Dr. Vaibhav Sharma

SYSTEM

System is a set of components that interact to accomplish some purpose. e.g. College system, Economic system, Language system, a Business and its parts - Marketing, Sales, Research, Shipping, Accounting, Government.

Information System (I.S.): Interrelated components working together to collect, process, store, and disseminate information to support decision making, coordination control analysis and visualization in an organization.

Information: Data that have been shaped into a form that is meaningful and useful to human beings.

Data: Streams of raw facts representing events occurring in organizations.

Input: The capture or collection of raw data from within the organization or from its external environment.

Processing: The conversion, manipulation, and analysis of raw input into a form that is more meaningful to humans.

Output: The distribution of processed information to the people or activities where it will be used.

Feedback: Output that is returned to the appropriate members of the organization to help them evaluate or correct the input.

Computer-Based I.S. (CBIS): I.S. that rely on computer hardware and software for processing and disseminating information.The process of examining a (business) situation with the intent of improving it through better procedures and methods.

System Analysis - Process of gathering and interpreting facts, diagnosing problems, and using the facts to improve the system.

Systems Design - Process of planning a new system to replace or complement the old. Analysis specifies what the system should do and design states how to achieve the objective.

SYSTEM ANALYSIS AND DESIGN

A **system** is a set of interacting or interdependent component parts forming a complex/intricate whole. Every system is delineated by its spatial and temporal boundaries, surrounded and influenced by its environment, described by its structure and purpose and expressed in its functioning.

The term *system* may also refer to a set of rules that governs structure or behavior. Alternatively, and usually in the context of complex social systems, the term is used to describe the set of rules that govern structure or behavior.

The Term system is derived from the Greek word 'Systema' which means an organized relationship among functioning units or components

Elements of the System

A system has three basic elements input, processing and output. The other elements include control, feedback, boundaries, environment and interfaces.

1. **Input:** Input is what data the system receives to produce a certain output.
2. **Output:** What goes out from the system after being processed is known as Output.
3. **Processing:** The process involved to transform input into output is known as Processing.
4. **Control:** In order to get the desired results it is essential to monitor and control the input, Processing and the output of the system. This job is done by the control.
5. **Feedback:** The Output is checked with the desired standards of the output set and the necessary steps are taken for achieving the output as per the standards, this process is called as Feedback. It helps to achieve a much better control in the system.

6. **Boundaries:** The boundaries are nothing but the limit of the system. Setting up boundaries helps for better concentration of the actives carried in the system.
7. **Environment:** The things outside the boundary of the system are known as environment. Change in the environment affects the working of the system.
8. **Interfaces:** The interconnections and the interactions between the sub-systems is known as the Interfaces. They may be inputs and outputs of the systems.

Characteristics of a system:

1. Organization: It implies structure and order. It is the arrangement of components that helps to achieve objectives.

2. Interaction: It refers to the manner in which each component functions with other components of the system.

3. Interdependence: It means that parts of the organization or computer system depend on one another. They are coordinated and linked together according to a plan. One subsystem depends on the output of another subsystem for proper functioning.

4. Integration: It refers to the holism of systems. It is concerned with how a system is tied together.

5. Central Objective: A system should have a central objective. Objectives may be real or stated. Although a stated objective may be the real objective, it is not uncommon for an organization to state one objective and operates to achieve another. The important point is that users must know the central objective of a computer application early in the analysis for a successful design and conversion.

Types of the System

1. Conceptual Systems

a. Are theoretical and explanatory in the nature.

b. Provide the much needed clarification.

c. Provide theoretical framework for which there may or may not be any real life counterpart.

d. E.g. of such systems can be philosophy, theology etc.

2. Empirical Systems

a. Are very practical, specific and also very operational in the nature.

b. Can be based on the conceptual system.

c. Examination system, surgery act as very good examples of the empirical systems.

3. Open Systems

a. Involve continuous interaction with the environment.

b. So exchanges the information, material, energy with the environment.

c. Is open and also self organizing in the nature.

d. Is also adoptive or adaptive to the changing environment as it is flexible.

4. Closed Systems

a. Shuns any kind of the exchange with the environment.

b. Is rigid in nature.

c. Is not at all amenable to the change.

d. Is also self contained.

e. Is somewhat isolated in the nature.

f. Is having a well defined boundary.

g. Is not at all adaptive in the nature.

5. Natural Systems

a. Such Systems exist and also abound in the nature.

b. Are also not at all the results of the human endeavors.

c. Rivers, mountains, minerals etc. are the major examples of the natural Systems.

6. Artificial Systems

a. Are manufactured (man made).

b. Examples of such Systems are dams, canals, roads, machines, factories etc.

7. Probabilistic Systems

a. Based on the predictability of the behavior or the outcome.

8. Deterministic Systems

a. In such Systems, the interaction of the elements is known.
b. As the behavior of the elements is pre-determined, it becomes possible to work upon the reaction well in the advance.

INFORMATION SYSTEM

Information system (IS) is any organized system for the collection, organization, storage and communication of information. More specifically, it is the study of complementary networks that people and organizations use collect, filter, process, create and distribute data.

A computer Information system is a system composed of people and computers that processes or interprets information. The term is also sometimes used in more restricted senses to refer to only the software used to run a computerized database or to refer to only a computer system.Information system is an academic study of systems with a specific reference to information and the complementary networks of hardware and software that people and organizations use to collect, filter, process, create and also distribute data. An emphasis is placed on an Information System having a definitive Boundary, Users, Processors, Stores, Inputs, Outputs and the aforementioned communication networks.Any specific information system aims to support operations, management and decision-making. An information system is the information and communication technology (ICT) that an organization uses, and also the way in which people interact with this technology in support of business processes.

Information systems typically include an ICT component but are not purely concerned with ICT, focusing instead on the end use of information technology. Information systems are also different from business processes. Information systems help to control the performance of business processes.

A work system is a system in which humans or machines perform processes and activities using resources to produce specific products or services for customers. An information system is a work system whose activities are devoted to capturing, transmitting, storing, retrieving, manipulating and displaying information.

As such, information systems inter-relate with data systems on the one hand and activity systems on the other. An information system is a form of communication system in which data represent and are processed as a form of social memory. An information system can also be considered a semi-formal language which supports human decision making and action. Information systems are the primary focus of study for organizational informatics.

System Analysis and Design (SAD)

A collection of components that work together to realize some objectives forms a system. Basically there are three major components in every system, namely input, processing and output.

In a system the different components are connected with each other and they are interdependent. For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system and so forth. The objective of the system demands that some output is produced as a result of processing the suitable inputs. A well-designed system also includes an additional element referred to as 'control' that provides a feedback to achieve desired objectives of the system.

SYSTEM DEVELOPMENT LIFE CYCLE

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system.

System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the System Analysis and Design terminology, the system development life cycle also means software development life cycle. Following are the different phases of system development life cycle:

1. **Preliminary Study**
2. **Feasibility study**
3. **Detailed system study**
4. **System analysis**
5. **System design**

6. Coding

7. Testing

8. Implementation

9. Maintenance

The different phases of system development life cycle is shown in this diagram

PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE

1. Preliminary System Study

Preliminary system study is the first stage of system development life cycle. This is a brief investigation of the system under consideration and gives a clear picture of what actually the physical system is? In practice, the initial system study involves the preparation of a 'System Proposal' which lists the Problem Definition, Objectives of the Study, Terms of reference for Study, Constraints and Expected benefits of the new system, etc. in the light of the user requirements. The system proposal is prepared by the System Analyst (who studies the system) and places it before the user management. The management may accept the proposal and the cycle proceeds to the next stage. The management may also reject the proposal or request some modifications in the proposal. In summary, we would say that system study phase passes through the following steps:

- Problem identification and project initiation
- Background analysis
- Inference or findings (system proposal)

2. Feasibility Study

In case the system proposal is acceptable to the management, the next phase is to examine the feasibility of the system. The feasibility study is basically the test of the proposed system in the light of its workability, meeting user's requirements, effective use of resources and of course, the cost effectiveness. These are categorized as technical, operational, economic and schedule feasibility. The main goal of feasibility study is not to solve the problem but to achieve the scope. In the process of feasibility study, the cost and benefits are estimated with greater accuracy to find the Return on Investment (ROI). This also defines the resources needed to complete the detailed investigation. The result is a feasibility report submitted to the management. This may be accepted or accepted with modifications or rejected. The system cycle proceeds only if the management accepts it.

