Week 2: SAD Fundamentals

Basics

Definition of Terms:

1. SYSTEM:

- A group of interrelated components working together toward a common goal by accepting inputs and producing outputs in an organized transformation process.
- A series of interrelated elements that performs some activity, function or operation.
- A collection of men, machines and methods organized to accomplish a set of specific functions.

2. ANALYSIS:

- Used to gain an understanding of an existing system and what is required of it. If there is no existing system, then analysis defines only the requirements.

3. DESIGN:

- Proposes the new system that meets system's requirements. This new system may be built afresh or by changing the existing system. Once the design is approved, the system is built.

4. SYSTEMS FEASIBILITY:

- A test of a system's proposal based on several criterion. These are:
  - Technical Feasibility: Hardware, Software, People ware
  - Operational Feasibility: Processes, Procedures and Logic
  - Economic Feasibility: Benefits should outweigh the costs

5. SYSTEMS ANALYST:

- Applies to the persons who investigate, analyze, design install and evaluate information systems.
- Does the actual work of gathering the necessary data and developing plans for new systems.
- An agent of change where some people may not want change.

6. GOAL OF ANAYLSYS AND DESIGN:

- To build Computer Based Information Systems (CBIS) that shall meet organizational goals and objectives.

7. GOAL OF SYSTEMS ANALYSIS AND DESIGN

- Is not simply to put in a computer but to improve the system as a whole.
- Analysts must then study the system to see how such goals can be met. This does not always mean putting in a computer system. It can sometimes simply require changes to existing procedures, rearranging workflows or making forms more understandable

The Systems Analyst

Requirements of a Good Systems Analyst:

1. Communication Skills

   - Oral
   - Written
   - Interpersonal

2. Technical Skills

   - Fact Gathering
   - Identification of Information Needs
   - Feasibility Analysis
   - Equipment Evaluation
   - System Design

3. Analytical Skills

   - Systematic problem solving

4. Creative and Imaginative

   - Total Quality Management

5. Getting Things Done

   - Value of Time
   - Respects Deadlines

Systems Analyst vs. Programmer

The systems analyst and the programmer both analyze and make programs. The only difference is in the frequency with which both perform the tasks.

Tasks frequently done by a Systems Analysts:

1. Defines requirements

2. Prepares functional specifications

3. Prepares system specifications

4. Prepares systems' modelling and scheduling frameworks

5. Designs forms and reports

6. Designs data items

7. Defines data organization

8. Defines system calculations

Tasks frequently done by a Programmer:

1. Translate detailed systems' model frameworks into programs

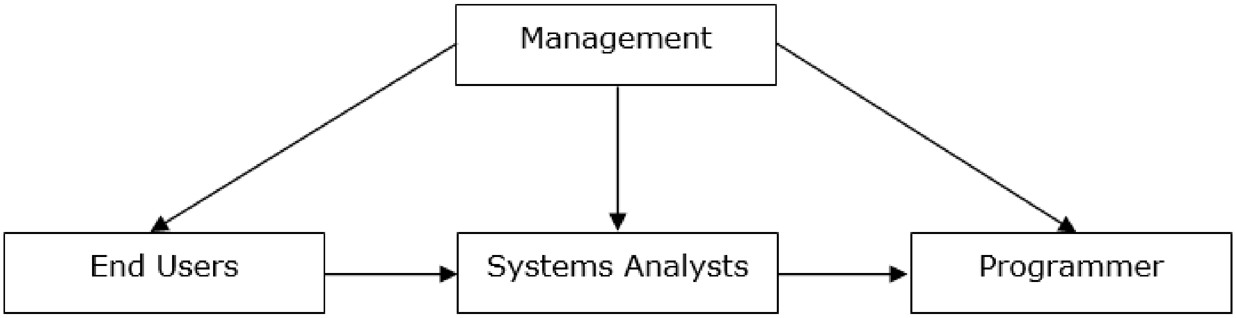
2. Maintains programs

3. Debugs or tests

4. Prepares test data

5. Prepares operator instructions

Organization Relational Block Diagram:



Week 3: Information Requirement Analysis

1. Interviewing

Definition:

This is the most common method of gathering information from users. A continuous process the systems analyst uses to build up a model of the system and to gain understanding of any system problems.

