Software Engineering

Narayanan K

Midas Communication Tech Pvt.. Ltd.

Presentation Objectives

- **Define software engineering.**
- **Identify the objectives of s/w engineering.**
- **Understand the role of a s/w engineer vis.a.vis a programmer.**
- **Understand System Development Life Cycle(SDLC).**
- **Models of s/w engineering.**
- **Software requirements definition.**
  - State-Oriented Notations.
    - Decision tree/ table,Transition table, Finite-State machine.
- **Software Design.**
  - Design Notations.
    - System Flow charts Programming logic Flow charts, DFD.
    - Hierarchy, structure charts,HIPO,PDL,Data dictionary
- **Software Implementation issues.**
- **Software verification and validation techniques.**
- **Proposed S/w Engg. and Maintenance model at Midas**
- **Case Study.**

Definitions & Objective

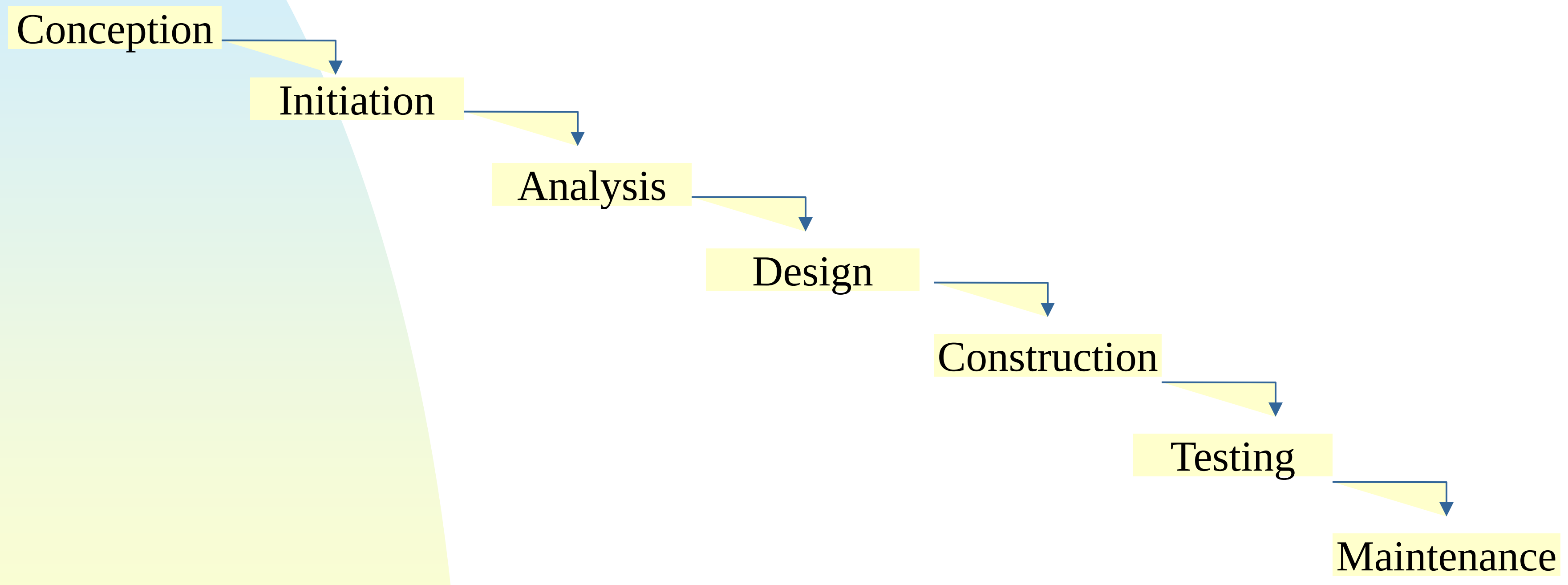
- **Software Engineering(Bauers):**
  - The process of application of sound engineering principle to obtain an economical s/w that is reliable and works efficiently
- **System Engineering:**
  - An arrangement of things/processes that are organized to accomplish a goal.Defines the element in the context of the overall hierarchy of systems.
- **Process:**
  - Is a framework for the tasks that are required to build a high quality software.
- **Aim of software Engineering:**
  - Improve the quality of the software. Quality-efficiency,clarity-readability,reliability and usefulness i.e conform to user needs.
  - Increase productivity and job satisfaction of sw engineers

Software Engineer Vs a Programmer

- **S/w engineering is concerned with development and maintenance of s/w products.**
- **Multiple users, maintainers, developers and limitations on time and cost necessitate a systematic approach to s/w production.**
- **A traditional programmer is concerned with details of implementation,packaging and modifying algorithms and data structures written in a specific language.**
- **A s/w engineer must have the following skills:**
  - Knowledge of one or more programming languages, data structures and algorithms.
  - Familiarity with several design approaches.
  - Ability to build and use a model of the required application.
  - Communication and interpersonal skills
  - Ability to schedule work.

Software Models

- **Waterfall Model or Phased Model**
  - Sequential move from one phase to another- 7 phases.



Waterfall Model

- **Advantages of Waterfall Model**
  - Stages well defined, Helps plan and schedule a project.
- **Disadvantages**
  - Assumes requirements are stable and frozen across project span.
  - Critical assumption of clarity of problem definition and desired solution.
  - Testing is only at the end of SDLC and not throughout SDLC

Software Model

- **Evolutionary Model – Prototyping model**
  - Developing a scaled down version of a system.
  - Whose stages consist of expanding increments of an operational product.
  - The direction of evolution being determined by experience.
- **Prototyping is a working model used for several purposes**
  - Validation of user requirements
  - Performing feasibility study of a complex system.
  - Establishing starting point of further evolution.
- **Advantages of Prototyping**
  - Prototype modification is faster and cheaper than that of the final s/w.
  - Definition of unfamiliar areas and uncertain requirements
- **Disadvantages**
  - Prototyping tools are expensive

Waterfall or Phased Model

- **Conception:**
  - Problem perception , Goal identification ,
  - Estimate Benefits and Identify the scope of the Project
- **Initiation:**
  - Understand user requirements:
  - The requirements specify ,Title of work , Nature of work , Date of request , Date of completion , Job objectives , Input/Output description and Users Sign/Designation.
- **Analysis:**
  - Generates functional specifications which contain:
    - Outputs to be produced, Inputs that are required. Procedures to be adopted. Defines the Audit and Control features to monitor the proper functioning of the system.Acceptance test plan for system acceptance.
    - Structural analysis tools are DFD, Data Dictionary, Structured English, Decision trees and tables. These tools are used to create the functional specification which forms the framework for design and implementation.