3. Detailed System Study

The detailed investigation of the system is carried out in accordance with the objectives of the proposed system. This involves detailed study of various operations performed by a system and their relationships within and outside the system. During this process, data are collected on the available files, decision points and transactions handled by the present system. Interviews, on-site observation and questionnaire are the tools used for detailed system study. Using the following steps it becomes easy to draw the exact boundary of the new system under consideration:

- Keeping in view the problems and new requirements

- Workout the pros and cons including new areas of the system

All the data and the findings must be documented in the form of detailed data flow diagrams (DFDs), data dictionary, logical data structures and miniature specification. The main points to be discussed in this stage are:

- Specification of what the new system is to accomplish based on the user requirements.

- Functional hierarchy showing the functions to be performed by the new system and their relationship with each other.

- Functional network, which are similar to function hierarchy but they highlight the functions which are common to more than one procedure.

- List of attributes of the entities – these are the data items which need to be held about each entity (record).

4. System Analysis

Systems analysis is a process of collecting factual data, understand the processes involved, identifying problems and recommending feasible suggestions for improving the system functioning. This involves studying the business processes, gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weaknesses of the system so as to achieve the organizational goals. System Analysis also includes subdividing of complex process involving the entire system, identification of data store and manual processes.

The major objectives of systems analysis are to find answers for each business process: What is being done, How is it being done, Who is doing it, When is he doing it, Why is it being done and How can it be improved? It is more of a thinking process and involves the creative skills of the System Analyst. It attempts to give birth to a new efficient system that satisfies the current needs of the user and has scope for future growth within the organizational constraints. The result of this process is a logical

system design. Systems analysis is an iterative process that continues until a preferred and acceptable solution emerges.

5. System Design

Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. It is the most crucial phase in the developments of a system. The logical system design arrived at as a result of systems analysis is converted into physical system design. Normally, the design proceeds in two stages:

- Preliminary or General Design: In the preliminary or general design, the features of the new system are specified. The costs of implementing these features and the benefits to be derived are estimated. If the project is still considered to be feasible, we move to the detailed design stage.

- Structured or Detailed Design: In the detailed design stage, computer oriented work begins in earnest. At this stage, the design of the system becomes more structured. Structure design is a blue print of a computer system solution to a given problem having the same components and inter-relationships among the same components as the original problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up in detail.

In the design stage, the programming language and the hardware and software platform in which the new system will run are also decided. There are several tools and techniques used for describing the system design of the system. These tools and techniques are:

1. Flowchart

2. Data flow diagram (DFD)

3. Data dictionary

4. Structured English

5. Decision table

6. Decision tree

The system design involves:

a. Defining precisely the required system output

b. Determining the data requirement for producing the output

c. Determining the medium and format of files and databases

d. Devising processing methods and use of software to produce output

e. Determine the methods of data capture and data input

f. Designing Input forms

g. Designing Codification Schemes

h. Detailed manual procedures

i. Documenting the Design

6. Coding

The system design needs to be implemented to make it a workable system. This demands the coding of design into computer understandable language, i.e., programming language. This is also called the programming phase in which the programmer converts the program specifications into computer instructions, which we refer to as programs. It is an important stage where the defined procedures are transformed into control specifications by the help of a computer language. The programs coordinate the data movements and control the entire process in a system. It is generally felt that the programs must be modular in nature. This helps in fast development, maintenance and future changes, if required.

7. Testing

Before actually implementing the new system into operation, a test run of the system is done for removing the bugs, if any. It is an important phase of a successful system. After codifying the whole programs of the system, a test plan should be developed and run on a given set of test data. The output of the test run should match the expected results. Sometimes, system testing is considered a part of implementation process.

Using the test data following test run are carried out:

- Program test: When the programs have been coded, compiled and brought to working conditions, they must be individually tested with the prepared test data. Any undesirable happening must be noted and debugged (error corrections)

- System Test: After carrying out the program test for each of the programs of the system and errors removed, then system test is done. At this stage the test is done on actual data. The complete system is executed on the actual data. At each stage of the execution, the results or output of the system is analysed. During the result analysis, it may be found that the outputs are not matching the expected output of the system. In such case, the errors in the particular programs are identified and are fixed and further

tested for the expected output. When it is ensured that the system is running error-free, the users are called with their own actual data so that the system could be shown running as per their requirements.

8. Implementation

After having the user acceptance of the new system developed, the implementation phase begins. Implementation is the stage of a project during which theory is turned into practice. The major steps involved in this phase are:

- Acquisition and Installation of Hardware and Software
- Conversion
- User Training
- Documentation

The hardware and the relevant software required for running the system must be made fully operational before implementation. The conversion is also one of the most critical and expensive activities in the system development life cycle. The data from the old system needs to be converted to operate in the new format of the new system. The database needs to be setup with security and recovery procedures fully defined.

During this phase, all the programs of the system are loaded onto the user's computer. After loading the system, training of the user starts. Main topics of such type of training are:

- How to execute the package
- How to enter the data
- How to process the data (processing details)
- How to take out the reports

After the users are trained about the computerized system, working has to shift from manual to computerized working. The process is called 'Changeover'. The following strategies are followed for changeover of the system.

(i) Direct Changeover: This is the complete replacement of the old system by the new system. It is a risky approach and requires comprehensive system testing and training.

(ii) Parallel run: In parallel run both the systems, i.e., computerized and manual, are executed simultaneously for certain defined period. The same data is processed by both the systems. This strategy is less risky but more expensive because of the following:

- Manual results can be compared with the results of the computerized system.

- The operational work is doubled.

- Failure of the computerized system at the early stage does not affect the working of the organization, because the manual system continues to work, as it used to do.

(iii) Pilot run: In this type of run, the new system is run with the data from one or more of the previous periods for the whole or part of the system. The results are compared with the old system results. It is less expensive and risky than parallel run approach. This strategy builds the confidence and the errors are traced easily without affecting the operations. The documentation of the system is also one of the most important activity in the system development life cycle. This ensures the continuity of the system. There are generally two types of documentation prepared for any system. These are:

- User or Operator Documentation

- System Documentation

The user documentation is a complete description of the system from the users point of view detailing how to use or operate the system. It also includes the major error messages likely to be encountered by the users. The system documentation contains the details of system design, programs, their coding, system flow, data dictionary, process description, etc. This helps to understand the system and permit changes to be made in the existing system to satisfy new user needs.

9. Maintenance

Maintenance is necessary to eliminate errors in the system during its working life and to tune the system to any variations in its working environments. It has been seen that there are always some errors found in the systems that must be noted and corrected. It also means the review of the system from time to time. The review of the system is done for:

- knowing the full capabilities of the system

- knowing the required changes or the additional requirements

- studying the performance.

If a major change to a system is needed, a new project may have to be set up to carry out the change. The new project will then proceed through all the above life cycle phases.

What is a 'Feasibility Study'

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it.

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below.

- To analyze whether the software will meet organizational requirements
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule
- To determine whether the software can be integrated with other existing software.