Preparing for the Interview:

- Begin by defining the purpose of the interview
  - Develop a picture of the existing system
  - Identify questions and missing pieces
  - Unknown factors represent initial outline of the interview objectives
- Select the person or group to be interviewed
  - Refer to the formal organizational chart or a report distribution list
  - Refer or interview the manager who is responsible for a given objective and who can tell you who can give the specific answer for your questions
- Do your homework
  - Know the topic
  - Know the basic functions of each employee
  - Be familiar with the relevant documents or procedures that the employee uses
- Prepare specific questions aimed at the individual or group you are planning to interview
  - Develop a written list of questions and follow-up questions in case the interview begins straying from the key point
- Schedule the interview
  - Be willing to arrange your schedule and to travel to the person's office
  - Schedule the interview at the person's convenience

Interview Checklist:

- Opening
  - Establish rapport
  - Identify yourself, the topic you plan to discuss and the purpose of the interview
  - Tell the individual why he was chosen for the interview
  - If appropriate, identify the manager who have authority in approving your requests
  - Avoid wasting time; get to the point

- Body
    - Consider asking the subject how his job relates to the project or how a particular system work
    - Ask follow-up questions to help focus the interview on a particular points of concern
    - Offer brief summary of what you have understood
    - Listen to the answers. Be flexible; don't focus your attention on your next question but listen intently to what the individual is saying. Stick to the subject.
    - Delete questions that seem unimportant or based on earlier responses cannot or will not be answered. Bypass questions that were already answered
    - Avoid attacks. Don't cross-examine the interviewee; remembering that he is the expert. Don't question the credibility of the person by saying *"you don't know what you're talking about…"*
    - If it seems that the interviewee is useless, you might close it earlier but always act professionally in spite of your disappointment.
    - Take down notes – jot down key points but not record every word that you might miss the meaning. You might also consider taping the interview but get permission first from the interviewee.
- Closing
    - Pay attention to the time, if the interview takes longer than expected, ask permission to continue and offer to reschedule. Thank the interviewee for his cooperation.
    - Maintain a good atmosphere with the interviewee.
    - Share summary with the interviewee because it provides an excellent opportunity for correcting misunderstandings.

Question Types:

- Open-Ended: Allow the respondent open options for responding

    -
    - Examples of open ended questions:
        - What's your opinion of the present manual system?
        - How do you view the goals of this department?
        - How does this form relate to the work you do?
        - What are some of the problems you experience in receiving information on time?
        - What are some of the common errors made in data entry?
        - Describe the most frustrating computer system you've worked with?
    - Benefits of an open ended questions:
        - Putting the interviewee at ease

        - Allowing the interviewer to pick up on the interviewee’s vocabulary, which reflects his or her education, values, attitudes and beliefs
        - Providing richness of detail
        - Revealing avenues of further questioning that may have gone untapped
        - Making more interesting of the interviewee
        - Allowing more spontaneity
        - Making phrasing easier for the interviewer
        - Using them in a pinch if the interviewer is caught unprepared
    - Drawbacks of an open ended questions:
        - Asking questions that may result in too much irrelevant detail
        - Possibly losing control of the interview
        - Allowing responses that may take too much time for the amount of useful information gained
        - Potentially seeming that the interviewer is unprepared
        - Possibly giving the impression that the interviewer is on a “Fishing expedition” with no real objective for the interview

- Closed Questions: Limit the options the respondent has for responding.
    - Examples closed questions:
        - How many reports do you generate in a month?
        - How long have you worked for the company?
        - Which of the following sources of information is most valuable to you:
            - Filled out customer complaint forms?
            - Face to face interaction with customer?
            - Returned merchandise itself?
        - Who receives this output?
    - Benefits of closed questions:
        - Saving Time
        - Easily comparing interviews
        - Getting to the point
        - Keeping control over the interview
        - Covering lots of ground quickly
        - Getting to relevant data
    - Drawbacks of closed questions:
        - Being boring for the interviewee
        - Failing to obtain rich detail
        - Missing main ideas because of the above

    - Failing to build rapport
  - Bipolar questions
    - Special kind of closed questions
  - Examples of bipolar questions
    - Do you use a computer?
    - Do you agree or disagree that automating teller functions would be worthwhile?
    - Do you want to receive a computer printout of your account status every month?
    - Does your accounting department offer automatic electronic transfer of payroll check for regular employees?
    - Is this form complete?