Waterfall Model or Phased Model

- **Design:**
  - Functional specifications - used to design the desired system.
- **Design Specifications contain**:
  - User Interfaces
  - Data Structures , Algorithms and Program Structures.
  - Specifies resource requirement and other procedures that are part of the system
- **Construction:**
  - Actual coding phase where individual modules developed are tested.
- **Testing:**
  - Integration and system testing for online response,volume of transactions,stress,recovery of failure and usability.
- **Maintenance:**
  - Involves the adding features to the existing system and getting the system up and running at the customer premises, training the operating staff.

Software Requirements Specification

- **Format of a software requirements specifications:**
  - Product overview and summary
  - Development,operating and maintenance environments
  - Information Description
    - Data Flow & Control Flow
  - Functional Description.
    - Process narration, Restrictions and limitations, Performance requirements, Design constraints.
  - Behavioral Description.
    - System states, Events and Actions.
  - Acceptance criteria
    - Classes of tests, Expected software response.
  - Cross-reference & glossary of terms

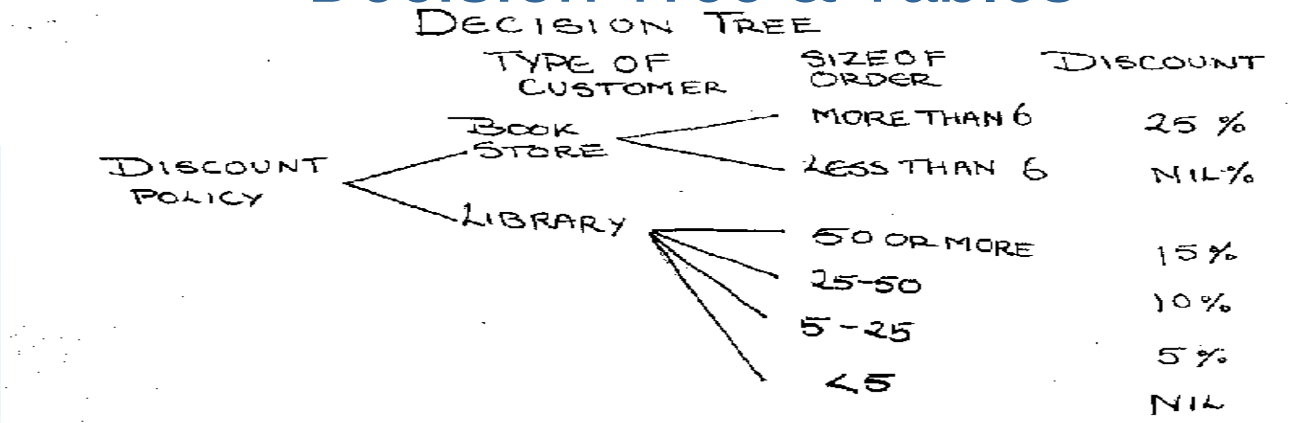
State Oriented Notations

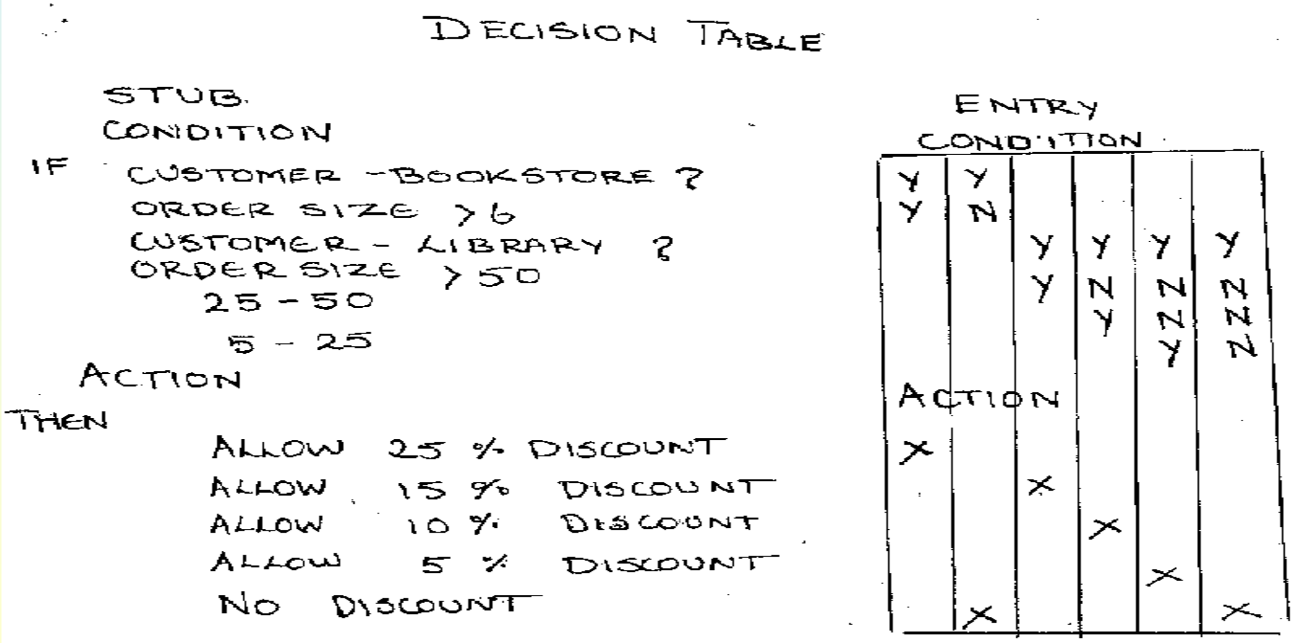
- Transition Tables

Present State	Input	Action	Output	NextState
s0	a			s0
s0	b			s1
s1	a			s1
s1	b			s0

- **f(Si,Cj) = Sk**
- Decision tables/trees
  - Complicated decision logic can be pictorially represented as trees or tables and tools are available to convert them directly into s/w code.