Types of Feasibility

Various types of feasibility that are commonly considered include technical feasibility, operational feasibility, and economic feasibility.

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. Technical feasibility also performs the following tasks.

- Analyzes the technical skills and capabilities of the software development team members
- Determines whether the relevant technology is stable and established

- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority

- Determines whether the solution suggested by the software development team is acceptable

- Analyzes whether users will adapt to a new software

- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization

- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis)

- Cost of hardware, software, development team, and training.

Feasibility Study Process

Feasibility study comprises the following steps.

1. Information assessment: Identifies information about whether the system helps in achieving the objectives of the organization. It also verifies that the system can be implemented using new technology and within the budget and whether the system can be integrated with the existing system.

2. Information collection: Specifies the sources from where information about software can be obtained. Generally, these sources include users (who will operate the software), organization (where the software will be used), and the software development team (which understands user requirements and knows how to fulfill them in software).

3. Report writing: Uses a feasibility report, which is the conclusion of the feasibility study by the software development team. It includes the recommendations whether the software development should continue. This report may also include information about changes in the software scope, budget, and schedule and suggestions of any requirements in the system.

4. General information: Describes the purpose and scope of feasibility study. It also describes system overview, project references, acronyms and abbreviations, and points of contact to be used. System overview provides description about the name of the organization responsible for the software development, system name or title, system category, operational status, and so on. Project references provide a list of the references used to prepare this document such as documents relating to the project or previously developed documents that are related to the project. Acronyms and abbreviations provide a list of the terms that are used in this document along with their meanings. Points of contact provide a list of points of organizational contact with users for information and coordination. For example, users require assistance to solve problems (such as troubleshooting) and collect information such as contact number, e-mail address, and so on.

5. Management summary: Provides the following information.

6. Environment: Identifies the individuals responsible for software development. It provides information about input and output requirements, processing requirements of the software and the interaction of the software with other software. It also identifies system security requirements and the system's processing requirements

7. Current functional procedures: Describes the current functional procedures of the existing system, whether automated or manual. It also includes the data-flow of the current system and the number of team members required to operate and maintain the software.

8. Functional objective: Provides information about functions of the system such as new services, increased capacity, and so on.

9. Performance objective: Provides information about performance objectives such as reduced staff and equipment costs, increased processing speeds of software, and improved controls.

10. Assumptions and constraints: Provides information about assumptions and constraints such as operational life of the proposed software, financial constraints,

changing hardware, software and operating environment, and availability of information and sources.

11. Methodology: Describes the methods that are applied to evaluate the proposed software in order to reach a feasible alternative. These methods include survey, modeling, benchmarking, etc.

12. Evaluation criteria: Identifies criteria such as cost, priority, development time, and ease of system use, which are applicable for the development process to determine the most suitable system option.

13. Recommendation: Describes a recommendation for the proposed system. This includes the delays and acceptable risks.

14. Proposed software: Describes the overall concept of the system as well as the procedure to be used to meet user requirements. In addition, it provides information about improvements, time and resource costs, and impacts. Improvements are performed to enhance the functionality and performance of the existing software. Time and resource costs include the costs associated with software development from its requirements to its maintenance and staff training. Impacts describe the possibility of future happenings and include various types of impacts as listed below.

15. Equipment impacts: Determine new equipment requirements and changes to be made in the currently available equipment requirements.

16. Software impacts: Specify any additions or modifications required in the existing software and supporting software to adapt to the proposed software.

17. Organizational impacts: Describe any changes in organization, staff and skills requirement.

18. Operational impacts: Describe effects on operations such as user-operating procedures, data processing, data entry procedures, and so on.

19. Developmental impacts: Specify developmental impacts such as resources required to develop databases, resources required to develop and test the software, and specific activities to be performed by users during software development.

20. Security impacts: Describe security factors that may influence the development, design, and continued operation of the proposed software.

21. Alternative systems: Provide description of alternative systems, which are considered in a feasibility study. This also describes the reasons for choosing a particular alternative system to develop the proposed software and the reason for rejecting alternative systems.

SYSTEM ANALYST

A systems analyst is an information technology (IT) professional who specializes in analyzing, designing and implementing information systems. Systems analysts assess the suitability of information systems in terms of their intended outcomes and liaise with end users, software vendors and programmers in order to achieve these outcomes. A systems analyst is a person who uses analysis and design techniques to solve business problems using information technology. Systems analysts may serve as change agents who identify the organizational improvements needed, design systems to implement those changes, and train and motivate others to use the systems.

Although they may be familiar with a variety of programming languages, operating systems, and computer hardware platforms, they do not normally involve themselves in the actual hardware or software development. They may be responsible for developing cost analysis, design considerations, staff impact amelioration, and implementation timelines.

A systems analyst is typically confined to an assigned or given system and will often work in conjunction with a business analyst. These roles, although having some overlap, are not the same. A business analyst will evaluate the business need and identify the appropriate solution and, to some degree, design a solution without diving too deep into its technical components, relying instead on a systems analyst to do so. A systems analyst will often evaluate code, review scripting and, possibly, even modify such to some extent.

Some dedicated professionals possess practical knowledge in both areas (business and systems analysis) and manage to successfully combine both of these occupations, effectively blending the line between business analyst and systems analyst.

Roles of System Analyst

A systems analyst may:

- Identify, understand and plan for organizational and human impacts of planned systems, and ensure that new technical requirements are properly integrated with existing processes and skill sets.

- Plan a system flow from the ground up.

- Interact with internal users and customers to learn and document requirements that are then used to produce business requirements documents.

- Write technical requirements from a critical phase.

- Interact with designers to understand software limitations.
- Help programmers during system development, e.g. provide use cases, flowcharts or even database design.
- Perform system testing.
- Deploy the completed system.
- Document requirements or contribute to user manuals.
- Whenever a development process is conducted, the system analyst is responsible for designing components and providing that information to the developer.

Roles of the Systems Analyst

The systems analyst systematically assesses how users interact with technology and how businesses function by examining the inputting and processing of data and the outputting of information with the intent of improving organizational processes. Many improvements involve better support of users' work tasks and business functions through the use of computerized information systems. This definition emphasizes a systematic, methodical approach to analyzing—and potentially improving—what is occurring in the specific context experienced by users and created by a business.

Our definition of a systems analyst is necessarily broad. The analyst must be able to work with people of all descriptions and be experienced in working with computers. The analyst plays many roles, sometimes balancing several at the same time. The three primary roles of the systems analyst are consultant, supporting expert, and agent of change.

1. Systems Analyst as Consultant

The systems analyst frequently acts as a systems consultant to humans and their businesses and, thus, may be hired specifically to address information systems issues within a business. Such hiring can be an advantage because outside consultants can bring with them a fresh perspective that other people in an organization do not possess. It also means that outside analysts are at a disadvantage because an outsider can never know the true organizational culture. As an outside consultant, you will rely heavily on the systematic methods discussed throughout this text to analyze and design appropriate information systems for users working in a particular business. In addition, you will rely on information systems users to help you understand the organizational culture from others' viewpoints.

2. Systems Analyst as Supporting Expert

Another role that you may be required to play is that of supporting expert within a business for which you are regularly employed in some systems capacity. In this role the analyst draws on professional expertise concerning computer hardware and software and their uses in the business. This work is often not a full-blown systems project, but rather it entails a small modification or decision affecting a single department.

As the supporting expert, you are not managing the project; you are merely serving as a resource for those who are. If you are a systems analyst employed by a manufacturing or service organization, many of your daily activities may be encompassed by this role.

3. Systems Analyst as Agent of Change

The most comprehensive and responsible role that the systems analyst takes on is that of an agent of change, whether internal or external to the business. As an analyst, you are an agent of change whenever you perform any of the activities in the systems development life cycle (discussed in the next section) and are present and interacting with users and the business for an extended period (from two weeks to more than a year). An agent of change can be defined as a person who serves as a catalyst for change, develops a plan for change, and works with others in facilitating that change.

