, from the highest to the lowest level, can be designed as a removable entity.

Advantages

- □ It can speed up the systems process in general & the computer programming function in particular.
- □ It eliminates unnecessary duplications.
- It provides better control over the total system project, since work can be segmented and assigned in smaller, more controllable units.
- Small parts of the system can be tested separately.

Arya S. V, Lecturer in Computer Science, School of Distance Education, University of Kerala



Coupling And Cohesion

- When a software program is modularized, its tasks are divided into several modules based on some characteristics. As we know, modules are set of instructions put together in order to achieve some tasks. They are though, considered as single entity but may refer to each other to work together. There are measures by which the quality of a design of modules and their interaction among them can be measured. These measures are called coupling and cohesion.
- Coupling is a measure that defines the level of interdependability among modules of a program. It tells at what level the modules interfere and interact with each other.
- □The cohesion of a component is a measure of the closeness of the relationship between its components.

Coupling

Content coupling – A module can directly access or modify or refer to the content of another module.

- Common coupling- Multiple modules have read and write access to some global data.
- Control coupling- Two modules are called control-coupled if one of them decides the function of the other module or changes its flow of execution.
- Stamp coupling- Multiple modules share common data structure and work on different part of it.
- Data coupling- Two modules interact with each other by means of passing data

Cohesion

- Co-incidental cohesion It is unplanned and random cohesion, which might be the result of breaking the program into smaller.
- Logical cohesion When logically categorized elements are put together into a module, it is called logical cohesion.
- Temporal Cohesion the elements of module are organized such that they are processed at a similar point in time.
- Procedural cohesion the elements of module are grouped together, which are executed sequentially in order to perform a task.
- Communicational cohesion When elements of module are grouped together, which are executed sequentially and work on same data
- Sequential cohesion When elements of module are grouped because the output of one element serves as input to another and so on.
- Functional cohesion Elements of module in functional cohesion are grouped because they all contribute to a single well-defined function. It can also be reused

Design Verification

- The output of software design process is design documentation, pseudo codes, detailed logic diagrams, process diagrams, and detailed description of all functional or non-functional requirements.
- □ If the outputs of design phase are in formal notation form, then their associated tools for verification should be used otherwise a thorough design review can be used for verification and validation.
- By structured verification approach, reviewers can detect defects that might be caused by overlooking some conditions. A good design review is important for good software design, accuracy and quality.

Testing

- □Testing is a process, to evaluate the functionality of a system with an intent to find whether the developed system met the specified requirements or not.
- A system testing is an expensive but critical process that can take as much as 50 percent of the budget for program development.
- The common view of testing held by users is that it is performed to prove that there are no errors in a program.
- The tester, who may be an analyst, programmer, or specialist trained in software testing, is actually trying to make the program fail.
- A successful test, then, is one that finds an error.

Test Plan

- The most important activity to ensure that there is initially a list of tasks and milestones in a baseline plan to track the progress of the project.
- It also defines the size of the test effort.
- It is the main document often called as master test plan or a project test plan and usually developed during the early phase of the project.

Test Case

- A test case is a set of data that the system will process as normal input.
- However, the data are created with the express intent of determining whether the system will process them correctly.
- Each test case is designed with the intent of finding errors in the way the system will process it.

SI. No	Test Plan Parameters	Description
1	Test plan identifier	Unique identifying reference.
2	Introduction	A brief introduction about the project and to the document.
3	Test items	A test item is a software item that is the application under test.
4	Features to be tested	A feature that needs to tested on the test ware.
5	Features not to be tested	Identify the features and the reasons for not including as part of testing.
6	Approach	Details about the overall approach to testing.
7	Item pass/fail	Documented whether a software item has passed or failed its test.
8	Test deliverables	The deliverables that are delivered as part of the testing process, such as test plans, test specifications and test summary reports.
9	Testing tasks	All tasks for planning and executing the testing.
10	Environmental needs	Defining the environmental requirements such as hardware, software
11	Responsibilities	Lists the roles and responsibilities of the team members.
12	Staffing and training needs	Captures the actual staffing requirements and any specific skills and training requirements.
13	Schedule	States the important project delivery dates and key milestones.
14	Risks and Mitigation	High-level project risks and assumptions and a mitigating plan for each identified risk.
15	Approvals	Captures all approvers of the document, their titles and the sign off date.

Test Planning Activities

- □ To determine the scope and the risks that need to be tested and that are NOT to be tested.
- Documenting Test Strategy.
- □ Making sure that the testing activities have been included.
- Deciding Entry and Exit criteria.
- Evaluating the test estimate.
- Planning when and how to test and deciding how the test results will be evaluated, and defining test exit criterion.
- The Test artifacts delivered as part of test execution

Types of testing

Manual Testing includes testing a software manually, i.e., without using any automated tool or any script.

- Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.
- □Black-Box Testing is a testing without having any knowledge of the interior workings of the application.
- ❑White-Box Testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called glass testing or open box testing.
- Grey-Box Testing is a technique to test the application with having a limited knowledge of the internal workings of an application.