Decision Tree & Tables

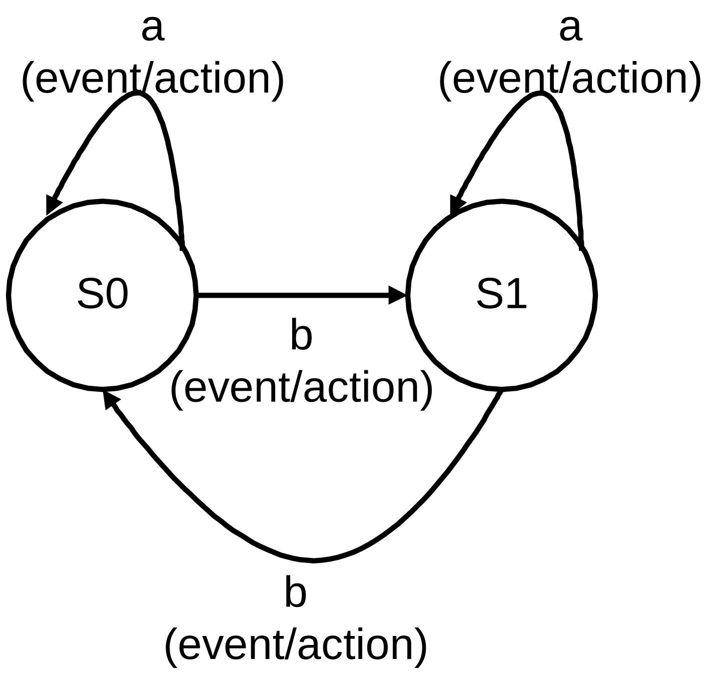



DECISION TABLE

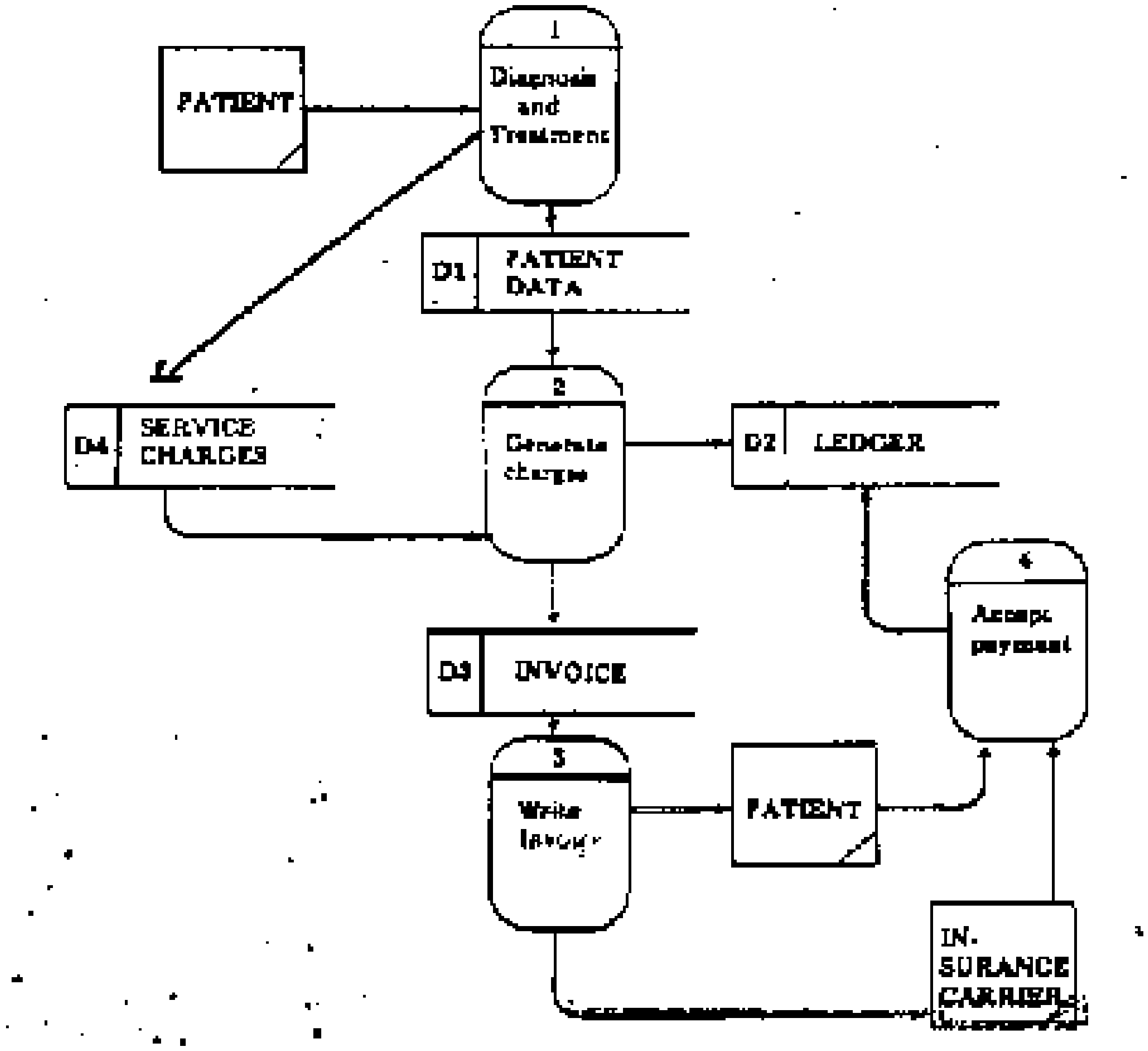
	STUB CONDITION	ENTRY CONDITION					
IF	CUSTOMER - BOOKSTORE ?	Y	Y				
	ORDER SIZE > 6	Y	N				
	CUSTOMER - LIBRARY ?			Y	Y	Y	Y
	ORDER SIZE > 50			Y	N	N	N
	25 - 50				Y	N	N
	5 - 25					Y	N
	ACTION	ACTION					
THEN	ALLOW 25 % DISCOUNT	X					
	ALLOW 15 % DISCOUNT			X			
	ALLOW 10 % DISCOUNT				X		
	ALLOW 5 % DISCOUNT					X	
	NO DISCOUNT		X				X