Your presence in the business changes it. As a systems analyst, you must recognize this fact and use it as a starting point for your analysis. Hence, you must interact with users and management (if they are not one and the same) from the very beginning of your project. Without their help you cannot understand what they need to support their work in the organization, and real change cannot take place.

If change (that is, improvements to the business that can be realized through information systems) seems warranted after analysis, the next step is to develop a plan for change along with the people who must enact the change. Once a consensus is reached on the change that is to be made, you must constantly interact with those who are changing.

As a systems analyst acting as an agent of change, you advocate a particular avenue of change involving the use of information systems. You also teach users the process of change, because changes in the information system do not occur independently; rather, they cause changes in the rest of the organization as well.

4. Qualities of the Systems Analyst

From the foregoing descriptions of the roles the systems analyst plays, it is easy to see that the successful systems analyst must possess a wide range of qualities. Many different kinds of people are systems analysts, so any description is destined to fall

short in some way. There are some qualities, however, that most systems analysts seem to display.

Above all, the analyst is a problem solver. He or she is a person who views the analysis of problems as a challenge and who enjoys devising workable solutions. When necessary, the analyst must be able to systematically tackle the situation at hand through skillful application of tools, techniques, and experience. The analyst must also be a communicator capable of relating meaningfully to other people over extended periods of time. Systems analysts need to be able to understand humans' needs in interacting with technology, and they need enough computer experience to program, to understand the capabilities of computers, to glean information requirements from users, and to communicate what is needed to programmers. They also need to possess strong personal and professional ethics to help them shape their client relationships.

The systems analyst must be a self-disciplined, self-motivated individual who is able to manage and coordinate other people, as well as innumerable project resources. Systems analysis is a demanding career, but, in compensation, an ever-changing and always challenging one.

Service- Analysis / Service-Oriented Analysis

Service Analysis represents one of the early stages in an SA initiative and the first phase in the service delivery cycle. It is a process that begins with preparatory information gathering steps that are completed in support of a service modeling a sub-process that results in the creation of conceptual service candidates, service capability candidates, and service composition candidates

The Service-Oriented Analysis process is commonly carried out iteratively, once for each business process. Typically, the delivery of a service inventory determines a scope that represents a meaningful domain of the enterprise, or the enterprise as a whole. All iterations of a Service-Oriented Analysis then pertain to that scope, with each iteration contributing to the service inventory blueprint.

A key success factor of the Service-Oriented Analysis process is the hands-on collaboration of both Business Analysts and Technology Architects. The former group is especially involved in the definition of service candidates within a business-centric functional context because they understand the business processes used as input for the analysis and because service-orientation aims to align business and IT more closely.

Figure 1 - A generic Service-Oriented Analysis process that can be further customized. The first two steps essentially collect information in preparation for a detailed service modeling sub-process.

Efficiency Analysis

Efficiency signifies a level of performance that describes a process that uses the lowest amount of inputs to create the greatest amount of outputs. Efficiency relates to the use of all inputs in producing any given output, including personal time and energy. Efficiency is a measurable concept that can be determined by determining the ratio of useful output to total input. It minimizes the waste of resources such as physical materials, energy and time, while successfully achieving the desired output.

Impacts of Efficiency

An efficient society is better able to serve its citizens and operate in a healthy manner. When goods are produced efficiently, they are often able to be sold at a lower price. The advances that have been made in efficiency have also allowed those affected to maintain a higher standard of living, including living in homes with electricity and running water and getting around more quickly. Efficiency results in a sharp drop in hunger and malnutrition, as goods are able to be moved farther and more quickly. Also, advances in efficiency have allowed the work week to decline considerably. More work can now be performed in a shorter amount of time, so it is no longer necessary to spend those extra hours laboring.

Efficiency is an important attribute because all inputs are scarce. Time, money and raw materials are limited, so it makes sense to try to conserve them while maintaining an acceptable level of output or a general production level.

Being efficient simply means reducing the amount of wasted inputs.

1. Economic efficiency

Economic efficiency implies an economic state in which every resource is optimally allocated to serve each individual or entity in the best way while minimizing waste and inefficiency. When an economy is economically efficient, any changes made to assist one entity would harm another. In terms of production, goods are produced at their lowest possible cost, as are the variable inputs of production.

Economic Efficiency and Scarcity

The principles of economic efficiency are based on the concept that resources are scarce. Therefore, there are not enough resources to have all aspects of an economy functioning at their highest capacity at all times. Instead, the scarce resources must be

distributed to meet the needs of the economy in an ideal way while also limiting the amount of waste produced. The ideal state is related to the welfare of the population as a whole with peak efficiency also resulting in the highest level of welfare possible based on the resources available.

2. Production efficiency

Production efficiency is an economic level at which the economy can no longer produce additional amounts of a good without lowering the production level of another product. This happens when an economy is operating along its production possibility frontier. Efficient production is achieved when a product is created at its lowest average total cost; production efficiency measures whether the economy is producing as much as possible without wasting precious resources.

3. efficiency ratio

The efficiency ratio is typically used to analyze how well a company uses its assets and liabilities internally. An efficiency ratio can calculate the turnover of receivables, the repayment of liabilities, the quantity and usage of equity, and the general use of inventory and machinery. This ratio can also be used to track and analyze the performance of commercial and investment banks.

Performance Analysis

A performance analysis methodology is a procedure that you can follow to analyze system or application performance. These generally provide a starting point and then guidance to root cause, or causes. Different methodologies are suited for solving different classes of issues, and you may try more than one before accomplishing your goal.

Analysis without a methodology can become a fishing expedition, where metrics are examined ad hoc, until the issue is found – if it is at all.

Methodologies documented in more detail on this site are:

- The USE Method: The Utilization Saturation and Errors (USE) Method is a methodology for analyzing the performance of any system. It directs the construction of a checklist, which for server analysis can be used for quickly identifying resource bottlenecks or errors. It begins by posing questions, and then seeks answers, instead of beginning with given metrics (partial answers) and trying to work backwards. Finally it is used for finding resource bottlenecks.

- The TSA Method: There are two basic performance analysis methodologies you can use for most performance issues. The first is the resource-oriented USE Method, which provides a checklist for identifying common bottlenecks and errors. The second is the thread-oriented TSA Method, for identifying issues causing poor thread performance.

The TSA Method is complementary to the USE Method as it has a different perspective: threads instead of resources. Like the USE Method, it provides a starting point for analysis, and then narrows the investigation to the problem area. Finally it is used for analyzing application time.

- Off-CPU Analysis: Off-CPU analysis is a performance methodology where high resolution off-CPU time is measured and studied. This reveals which code-paths led to time spent waiting (blocked), and can be a quick and generic way to root-cause a wide range of performance issues. finally it is used for analyzing any type of thread wait latency

- Active Benchmarking: With active benchmarking, we analyze performance while the benchmark is still running (not just after it's done), using other tools. You can confirm that the benchmark tests what you intend it to, and that you understand what that is. Data becomes Information. This can also identify the true limiters of the system under test, or of the benchmark itself. Finally it is used for accurate and successful benchmarking

Efficiency Analysis

Efficiency concerns how to generate as much output with minimum input. Here is an indication that a system can be said to be inefficient:

1. Much time is wasted on the activities of human resources, machines, or computers.

2. Data is input or copied to excess.

3. Data processed in excess.

4. Information is generated in excess.

5. Effort required for these tasks are too excessive.

6. Material required for these tasks are too excessive.

FACT ANALYSIS

Once a data has been collected it is must to bring the information in an organized & evaluated & conclusion drawn for preparing a report to the used for final review and approval. There are many tools, which are used for organizing & analysis on collected information. The tools used for that purpose are input/output analysis, decision tables & structured chart.