Unit Testing

- In unit testing the analyst tests the programs making up a system. For this reason unit testing is sometimes called program testing.
- The software units in a system are the modules and routines that are assembled and integrated to perform a specific function. In a large system, many modules at different levels are needed.
- □Unit testing focuses first on the modules, independently of one another, to locate errors.
- This enables the tester to detect errors in coding and logic that are contained within that module alone.
- Those resulting from the interaction between modules are initially avoided.

Integration testing

- Integration testing does not test the software per se but rather the integration of each module in the system.
- □It also tests to find discrepancies between the system and its original objective, current specifications, and systems documentation.
- The system itself may not report this as an error, but the output may show unexpected results.
- If a record created and stored in one module, using the identification number as a numeric field, is later sought on retrieval with the expectation.
- Integration testing must also verify that file sizes are adequate and that indices have been built properly.

System Control and Quality Assurance

- A well-designed system should have controls to ensure proper operation and routine auditing. A candidate systems failure often results from lack of emphasis on data control.
- The programmer's ability to generate a code that reflects exactly the system specifications. These factors put an increasing burden on systems analysts to ensure the success of the system developed.
- The quality of a system depends on its design, development, testing and implementation.
- Quality assurance in a software organization is monitoring the engineering process and methods adopted to develop the software product in order to ensure conformance of quality as per organization standards.

Levels of Assurance

- □ **Testing** is the process of proving that there are no errors in a program. However, this is virtually impossible, since analysts cannot prove that software is free of errors.
- ■Verification is the process of checking a system achieves its goal without any bugs and verifies the developed product fulfills the requirements that we have.
- Validation is the process of checking whether the system has high level requirements. It checks whether the product we are developing is the right product or not. it is validation of actual and expected product.
- Certification is an endorsement of the correctness of the program, an issue that is rising in importance for information systems applications.

Training

Training is the process of engaging in a course or an educational process with users and analysts(systems operators).

Training systems operators

Systems operators training must ensure that they are able to handle all possible operations, both routine and extraordinary. Operator training must also involve the data entry personnel.

User Training

User training may involve equipment use, particularly in the case where, say, a microcomputer is in use and the individual involved is both operator and user. In these cases, user must be instructed first in how to operate the equipment.

Training Methods

Vendor &In-Service Training: In – house Training:

Most vendors offer extensive educational programs as part of their services, in some cases, there is a charge, but in many instances training is free.

The set of activities conducted by the employees within the organization or the company irrespective of the workers outside the organization.

Advantages of In-house Training

- Training Cost saving
- Travel Cost saving
- ✓ Team Building
- Capacity Building
- ✓ Convenience

Disadvantages of In-house Training

- Extra administration burden
- Not taken as seriously
- Not up to speed with current best practice

Conversion

Conversion is the process of changing form the old system to the new one.

Conversion Methods

- ✓ Parallel system
- Direct Cutover
- ✓ Pilot system
- ✓ Phase in

Method	Description	Advantages	Disadvantages
Parallel system	The old system is operated along with the new system	Offers greatest security. The old system can take over if errors are found in the new system or if usage problems occur	Doubles operating costs. The new system may not get fair trail.
Direct cutover	The old system replaced by the new one. The organization relies fully on the new system.	Forces users to make the new system work. There are immediate benefits from new methods and controls.	There is no other system to fall back on if difficulties arise with new system. Requires the most careful planning.
Pilot system	Working version of system implemented in one part of the organization. Based on feedback, changes are made and the system is installed in rest of the organization by one of the other methods	Provides experience and live test before implementation	May give the impression that the old system is unreliable and not error free.
Phase – in	Gradually implement system across all users.	Allows some users to take advantage of the system early. Allows training and installation without unnecessary use of resources.	A long phase – in causes user problems whether the project goes well (over enthusiasm) or not (resistance and lack of fair trial).

Conversion Plan

- The conversion plan includes a description of all the activities that must occur to implement the new system and put it into operation.
- During the pre-implementation stages, when the conversion is being planned, analysts should assemble a list of all tasks, including the following:
 - 1. List all files for conversion.
 - 2. Identify all data required to build new files during conversion.
 - 3. List all new documents and procedures that go into use during conversion.
 - Identify all controls to be used during conversion. Establish procedures for crosschecking the old and new systems. Determine how team members will know if something has not been completed properly.
 - 5. Assign responsibility for each activity.
 - 6. Verify conversion schedules.

Operating Plan

The operating plan is checking of all arrangements. It includes reviewing conversion plans, verifying the delivery of equipment, software, forms, preparing the site and preparing the data and files.

- Site Preparation: Analysts often work with vendor personnel to outline site preparation guidelines.
- Data and File Preparation: For a new system to begin master files and system files need to be entered into the system before the normal functioning of the system. Master files are generally created manually. The number of records in older system master file should tally with the number of records in new master file.