Finite State Machine

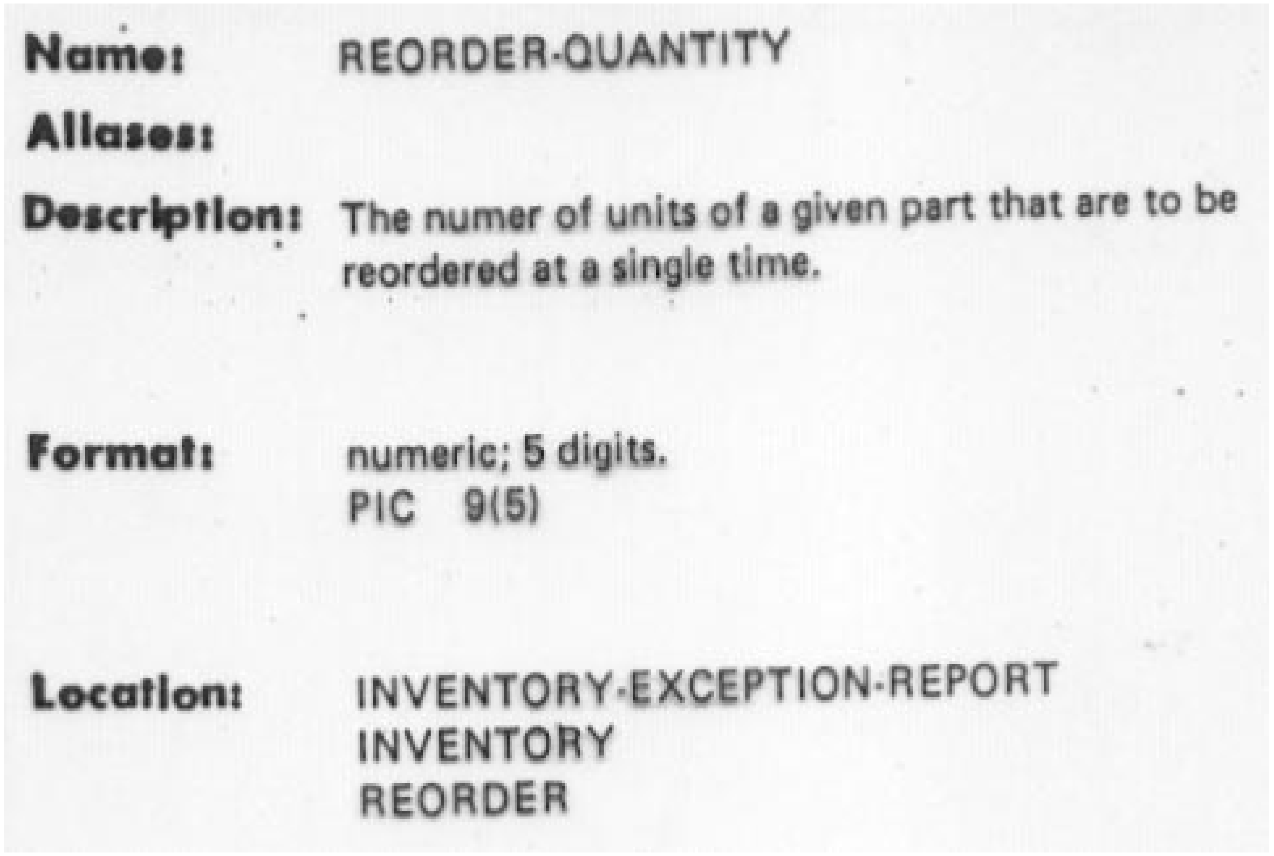
- Indicates the States, Events that trigger State change and the Action that is taken before changing the state in response to the events.