Input/Output Analysis

This analysis identifies the elements that are related to the input/output of the given system. The data flow diagram & flowcharts are the best way to show the input/output analysis.

Dataflow Diagram

It is the best way to show the input output analysis it enables the user to focus on the logic of the system & developed feasible alternatives. The circle represents the processing point with in the system. The square are department or no of people involved in the system the major's steps in the billing process are extracting the customer account, applying for renewable, preparing the bill processing the payment and account for cash receipt.

Decision Table

This table described the data flow within the system. It is used when a complex decision logic can't be represented clearly in the flow chart as a documenting tool they provide a simpler form of data analysis then the flow chart. It is easy to follow communication devices between technical & non-technical person:

Structured Chart

There are several variation in the working tool of structured chart .the analyst start with single input /output processing/ out put chart, locate the module associated with I/O processing.

BACKGROUND ANALYSIS

Once the project is initiated the analyst begin to learn about the settings the existing system & the physical process related to the revised system. For eg.Its important to understand structure of the bank, who runs it, who report's to whom in the safe deposit area. The relationship between the safe deposit & the teller line, accounting & customer

service. The nature , frequency & level of interaction between the staff & the department . it could be reorganized the staff to get the better customer service. Therefore analyst should prepare origination chart the list of function's required along with the people who performed them.

Fact finding technique: analyst began to collect the information on the existing system or making a new system's output, input & cost. The tool use in collection of information are :

1. conducting interview of the user's,

2. on site observation,

3. questionnaires,

4. review of written documents.

Service- Analysis / Service-Oriented Analysis

Service Analysis represents one of the early stages in an SA initiative and the first phase in the service delivery cycle. It is a process that begins with preparatory information gathering steps that are completed in support of a service modeling a sub-process that results in the creation of conceptual service candidates, service capability candidates, and service composition candidates

The Service-Oriented Analysis process is commonly carried out iteratively, once for each business process. Typically, the delivery of a service inventory determines a scope that represents a meaningful domain of the enterprise, or the enterprise as a whole. All iterations of a Service-Oriented Analysis then pertain to that scope, with each iteration contributing to the service inventory blueprint.

A key success factor of the Service-Oriented Analysis process is the hands-on collaboration of both Business Analysts and Technology Architects. The former group is especially involved in the definition of service candidates within a business-centric functional context because they understand the business processes used as input for the analysis and because service-orientation aims to align business and IT more closely.

A generic Service-Oriented Analysis process that can be further customized. The first two steps essentially collect information in preparation for a detailed service modeling sub-process.

Efficiency Analysis

Efficiency signifies a level of performance that describes a process that uses the lowest amount of inputs to create the greatest amount of outputs. Efficiency relates to the use

of all inputs in producing any given output, including personal time and energy. Efficiency is a measurable concept that can be determined by determining the ratio of useful output to total input. It minimizes the waste of resources such as physical materials, energy and time, while successfully achieving the desired output.

Impacts of Efficiency

An efficient society is better able to serve its citizens and operate in a healthy manner. When goods are produced efficiently, they are often able to be sold at a lower price. The advances that have been made in efficiency have also allowed those affected to maintain a higher standard of living, including living in homes with electricity and running water and getting around more quickly. Efficiency results in a sharp drop in hunger and malnutrition, as goods are able to be moved farther and more quickly. Also, advances in efficiency have allowed the work week to decline considerably. More work can now be performed in a shorter amount of time, so it is no longer necessary to spend those extra hours laboring.

Efficiency is an important attribute because all inputs are scarce. Time, money and raw materials are limited, so it makes sense to try to conserve them while maintaining an acceptable level of output or a general production level.

Being efficient simply means reducing the amount of wasted inputs.

Economic efficiency

Economic efficiency implies an economic state in which every resource is optimally allocated to serve each individual or entity in the best way while minimizing waste and inefficiency. When an economy is economically efficient, any changes made to assist one entity would harm another. In terms of production, goods are produced at their lowest possible cost, as are the variable inputs of production.

Economic Efficiency and Scarcity

The principles of economic efficiency are based on the concept that resources are scarce. Therefore, there are not enough resources to have all aspects of an economy functioning at their highest capacity at all times. Instead, the scarce resources must be distributed to meet the needs of the economy in an ideal way while also limiting the amount of waste produced. The ideal state is related to the welfare of the population as a whole with peak efficiency also resulting in the highest level of welfare possible based on the resources available.

Production efficiency

Production efficiency is an economic level at which the economy can no longer produce additional amounts of a good without lowering the production level of another product. This happens when an economy is operating along its production possibility frontier.

Efficient production is achieved when a product is created at its lowest average total cost; production efficiency measures whether the economy is producing as much as possible without wasting precious resources

Efficiency ratio

The efficiency ratio is typically used to analyze how well a company uses its assets and liabilities internally. An efficiency ratio can calculate the turnover of receivables, the repayment of liabilities, the quantity and usage of equity, and the general use of inventory and machinery. This ratio can also be used to track and analyze the performance of commercial and investment banks.

PROBLEM DEFINITION

System analysis involves examining the business situation through which it is determined how to find a solution for a problem or develop a system successfully. This activity involves breaking the total development process in to smaller activities or phases that the actual task may be done in a smooth manner. In order that a successful system may be designed, developed and implemented, there is a great need and importance of defining a problem, so that the solution may be ascertained accordingly.

This leads to the phase of system investigation. The phase of system investigation involves defining the problem clearly. In order that it may be done as conveniently as possible, an analyst or a consultant is invited, so that the user can define the problem of which, at a later stage, the solution may be ascertained. This phase involves determining whether there is a need for developing a completely new system or the existing system can be modified or improved. It's just a preliminary stage which describes the business system. In this way, preliminary investigation is the first step in the system development project.

System investigation includes problem definition in a more detailed way. It is important that it should be done in such a way that it can be ensured all essential points are covered up. It therefore leads to a stage in which there is a requirement of problem definition in a very clearer way. This necessitates interviewing and clarifying issues with the users. So, a system analyst needs to discuss with the users and prepare a written statement covering the objectives and scope of the problem. In this way, it becomes important to prepare a statement of objectives while understanding the problem and defining it clearly. Here are some possible definitions of problems:

System analyst needs to consider and clarify whether the existing system is working well or there is any problem in terms of speed or process etc. It also includes observing if the existing system is giving proper response and meets the requirement of the users. Also, determining that the system is accessing the workload successfully and there are not any technical hitches while it works is also included during this process. Furthermore, system analyst needs to ensure the accuracy and reliability of the system. Apart from that, considering the cost of a system to be implemented is one of the most

important aspects, a system analyst needs to provide a rough estimation of cost involved. These things are to be done in a timely manner and performed with quality before it is too late in the system development process.

PROJECT INITIATION

The initiation stage determines the nature and scope of the development. If this stage is not performed well, it is unlikely that the project will be successful in meeting the business's needs. The key project controls needed here are an understanding of the business environment and making sure that all necessary controls are incorporated into the project. Any deficiencies should be reported and a recommendation should be made to fix them.

The initiation stage should include a cohesive plan that encompasses the following areas:

- Study analyzing the business needs in measurable goals.
- Review of the current operations.
- Conceptual design of the operation of the final product.
- Equipment requirement.
- Financial analysis of the costs and benefits including a budget.
- Select stake holders, including users, and support personnel for the project.
- Project charter including costs, tasks, deliverables, and schedule.