Hardware Selection

- Selecting a system is a serious and time concurring activity. Many systems are still selected based on vendor reputation only or other subjective factors.
- Systems capacity is frequently the determining factor. Relevant features to consider include the following:
 - Internal memory size
 - Cycle speed of system for processing
 - Characteristics of display and communication components
 - Types and numbers of auxiliary storage units that can be attached
 - Systems support and utility software provided or available



Benchmarking

- A benchmark is the application of synthetic programs to emulate the actual processing work handled by a computer system.
- Benchmark programs permit the submission of a mix of jobs that are representative of the users projected workload.
- Benchmarks can be run in virtually any type of system environment, including batch and online job streams, and with the users linked to the system directly or through telecommunications method.
- Common benchmarks are the speed of the central processor, with typical instruction executed in a set of programs, as well as multiple streams of jobs in a multiprogramming environment.

Financial Factors

- ❑ Rental-Computer rental is for the short term use of a system, generally form 1 to 12 months. Each month a payment is made for the use of the equipment. Both the user and supplier have the option of canceling the rental with advance notice, usually 30 or 60 days ahead of the termination date.
- □ Lease-As lease is a commitment to use a system for a specific time, generally from three to seven years. Payments are predetermined and do not change throughout the course of the lease. Depending on the terms of the lease, payments are monthly, quarterly, semiannual, or annual and include the cost of equipment service and maintenance.
- Purchase-The ownership of computers through outright purchase is the most common method of computer acquisition and is increasing in popularity as lease costs rise.

Method of acquisition	Advantages	Disadvantages
Rental	Short – term commitment. High level of flexibility. Does not require cash up front.	Most expensive option. Little control of equipment change. Not all vendors will rent.
Lease	Predetermined payments for fixed period. Does not require cash up front. Usually better service from vendor than under rental. Little risk of obsolescence. Less expensive than rental	More expensive than purchase. May have limitations on hours of equipment use.
Purchase	Least cost in long run. Distinct tax advantages if a profit – making firm. A business investment. Full control over equipment use.	Risk of obsolescence. Permanent commitment. Full responsibility for all problems. Greater early cash requirements than

other options.

Vendor Selection

- This step determines the "winner" among the list of vendors available.
- The vendor with the best combination of reputation, reliability, service record, training delivery time, lease / finance terms & conversion schedule is selected.
- Initially a decision is made as to which vendor to contact.
- sources available to check on vendors include:

- ✓ Users
- Software houses
- Trade Associations
- Universities
- Publications
- ✓ Vendor software list
- Vendor referral directories
- Published Directories
- Consultants
- Industry Contacts

Software Selection

Software selection is a critical Criteria for software selection aspect of system development.

- The search starts with the software, followed by the hardware.
- There are two ways of acquiring software: custom made or "off - the -shelf" packages.
- Today's trend is toward purchasing packages, which represent roughly 10 percent of what it costs to develop the same in house.

- Reliability
- Functionality
- Capacity
- ✓ Flexibility
- ✓ Usability
- ✓ Security
- Performance
- Serviceability
- ✓ Ownership
- Minimal costs

Advantages of Software Selection

- A good package can get the system running in a matter of days rather than the weeks or months required for "home-grown" packages.
- MIS personnel are released for other projects.
- Packages are generally reliable and perform according to stated documentation.
- Delays in completing software projects in house often occur because programmers quit in midstream.
- It is difficult to predict the cost of "home-grown" software.
- The user has a change of seeing how well the package performs before purchasing it.

Disadvantages of Software Selection

- The package may not meet user requirements adequately.
- Extensive modification of a package usually results in loss of the vendor's support.
- The methodology for package evaluation and selection is often poorly defined. The result is a haphazard review based on a faulty process or questionable selection criteria.
- For first time software package users, the overall expectation from a package is often unclear and ill defined.



Performance Evaluation

- Evaluating a system includes the hardware and software as a unit.
- Hardware selection requires an analysis of several performance categories.
 - System availability. When will the system be available?
 - Compatibility. How compatible is the system with existing programs?
 - Cost. What is the lease or purchase price of the system? What about maintenance and operation costs?
 - Performance. What are the capacity and throughput of the system?
 - Uptime. What is the 'uptime' record of the system? What maintenance schedule is required?
 - Support. How competent and available is the vendor's staff to support the system?
 - ✓ Usability. How easy is it to program, modify, and operate the system?



Performance Evaluation (Continue..)

- For the software evaluation, the following factors are considered:
 - The programming language and its suitability to the application(s).
 - ✓ Ease of installation and training.
 - Extent of enhancements to be made prior to installation.
- In addition to hardware/software evaluation, the quality of the vendor's services should be examined.
- Vendor support service includes the following:
 - ✓ Backup. Emergency computer backup available from vendor
 - Conversion. Programming and installation service provided during conversion.
 - ✓ Maintenance. Adequacy and cost of hardware maintenance.
 - System development. Availability of competent analysts and programmers for system development.

Thank You....