2. Sampling and Investigating Hard Data

- Sampling:

The process of systematically selecting representative elements of a population. When these selected elements are examined closely, it is assumed that the analysis will reveal useful information about the population as a whole.

-
  - The need for sampling
    - Containing costs
    - Speeding up the data gathering
    - Improving effectiveness
    - Reducing bias
  - Kinds of information sought in Investigation
    - Facts and figures
    - Financial information
    - Organizational context
    - Document types and problems

Note: Hard data reveal where the organization has been and where its members believe it is going. The analyst needs to examine both quantitative and qualitative hard data in order in order to piece together an accurate picture.

3. Questionnaire

- Kinds of information sought
  - Attitudes

    - Beliefs
    - Behavior
    - Characteristics
- Guidelines using questionnaires
    - The people you need to question are widely dispersed. Different branches of the same corporation.
    - There is a large number of people involved in the system project and it is meaningful to know what proportion of a given group approve or disapprove of a particular feature of the proposed system
    - You are doing an exploratory study and want to gauge overall opinion before the system project is given any specific direction
    - You desire to do problem sensing so that any problems with the current system are identified and addressed in follow-up interviews

4. Observing Decision Maker's Behavior and Office Environment

- Kinds of information sought
    - Activities
    - Messages
    - Relationships
    - Influence
- Significance
    - System analysts use observation for many reasons. One reason is to gain information about decision makers and their environment that is unavailable through any other method. Observing also helps confirm what has been found through interviewing and questionnaires. A third reason for observing is to negate or reverse what was found by other methods.

5. Prototyping

- Kinds of information sought
    - User reactions
    - Innovations
    - User suggestions
    - Revision plans
- Kinds of prototype
    - Patched-up prototype
    - None operational prototype
    - First-of-a-series prototype
    - Selected features prototype

Week 4: Systems Development Life Cycle

7 Phases of SDLC

PHASE 1: Identifying Problems, Opportunities and Objectives

During this stage, the systems analyst tries to identify the problem. Failure to identify the problem correctly may result in difficulties in designing a workable system and it may also result in frequent changes and backtracking, which increases the development cost and delays in completion time. A misunderstood problem definition guarantees that the system will fail to solve the problem. As the saying goes *"What is the use of an excellent answer to a wrong question?"*

During this stage, the key question to be answered is *"What is the problem?"*

- Sources of problem definition
    - Usually this question is asked directly to the personnel involved. The user is the first one who recognizes that a problem exists because he is the one who encounters difficulties in the system. The management identifies an area of poor performance within the function of the user. The management and the user may want the analyst to look into it.
    - After the problem has been correctly identified, the first responsibility of the analyst is to prepare a written statement of the scope and objectives of the problem. This is a written memorandum that contains:
        - The problem
        - Rough estimate of the development cost
        - Approximate time of completion
        - Preliminary ideas, solutions
        - The proposal of a feasibility study
    - The function of this statement is for communicating ideas, where:
        - Scope – the extent to which you are going to get
        - Objectives - aims, what the analyst intends to do.

PHASE 2: Determine Information Requirements

The next phase is determining information requirements for the particular users involved. Several tools are used to define information requirements in the business (refer to MODULE 3/4).

In this phase, the analyst is striving to understand what information users need to perform their jobs. This phase serves to fill in the picture that the analyst has of the organization and its objectives.

PHASE 3: Analyzing Systems Needs

At this point in the systems development life cycle, the systems analyst prepares a system proposal that summarizes what has been found, provides cost benefit analysis (CBA) of alternatives, and makes recommendation on what (if anything) should be done.

- Feasibility Study
    - A high level capsule version of the entire process intended to answer a number of questions.
    - A feasibility study is an investigation that ascertains the viability of an undertaking. The undertaking may be a new or proposed system.
    - Objectives:
        - Guide the system analyst, the management and the user in determining the actions they must take on a project in order to bring about its successful operation.
        - Determine if there is a feasible solution
    - The purpose of the study is not to solve the problem but to determine if the problem is worth solving.
    - At this stage, the systems analyst's main concern is to answer the question *"Is there a feasible solution?"* To answer this question, the analyst makes use of his knowledge in advance problem solving techniques. Once the feasibility study is finished and submitted by the analyst to management, the management makes a decision based on the study. Many projects are not continued this early phase of the development cycle because management can already determine if the project is feasible or not.
    - Steps in a Typical Feasibility Study

A. Clarify the scope and objectives:

The analyst must have a clear understanding of what the scope and objectives mean before digging into the system. The analyst must learn certain basics about the application of the investigated system before analyzing it. Gain a sense of application.