INTERVIEW

An interview is a conversation between two or more people where questions are asked by the interviewer to elicit facts or statements from the interviewee. Interviews are a standard part of qualitative research. They are also used in journalism and media reporting (see Interview (journalism)) and in various employment-related contexts.

The qualitative research interview seeks to describe and the meanings of central themes in the life world of the subjects. The main task in interviewing is to understand the meaning of what the interviewees say. Interviewing, when considered as a method for conducting qualitative research, is a technique used to understand the experiences of others.

Meaning of interview: The word interview comes from Latin and middle French words meaning to “see between” or “see each other”. Generally, interview means a private meeting between people when questions are asked and answered. The person who answers the questions of an interview is called in interviewer. The person who asks the questions of our interview is called an interviewer. It suggests a meeting between two persons for the purpose of getting a view of each other or for knowing each other. When we normally think of an interview, we think a setting in which an employer tries to size up an applicant for a job.

Types of interviews: There are many types of interviews that an organization can arrange. It depends on the objectives of taking the interview. Some important types of interviews are stated below:

1. Personal interviews: Personal interviews include:

- Selection of the employees
- Promotion of the employees
- Retirement and resignation of the employees

Of course, this type of interview is designed to obtain information through discussion and observation about how well the interviewer will perform on the job.

2. Evaluation interviews: The interviews which take place annually to review the progress of the interviewee are called the evaluation interviews. Naturally, it is occurring between superiors and subordinates. The main objective of this interview is to find out the strengths and weaknesses of the employees.

3. Persuasive interviews: This type of interview is designed to sell someone a product or an idea. When a sales representative talk with a target buyer, persuasion takes the form of convincing the target that the product or idea meets a need.

4. Structured interviews: Structured interviews tend to follow formal procedures; the interviewer follows a predetermined agenda or questions.

5. Unstructured interviews: When the interview does not follow the formal rules or procedures. It is called an unstructured interview. The discussion will probably be free flowing and may shift rapidly form on subject to another depending on the interests of the interviewee and the interviewer.

6. Counseling interviews: This may be held to find out what has been troubling the workers and why someone has not been working.

7. Disciplinary interviews: Disciplinary interviews are occurring when an employee has been accused of breaching the organization’s rules and procedures.

8. Stress interviews: It is designed to place the interviewee in a stress situation in order to observe the interviewees reaction.

9. Public interviews: These include political parties' radio-television and newspaper.

10. Informal or conversational interview: In the conversational interview, no predetermined questions are asked, in order to remain as open and adaptable a possible to the interviewee's nature and priorities; during the interview the interviewer "goes with the flow".

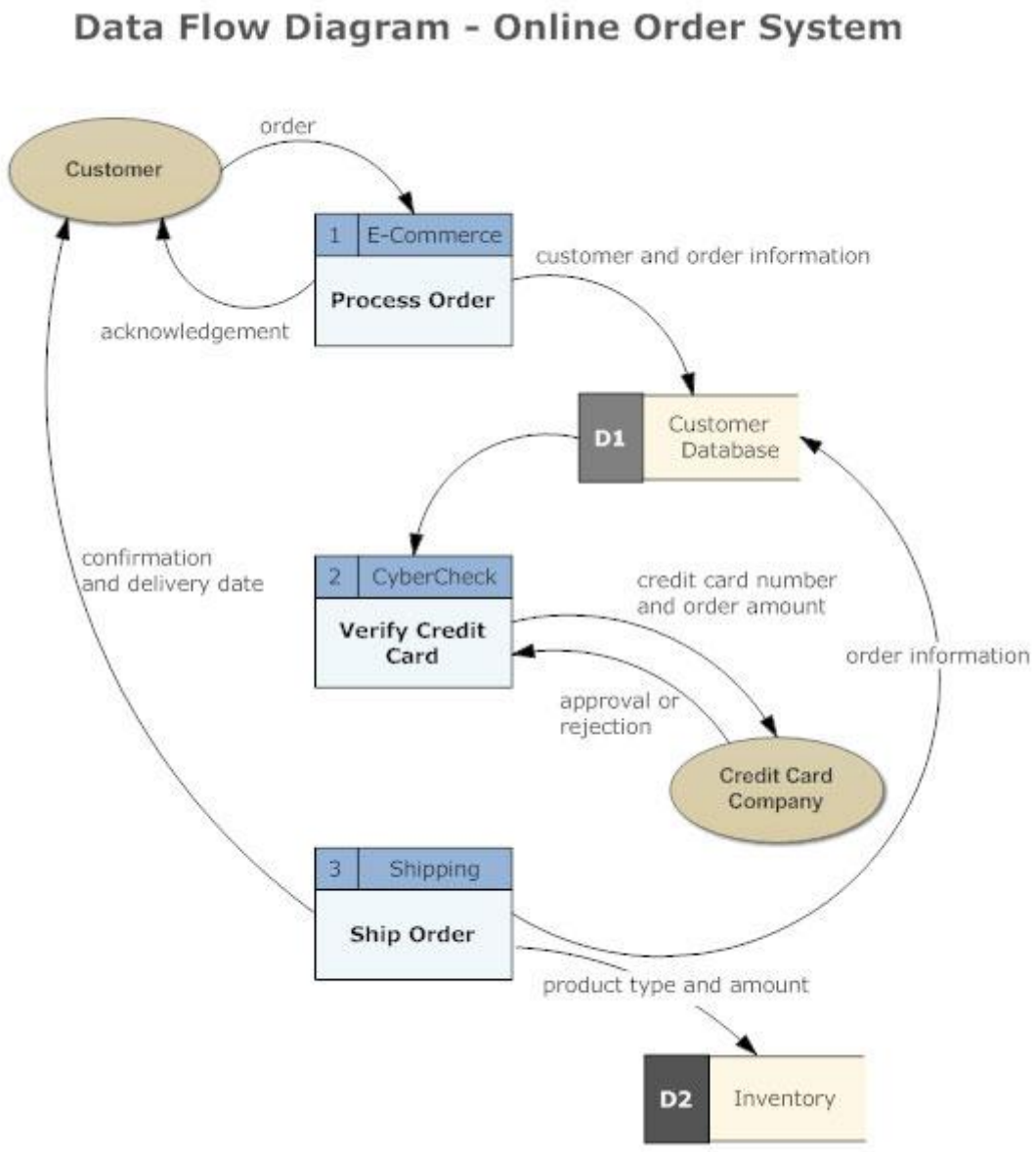
11. General interview guide approach: The guide approach is intended to ensure that the same general areas of information are collected from each interviewee this provides more focus than the conversational approach but still allows a degree of freedom and adaptability in getting the information from the interviewee.

12. Standardized or open-ended interview: Here the same open-ended questions are asked to all interviewees; this approach facilitates faster interviews faster interviews that can be more easily analyzed and compared.

13. Closed or fixed-response interview: It is an interview where all interviewers ask the same questions and asked to choose answers from among the same set of alternatives. This formal is useful for those not practiced in interviewing.

Data Flow Diagram

What is a data flow diagram? A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.



History of Data Flow Diagrams

Data flow diagrams became popular in the 1970s in software development. They were first described in a classic text about *Structured Design* written by Larry Constantine and Ed Yourdon. Yourdon & Coad's Object Oriented Analysis and Design (OOA/OOD) was a way of visualizing software systems before UML diagrams.

Data Flow Diagrams Notations

There are essentially two different types of notations for data flow diagrams (Yourdon & Coad or Gane&Sarson) defining different visual representations for processes, data stores, data flow and external entities.

Yourdon and Coad type data flow diagrams are usually used for system analysis and design, while Gane and Sarson type DFDs are more common for visualizing information systems.

Visually, the biggest difference between the two ways of drawing data flow diagrams is how processes look. In the Yourdon and Coad way, processes are depicted as circles, while in the Gane and Sarson diagram the processes are squares with rounded corners.