Requirements/Design tool - DFD

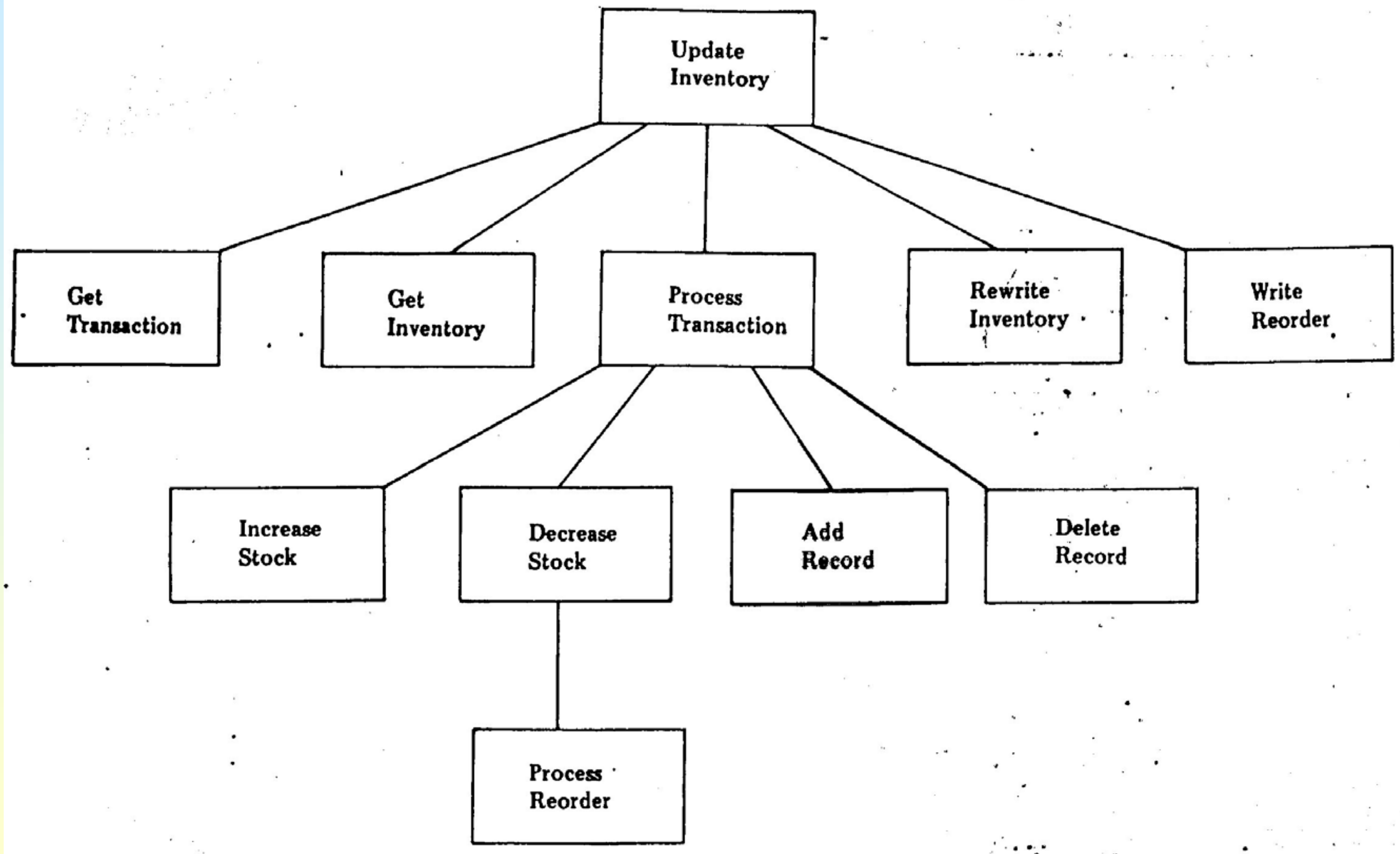


Design tool-Data Dictionary

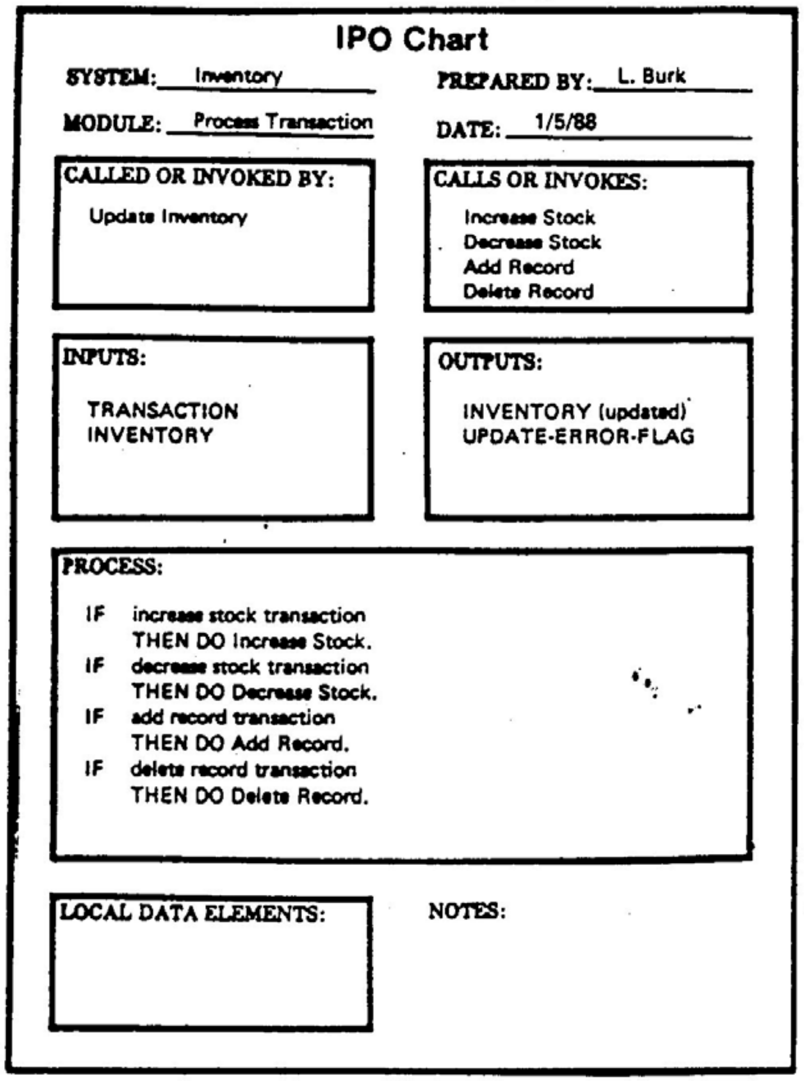


Name:	REORDER-QUANTITY
Aliases:	
Description:	The numer of units of a given part that are to be reordered at a single time.
Format:	numeric; 5 digits. PIC 9(5)
Location:	INVENTORY-EXCEPTION-REPORT INVENTORY REORDER

Design tool-Structured Chart



Design tool- HIPO



IPO Chart

SYSTEM:	Inventory	PREPARED BY:	L. Burk
MODULE:	Process Transaction	DATE:	1/5/88

CALLED OR INVOKED BY:

Update Inventory

CALLS OR INVOKES:

Increase Stock
Decrease Stock
Add Record
Delete Record

INPUTS:

TRANSACTION
INVENTORY

OUTPUTS:

INVENTORY (updated)
UPDATE-ERROR-FLAG

PROCESS:

```
IF   increase stock transaction
     THEN DO Increase Stock.
IF   decrease stock transaction
     THEN DO Decrease Stock.
IF   add record transaction
     THEN DO Add Record.
IF   delete record transaction
     THEN DO Delete Record.
```

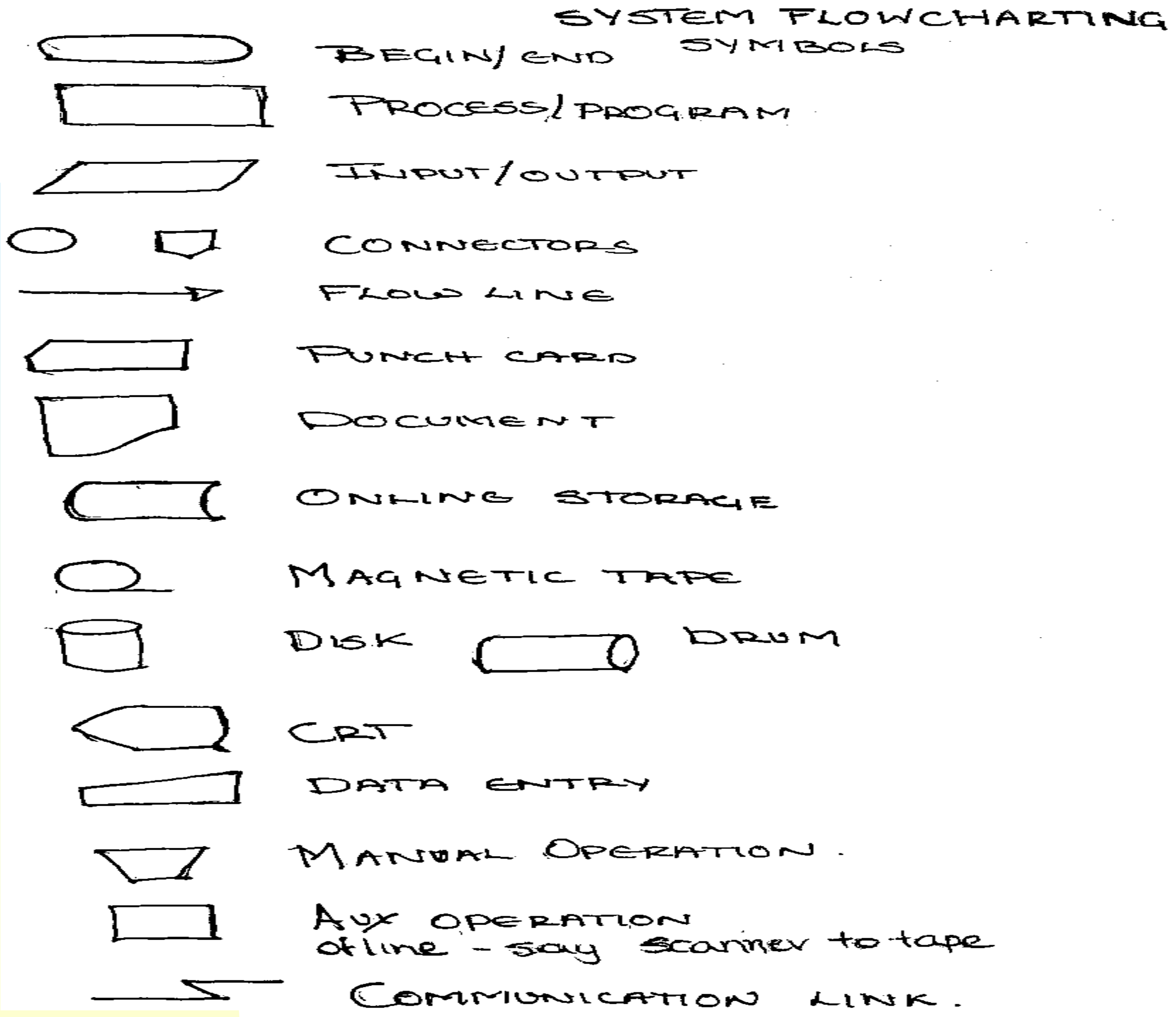
LOCAL DATA ELEMENTS:

NOTES:

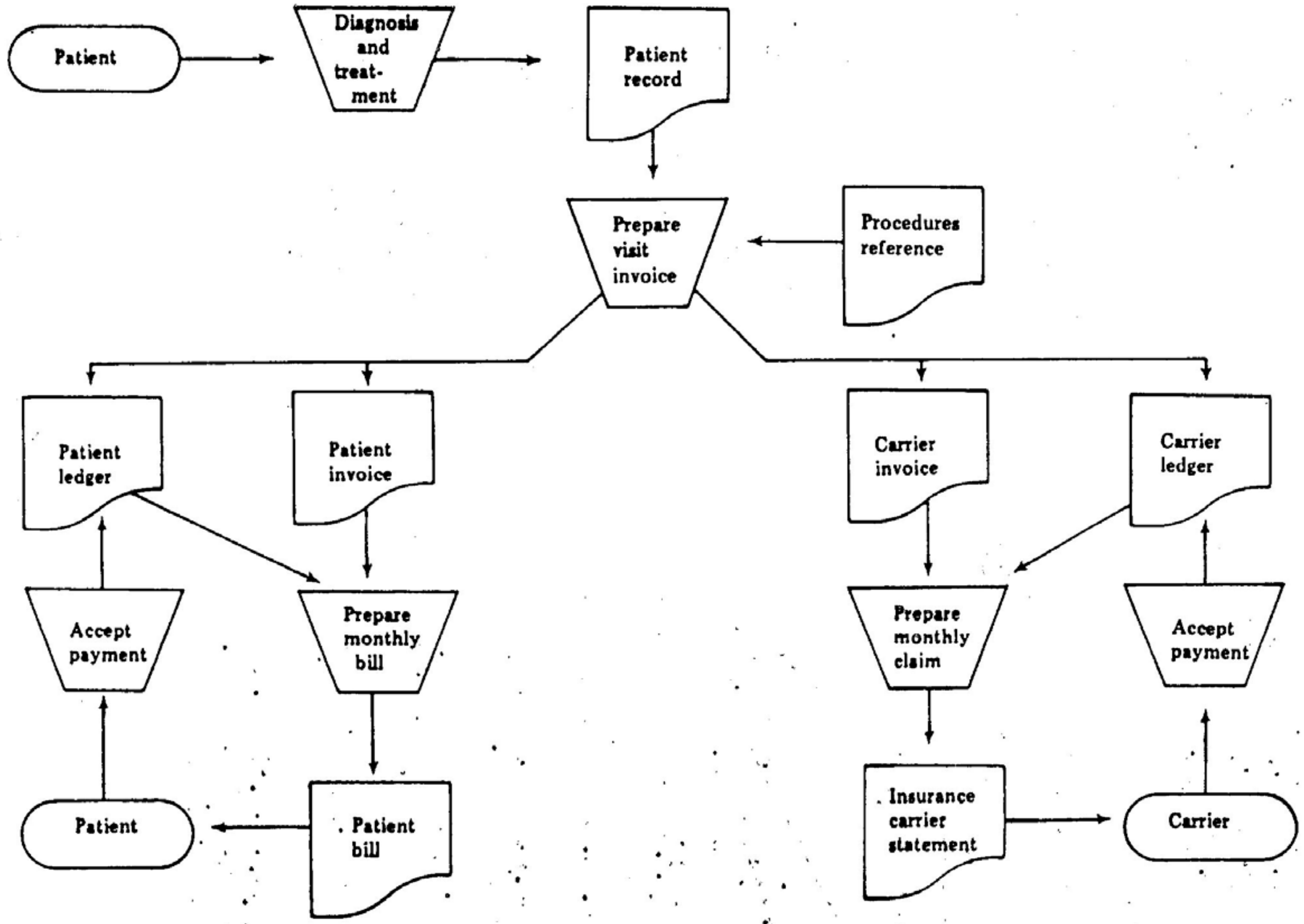
Structured English/ Pseudocode

```
INITIALIZE tables and counters;OPEN files
READ the first record
WHILE there are more records DO
    EXTRACT the first word
    SEARCH the word table
    IF word found THEN
    INCREMENT the extracted word
    ELSE
    INSERT the word in the word table
    ENDIF
ENDWHILE
CLOSE FILE
TERMINATE
```