B. Study the existing system:

The quickest way to understand any application. The first step is to interview key people. To summarize what we have understood of the system, we need to draw a system flowchart or equivalent. Understand the existing system.

-
    -
        -
        - System flowcharting symbols represent the physical system of a certain information system.

C. Develop a High-Level Model

The objective is not to reproduce the existing physical system but to create a new or improved system that performs the same functions. Therefore, the analyst would summarize the existing system in a way that stresses function rather than physical implementation.

D. Redefine the scope and objectives

This is the time for follow-up interviews with user and management. At this stage, the understanding of the system is much deeper that it was during the problem definition. If the system's scope and objectives is not valid, the analyst should be ready to start over again.

Steps A to C should be viewed as a loop. This should continue until such time that the logical model is accurate.

E. Develop alternative solutions

Is there a feasible solution? To answer this question, the analyst must develop a number of alternative solutions and analyze them.

Alternative solutions, however should still be consider the three feasibilities: Technical, Operational, Economical.

F. Perform Cost Benefit Analysis (CBA)

Developing a system is an investment because funds are committed knowing that in return we expect future benefits. Benefits can either be tangible, intangible of both.

G. Recommend a Course of Action

Given the established scope and objectives, the systems analyst must then recommend the best solution suited for the system

H. Rough out a Development Plan

Development plan based on the recommended course of action. Reasonable time estimates based on the system life cycle are possible.

Develop time frames using Gantt Chart or the like.

I. Write and Present the Feasibility Study

Collect the information and write a formal report which would be presented to management and user.

PHASE 4: Designing The Recommended System

In this phase of the systems development life cycle, the system analyst uses the information collected earlier in order to accomplish the logical design of the information system. The analyst designs accurate data entry procedures so that data going into the information system are correct. The analyst also designs effective input modules to the information system using techniques of good form and screen design.

Part of logical design of the information system is designing the user interface. This interface is what connects the user with the system. Examples of user interfaces include using keyboard to type in questions and answers, use of on-screen menus of possible user commands, use of mouse, and many others.

The design phase also includes designing the files or database that stores much of the data needed by decision makers in the organization. In this phase, the analyst also designs output (either printed or on-screen), along with users, to meet their information needs.

PHASE 5: Developing and Documenting Software

In this phase, the analyst works with the programmers to develop any original software that is needed. Some of the structured techniques for designing and documenting software include HIPO method, flowcharts, Nassi-Shneiderman charts, Warnier-Orr diagrams, UML, DFD and pseudocode. The system analyst communicates to the programmer what needs to be programmed.

During this phase, the analyst also works with the users to develop worthwhile documentation for software, including procedure manuals. Documentation tells users how to use software, and also, what to do if software problem occurs.

PHASE 6: Testing and Maintaining the System

Before the information system can be used, it must be tested. It is much less costly to catch problems before the system is signed over to users. Some of the testing is done completely by programmers alone, some of it by the system analysts in conjunction with the programmers. Series of test to pinpoint problem is run first with sample data and eventually with actual data from the current system.

Maintenance of the system and its documentation just begins in this phase. It is carried out routinely throughout the life cycle of the information system. Much of the programmer’s routing work is composed of maintenance work, and businesses spend a great deal of money on maintenance. Many of the systematic procedures the analyst employs throughout the systems development life cycle can help ensure that maintenance is kept to a minimum.

PHASE 7: Implementation and Evaluating the System

In the last phase of the system development, the analyst helps implement the information system. This involves training the users to use the system. Some training is done by vendors, but overseeing training is the responsibility of the systems analyst. Additionally, the analyst needs to plan for the smooth conversion of the old system to the new one.

Although evaluation is shown as part of the last phase of the systems development life cycle, this is mostly for convenience of discussion. Actually, evaluation takes place during every phase. A key criterion that must be satisfied is whether the intended users are indeed using the system.