Process Notations. A process transforms incoming data flow into outgoing data flow.

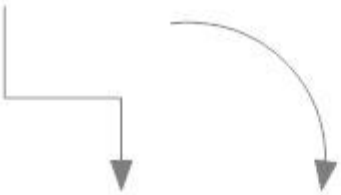
Gane & Sarson

Data store Notations. Data stores are repositories of data in the system. They are sometimes also referred to as files.

Yourdon & Coad

Gane & Sarson

Dataflow Notations. Data flows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.



Data flow

External Entity Notations. External entities are objects outside the system, with which the system communicates. External entities are sources and destinations of the system's inputs and outputs.

External Entity

External Entity

Context Diagrams and DFD Layers and Levels

Context Diagram. A context diagram is a top level (also known as "Level 0") data flow diagram. It only contains one process node ("Process 0") that generalizes the function of the entire system in relationship to external entities.

DFD Layers. Draw data flow diagrams can be made in several nested layers. A single process node on a high level diagram can be expanded to show a more detailed data flow diagram. Draw the context diagram first, followed by various layers of data flow diagrams.

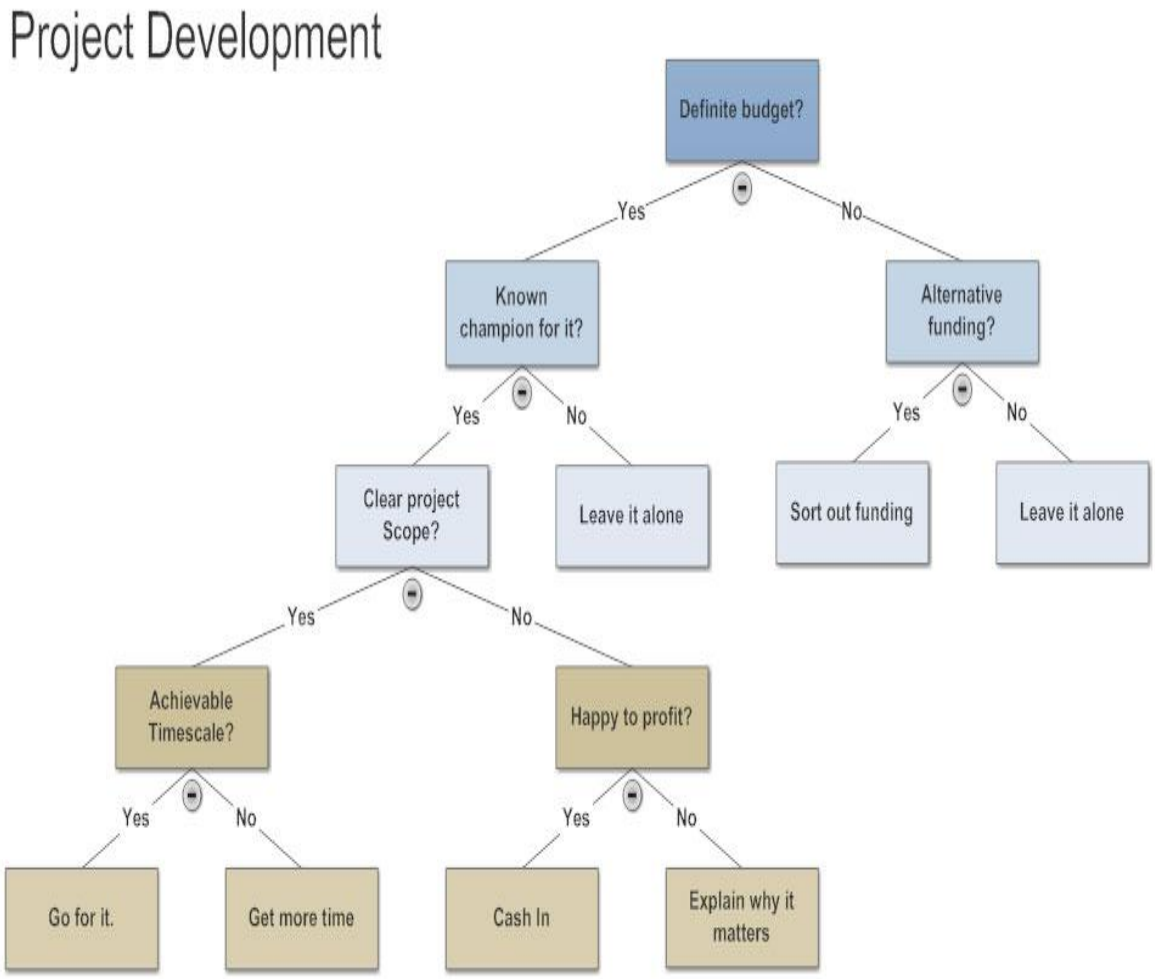
DFD Levels. The first level DFD shows the main processes within the system. Each of these processes can be broken into further processes until you reach pseudo code.

Decision Tree

A decision tree is a flowchart-like structure that shows the various outcomes from a series of decisions. It can be used as a decision-making tool, for research analysis, or for planning strategy. A primary advantage for using a decision tree is that it is easy to follow and understand.

Structure of a Decision Tree

Decision trees have three main parts: a root node, leaf nodes and branches. The root node is the starting point of the tree, and both root and leaf nodes contain questions or criteria to be answered. Branches are arrows connecting nodes, showing the flow from question to answer. Each node typically has two or more nodes extending from it. For example, if the question in the first node requires a "yes" or "no" answer, there will be one leaf node for a "yes" response, and another node for "no."



Decision Tree Uses

A decision tree can be used in either a predictive manner or a descriptive manner. In either instance they are constructed the same way and are always used to visualize all possible outcomes and decision points that occur chronologically. Decision trees are most commonly used in the financial world for areas such as loan approval, portfolio management, and spending. A decision tree can also be helpful when examining the viability of a new product or defining a new market for an existing product.

Seven Tips for Creating a Decision Tree

Here are some best practice tips for creating a decision tree diagram:

- **Start the tree.** Draw a rectangle near the left edge of the page to represent the first node. In this rectangle, write the first question, main idea, or criterion that will lead to a decision.
- **Add branches.** For every possible alternative draw a separate line that begins at the node and moves away toward the right of the page. Using a loan approval process as an example, the first node may have been "Income", and the associated branches might be <$50K, $51K - $100K, >$101K.
- **Add leaves.** The bulk of the decision tree will be leaf nodes. At the end of each branch add a leaf node. Fill each of these leaf nodes with another question or criterion.
- **Add more branches.** Repeat the process of adding a branch for each possible alternative leading from a leaf. Label each branch just as before.
- **Complete the decision tree.** Continue adding leaves and branches until every question or criterion has been resolved and an outcome has been reached.
- **Terminate a branch.** Continue adding leaves and branches until every question or criterion has been resolved and an outcome has been reached.
- **Verify accuracy.** Consult with all stakeholders to verify accuracy.

Data Dictionary

A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them. A first step in analyzing a system of objects with which users interact is to identify each object and its relationship to other objects. This process is called data modeling and results in a picture of object relationships. After each data object or item is given a descriptive name, its relationship is described (or it becomes part of some structure that implicitly describes relationship), the type of data (such as text or image or binary value) is described, possible predefined values are listed, and a brief textual description is provided. This collection can be organized for reference into a book called a data dictionary.

When developing programs that use the data model, a data dictionary can be consulted to understand where a data item fits in the structure, what values it may contain, and basically what the data item means in real-world terms. For example, a bank or group of banks could model the data objects involved in consumer banking. They could then provide a data dictionary for a bank's programmers. The data dictionary would describe each of the data items in its data model for consumer banking (for example, "Account holder" and ""Available credit").