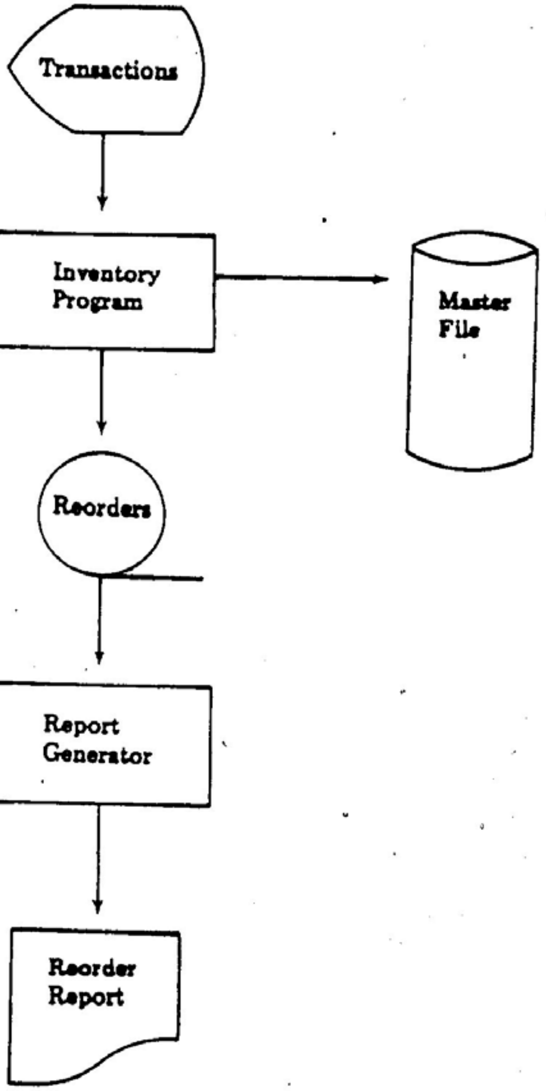
Design Tool - Flow chart



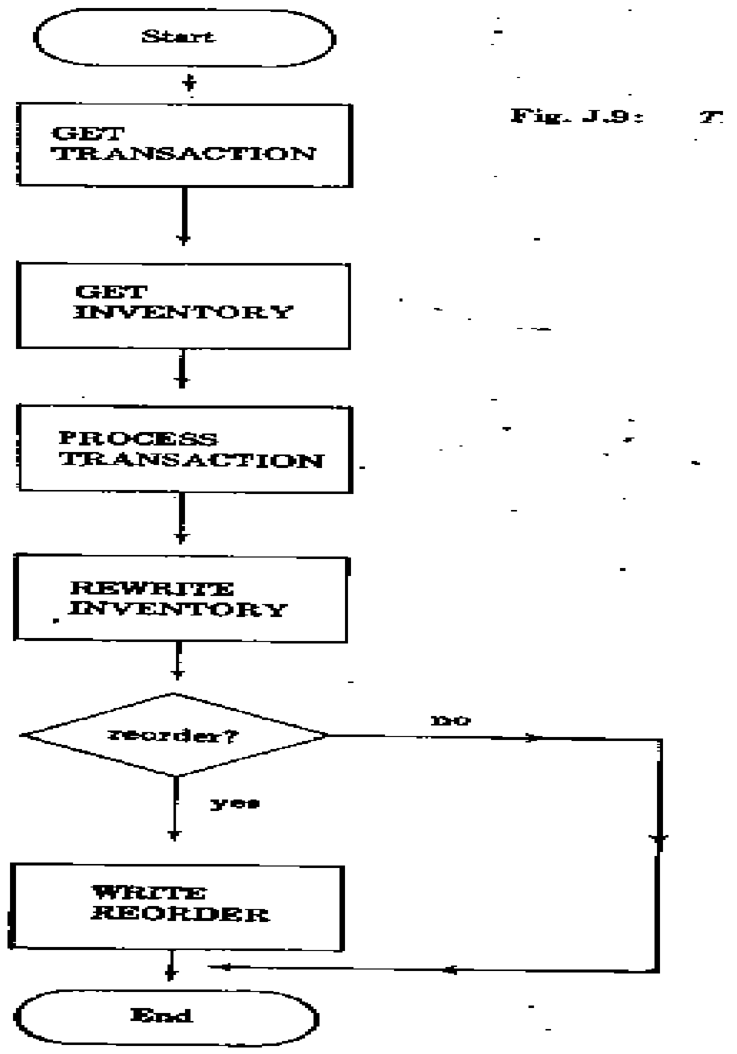
Design Tool - Flow chart



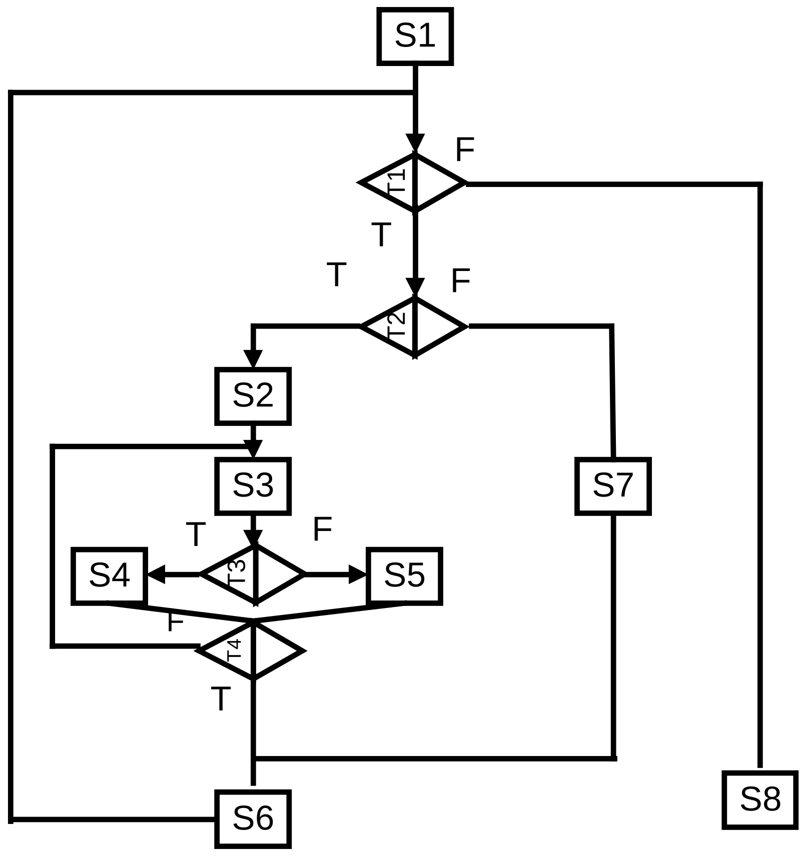
Design Tool - Flow chart



Design Tool - Flow chart



Program Logic Flow Diagram



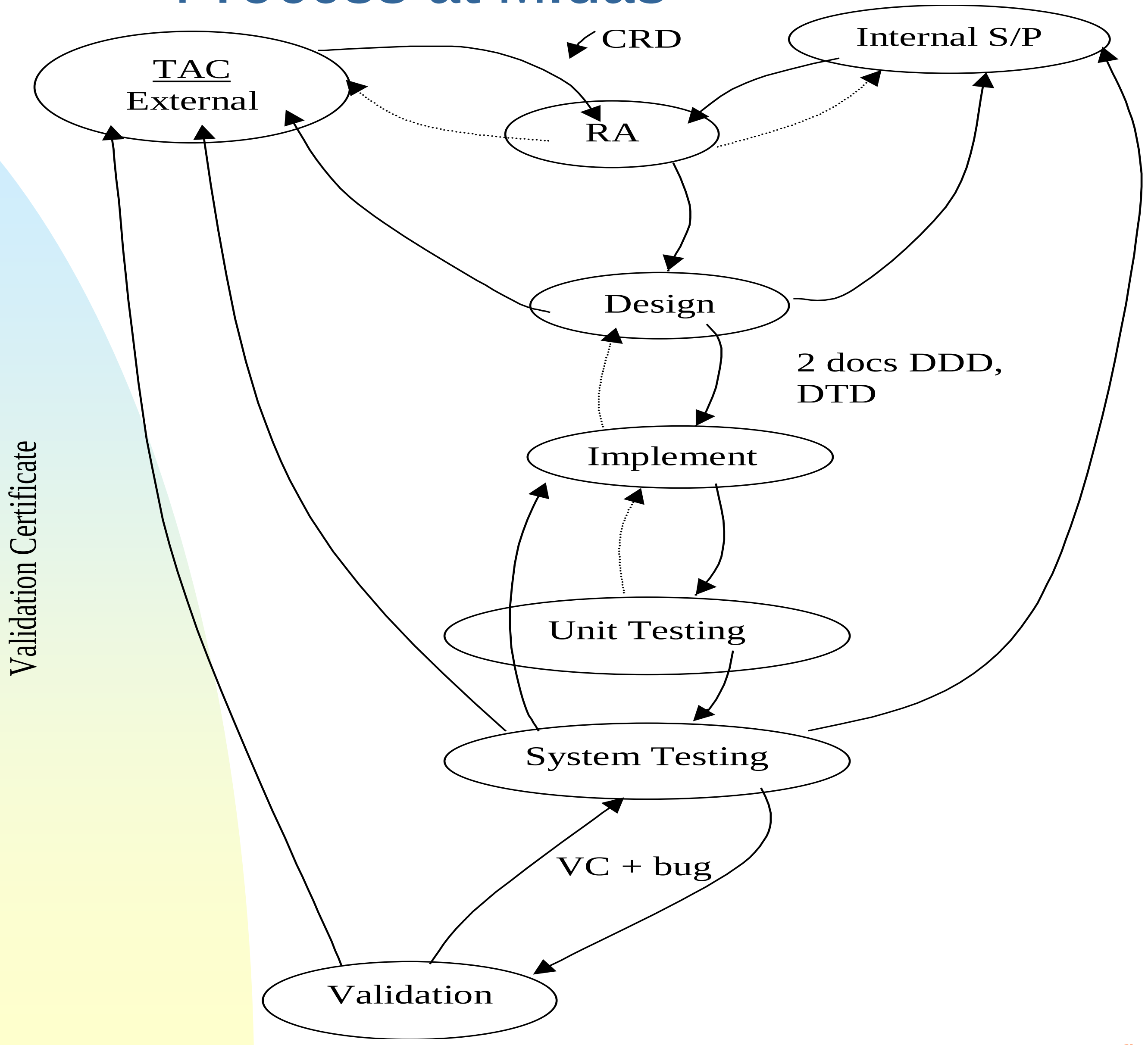
Verification & Validation

- Validation is the process of evaluating s/w at the end of s/w development process to determine the compliance with the requirements
- Unit Testing, Integration Testing,System Testing
  - Functional Tests:
    - Tests on exercising the code with nominal input values for which the expected results are known, as well as boundary values,Min/Max values on the functional boundary.
  - Performance Test:
    - Determines the amount of execution time spent in various parts of the unit, program throughput,response time and device utilisation by the unit.
    - Fine Tune the performance of units that contribute to the overall performance of the system.

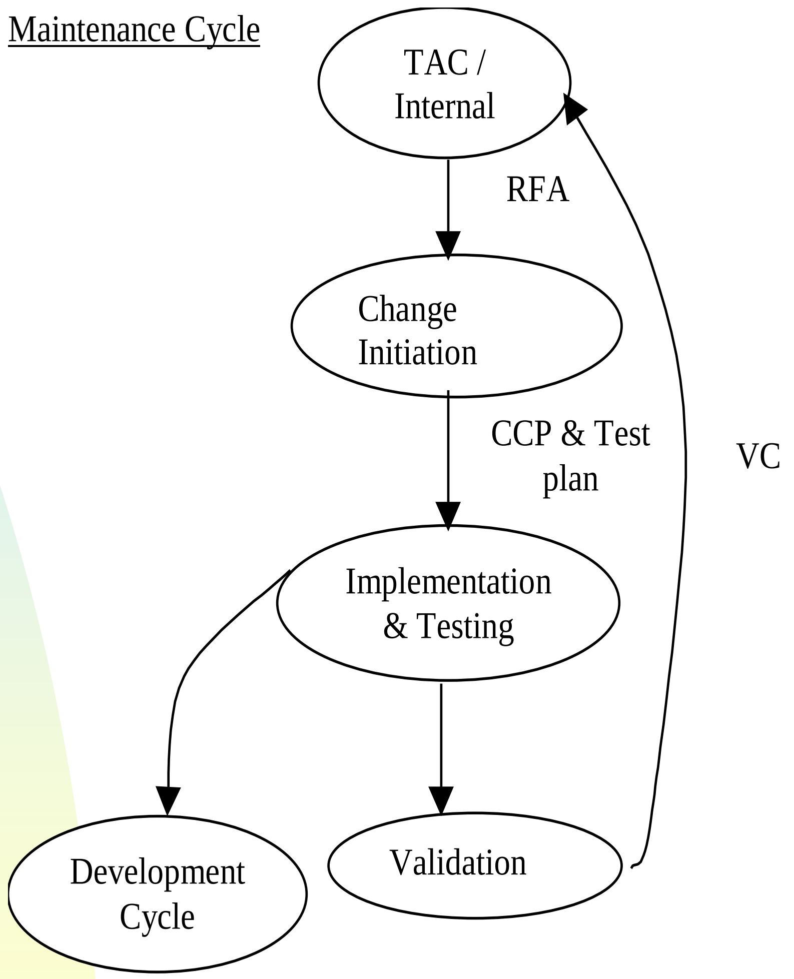
Verification & Validation (contd.)

- Stress Test:
  - Are tests designed intentionally to break the unit.Strengths and limitations of a program may be learnt from outcome of these tests.
- Structure Tests:
  - Exercises the internal Logic of the program
  - Black-Box testing = Functional + Stress Tests
  - White-Box testing = Structured Tests.
    - Decide which paths to test
    - Derive test data to exercise those paths
    - Determine the test coverage criterion to be used
    - Conduct the test.