The Analysis Process

Data Flow Diagram (DFD)

A. Using Data Flow Diagrams

When systems analysts attempt to understand the information requirements of users, they must be able to conceptualize how data moves through the organization, the processes or transformation that the data undergoes, and what the outputs are. Although interviews and investigation of hard data provide a verbal narrative of the system, a visual depiction can crystallize it in a useful way.

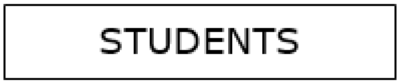
B. Advantages of Data Flow Diagrams

-
    - Freedom from committing to the technical implementation of the system too early
    - Further understanding of the interrelatedness of systems and subsystems
    - Communicating current system knowledge to users through data flow diagrams

C. Convention Used in Data Flow Diagrams

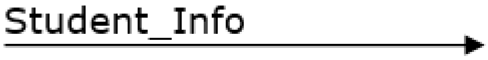
1. Entity

The rectangle is used to depict an external entity (a business, person or machine) that can send or receive data from the system. The external entity is also called a source or destination of data, and is considered to be external to the study. Each external entity is labeled with an appropriate name (noun).



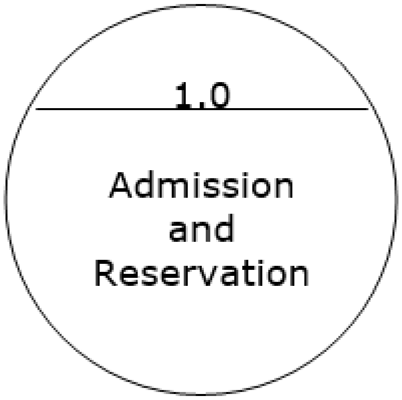
2. Data Flow Lines

The arrow shows movement of data from one point to another, with the head of the arrow pointing toward the data's destination. Data flows occurring simultaneously can be depicted doing just that through the use of parallel arrows. Each arrow is labeled with an appropriate data flow name (noun).



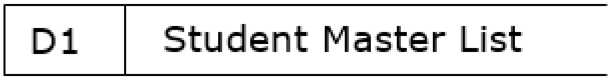
3. Process

The circle is used to show the occurrence of a transforming process. Processes always denote a change in or transformation of data; hence, the data flow leaving a process is always labeled differently than the one entering it. Every process should be labeled accordingly using action verbs (gerund verbs).



4. Data Store

The open-ended rectangle with a vertical line on the left side is used in data flow diagrams, which represents a data store. These are drawn only wide enough to allow lettering between the parallel lines. In data flow diagrams, the type of physical storage (i.e. tape, diskette, hard disk, etc.) is not specified. At this point, the data store symbol is simply showing a repository of data that allows updating and retrieval of data.

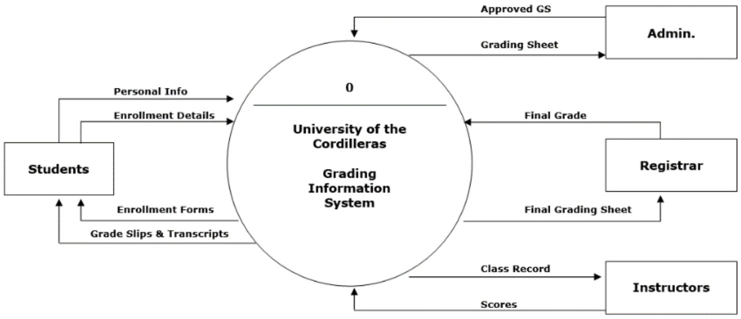


D. Levels of Data Flow Diagrams

1. Context Diagram

Shows the overview of the system. Includes all external entities, inputs and outputs. A big process symbol is used to denote the name of the information system being studied.