A data dictionary, or Metadata Repository, as defined in the IBM Dictionary of Computing, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format." The term can have one of several closely related meanings pertaining to databases and database management systems (DBMS):

- A document describing a database or collection of databases
- An integral component of a DBMS that is required to determine its structure
- A piece of middleware that extends or supplants the native data dictionary of a DBMS

Database about a database. A data dictionary defines the structure of the database itself (not that of the data held in the database) and is used in control and maintenance of large databases. Among other items of information, it records

(1) what data is stored,

(2) name, description, and characteristics of each data element,
(3) types of relationships between data elements,

(4) access rights and frequency of access.

Also called system dictionary when used in the context of a system design.

Structure Chart

A Structure Chart (SC) in software engineering and organizational theory, is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name. The tree structure visualizes the relationships between modules.

A structure chart is a top-down modular design tool, constructed of squares representing the different modules in thesystem, and lines that connect them. The lines represent the connection and or ownership between activities and subactivities as they are used in organization charts.

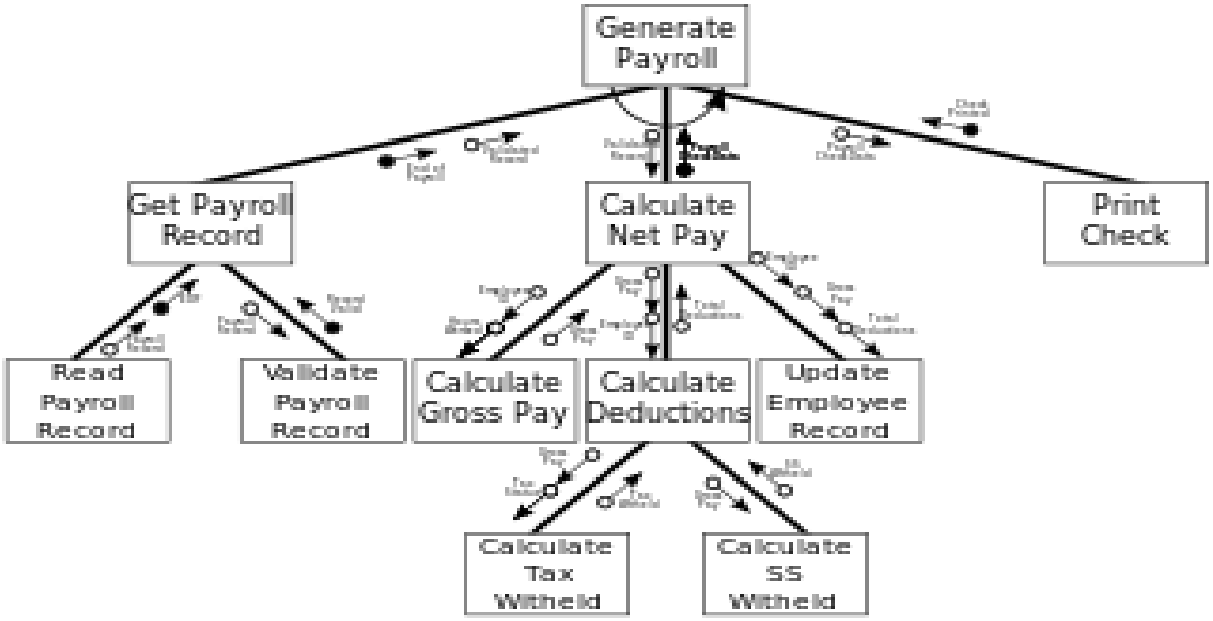
In structured analysis structure charts, according to Wolber (2009), "are used to specify the high-level design, or architecture, of a computer program. As a design tool, they aid the programmer in dividing and conquering a large software problem, that is, recursively breaking a problem down into parts that are small enough to be understood by a human brain. The process is called top-down design, or functional decomposition. Programmers use a structure chart to build

a program in a manner similar to how an architect uses a blueprint to build a house. In the design stage, the chart is drawn and used as a way for the client and the various software designers to communicate. During the actual building of the program (implementation), the chart is continually referred to as "the master-plan".

A structure chart depicts

- the size and complexity of the system, and
- number of readily identifiable functions and modules within each function and
- whether each identifiable function is a manageable entity or should be broken down into smaller components.

A structure chart is also used to diagram associated elements that comprise a run stream or thread. It is often developed as a hierarchical diagram, but other representations are allowable. The representation must describe the breakdown of the configuration system into subsystems and the lowest manageable level. An accurate and complete structure chart is the key to the determination of the configuration items, and a visual representation of the configuration system and the internal interfaces among its CIs). During the configuration control process, the structure chart is used to identify CIs and their associated artifacts that a proposed change may impact.



Structured English

Structured English is the use of the English language with the syntax of structured programming to communicate the design of a computer program to non-technical users by breaking it down into logical steps using straightforward English words. Structured English aims to get the benefits of both the programming logic and natural language: program logic helps to attain precision, whilst natural language helps with the familiarity of the spoken word.

It is the basis of some programming languages such as SQL (Structured Query Language) "for use by people who have need for interaction with a large database but who are not trained programmers".

Elements

Structured English is a limited form "pseudocode" and consists of the following elements:

1. Operation statements written as English phrases executed from the top down
2. Conditional blocks indicated by keywords such as IF, THEN, and ELSE
3. Repetition blocks indicated by keywords such as DO, WHILE, and UNTIL

The following guidelines are used when writing Structured English

1. All logic should be expressed in operational, conditional, and repetition blocks
2. Statements should be clear and unambiguous
3. Logical blocks should be indented to show relationship and hierarchy
4. Use one line per logical element, or indent the continuation line
5. Keywords should be capitalized
6. Group blocks of statements together, with a capitalized name that describes their function and end with an EXIT.
7. Underline words or phrases defined in a data dictionary
8. Mark comment lines with an asterisk

Example of Structured English

A bank will grant loan under the following conditions

1. If a customer has an account with the bank and had no loan outstanding, loan will be granted.
2. If a customer has an account with the bank but some amount is outstanding from previous loans then loan will be granted if special approval is given.
3. Reject all loan applications in all other cases.

```
APPROVE LOAN
IF customer has a Bank Account THEN
  IF Customer has no dues from previous account THEN
     Allow loan facility
  ELSE
     IF Management Approval is obtained THEN
        Allow loan facility
     ELSE
        Reject
```

```
      ENDIF
   ENDIF
ELSE
   Reject
ENDIF
EXIT
```

Definition - What does *Structured English* mean

Structured English is a narrative form of English written as a series of blocks that use indentation and capitalization to represent a hierarchical structure of logic specifications. This method does not show any decisions or rules, but it states the rules and is used when an individual or an organization is trying to overcome the problems of an ambiguous language by stating the actions and conditions used when making decisions and formulating procedures.

Structured English is based on structured logic; it is used when process logic involves formulas or iteration, or when structured decisions are not too complex. Structured English is used to express all logic in terms of sequential structures, decision structures, iterations and case structures. This modified form of English is used to specify the logic of information processes by using a subset of English vocabulary to express process procedures.

Structured English

Structured English derives from structured programming and its use of logical construction and imperative statements. This process is designed to carry out instructions for actions by creating decision statements that use structured programming terms, such as "IF", "ELSE" and "THEN".

Structure statements are developed and defined by using the following types of statements:

- Sequence Structure: This is the single step or action included in the sequence process; it does not depend on the existence of other conditions. If the sequence structure does encounter a condition, it is taken into consideration.
- Decision Structure: This occurs when two or more actions rely on the value of a specific condition. The condition is expanded and the necessary decisions are made.
- Iteration/Repetition Structure: Certain conditions will only occur after specific conditions are executed. Iterative instructions help an analyst describe these specific cases.