Proposed Software Engineering Process at Midas



Proposed Maintenance Cycle at Midas



Case Study

- Problem Defn: WSIP application code upgrade in the Flash prom through the serial interface.
- S/W Model used is the Evolutionary Model
- The prior implementation of the same had few drawbacks I.e:
  - Writing to flash used to fail.
  - Code upgrade of the WSIP hampered with the normal functioning.
  - Downtime of WSIP was very large say 5 min.
  - The interfaces were not suitable for the envisaged On-Air code upgrades
- Requirements of the new code upgrade system :
  - Eliminate the above mentioned drawbacks
    - To provide a reliable mechanism with minimum downtime
    - Build interfaces suitable for future enhancements

Case Study (cont.)

- Analysis
  - The current system was understood in terms of the state - diagrams
  - The problems that caused the drawbacks in the prev implementation were identified.
- Design
  - The changes to overcome the drawbacks were specified.
  - A new state machine was designed in view of the requirements to overcome the drawbacks and incorporate new features.
  - Data structures, Module interfaces and Data/Control coupling were specified.
- Implementation, Testing and Integration with main stream done.
- Formal Documentation of the process change done.

Summary

- Need for software engineering understood.
- SDLC and Software Models studied.
- Tools and Notations used in analysis and design studied
- The proposed SEP & Maintenance Model at Midas was presented.
- A Case study was also taken up.