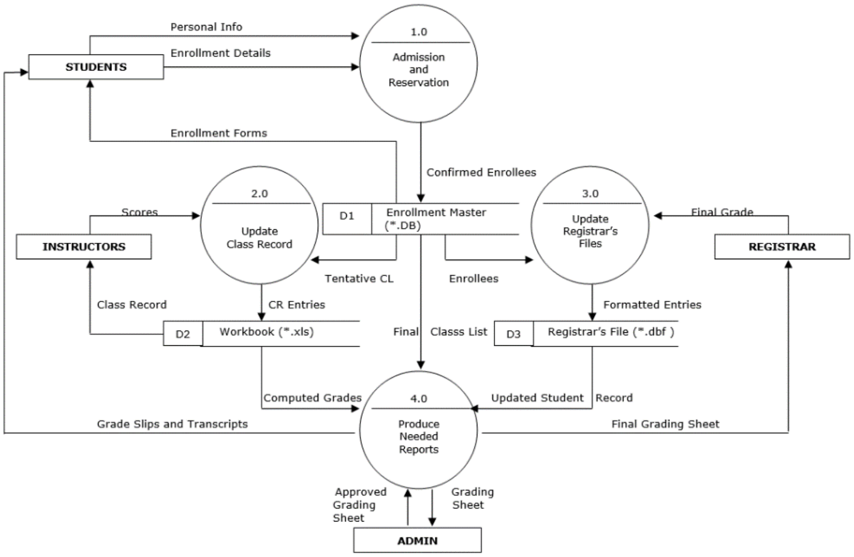
CONTEXT DIAGRAM



2. Top Level Diagram

Shows all the external entities, input and output data flow lines as well as the major processes involved. Data stores are also shown in this diagram. Processes shown are the major processes only, about 4 – 6 are ideal number of processes.

TOP LEVEL DIAGRAM



3. Explosions

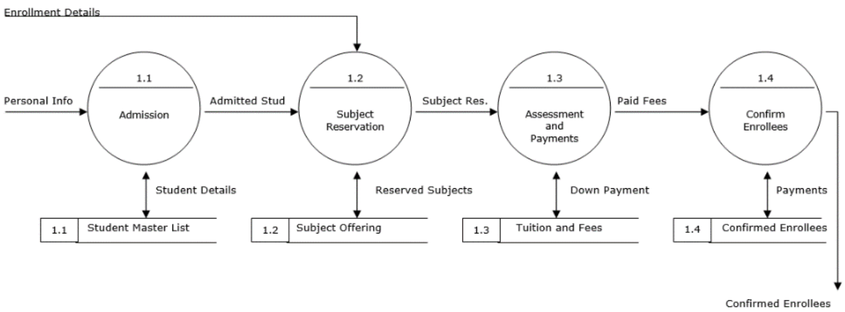
Break down a major process into its detailed sub-processes, including their input and output data flow lines. Data stores are also included if necessary. The entities are no longer shown in this diagram.

Note: All diagrams should be contained in one page as much as possible.

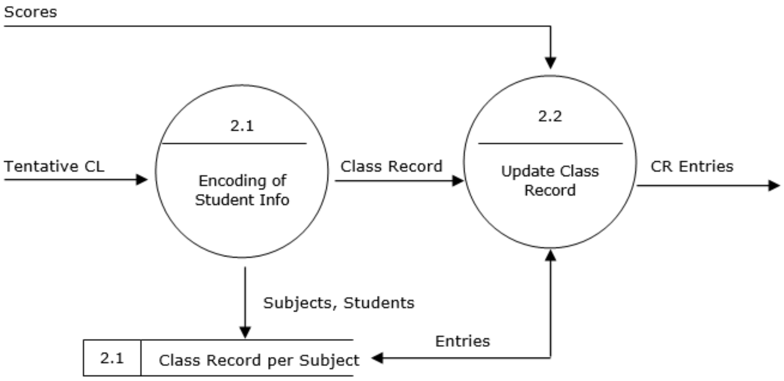
Example:

EXPLOSIONS

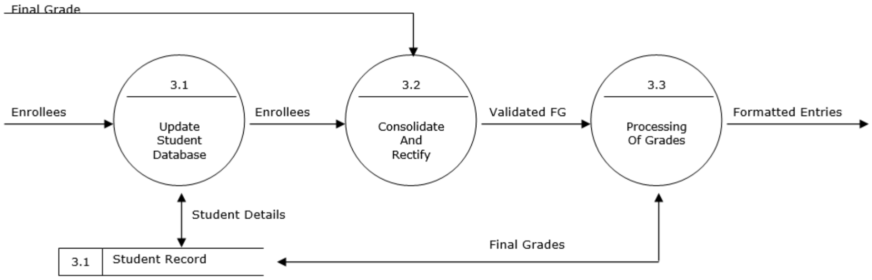
Explosion of Process 1.0 (Admission and Reservation)



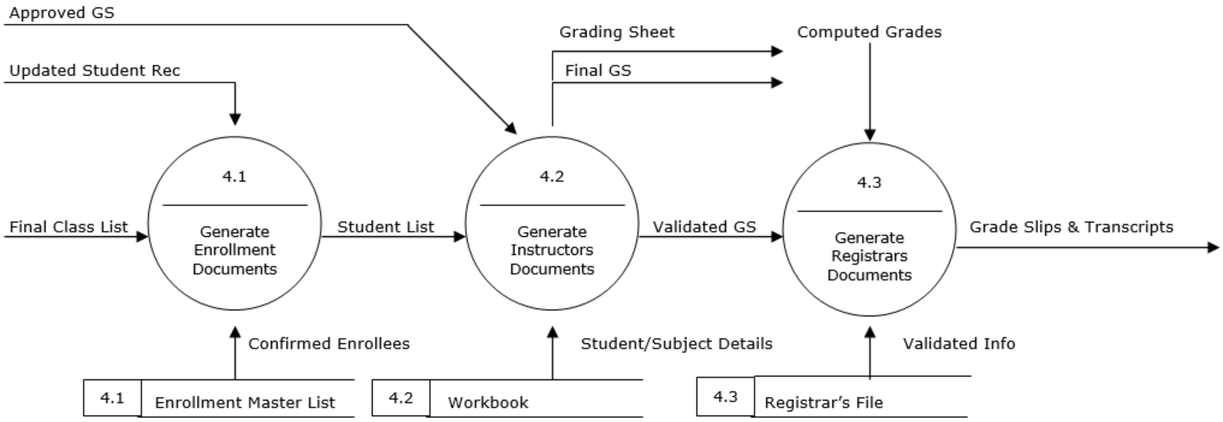
Explosion of Process 2.0 (Update Class Record)



Explosion of Process 3.0 (Update Class Record)

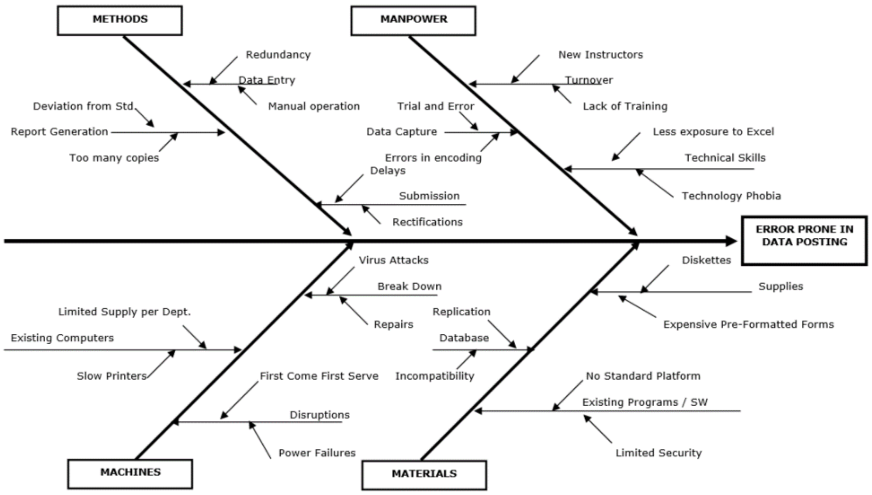


Explosion of Process 4.0 (Report Generation)



ISHIKAWA DIAGRAM

CAUSE AND EFFECT DIAGRAM (ISHIKAWA)


METHODS
MANPOWER
Redundancy
Data Entry
Manual operation
Deviation from Std
Report Generation
Too many copies
Delays
Submission
Rectifications
New Instructors
Turnover
Lack of Training
Trial and Error
Data Capture
Errors in encoding
Less exposure to Excel
Technical Skills
Technology Phobia
ERROR PRONE IN DATA POSTING
Virus Attacks
Break Down
Repairs
Limited Supply per Dept.
Existing Computers
Slow Printers
First Come First Serve
Disruptions
Power Failures
Diskettes
Supplies
Expensive Pre-Formatted Forms
Replication
Database
Incompatibility
No Standard Platform
Existing Programs / SW
Limited Security
MACHINES
MATERIALS