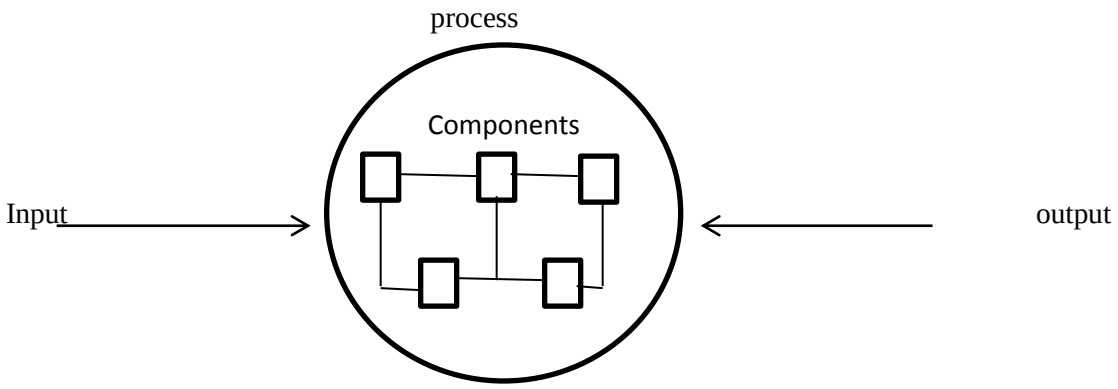
.

➢ What is System?

→ " A system is a collection of interrelated components that work together to achieve some objective"



→ In business system analysis define a set of components that interact to accomplish some purpose.

The word system is derived from the Greek word "systema" which means amount of the functioning units or components.

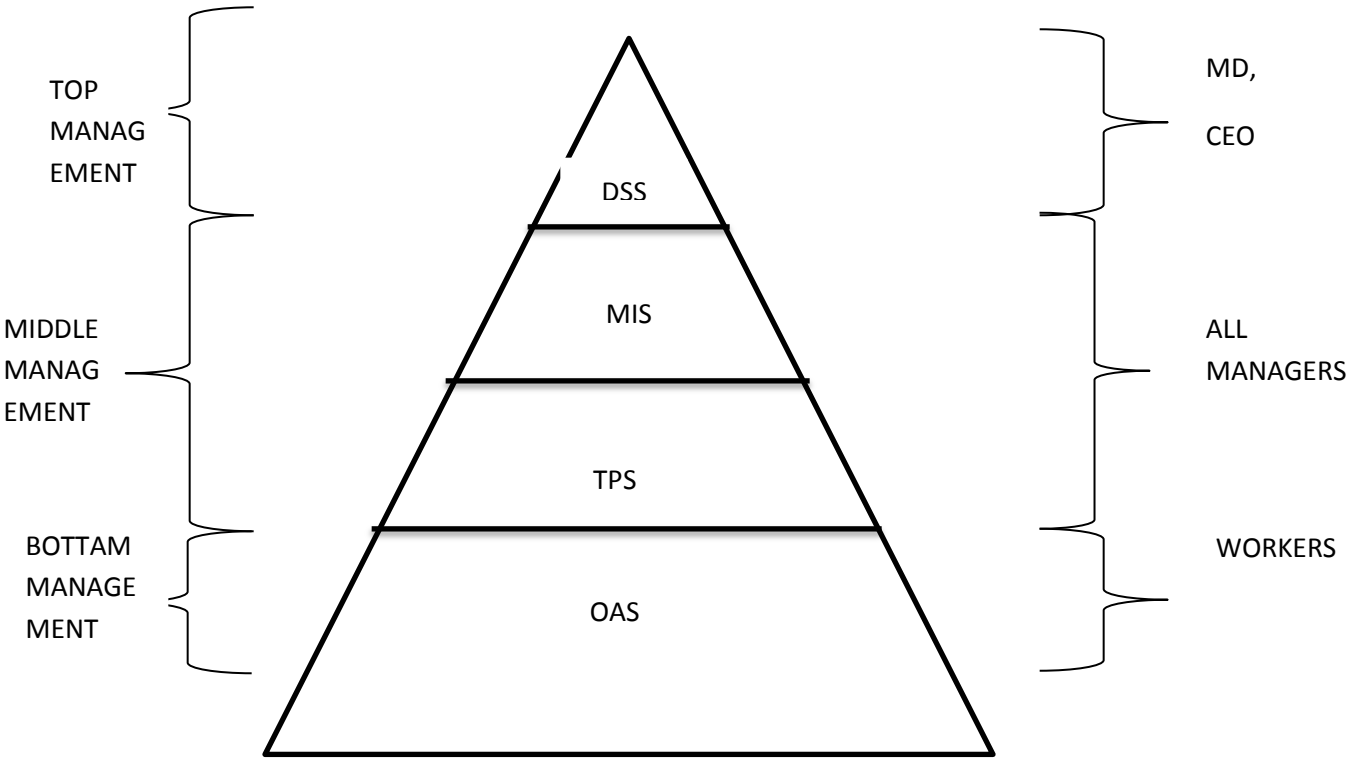
→ In order daily life , we come into contact with the transportation system, education system, communication system ,accounting system , computer system etc.

→ There are business system that means which business organization achieve their pre-determined goals, some organization has to produce profit some has only service not profit for example civil hospital, Gujarat state board transport service some colleges.

➢ Types of System:

➢

1. Office automation system (OAS)
2. Transaction processing system (TPS)
3. Management information system (MIS)
4. Decision supporting system (DSS)



→ Office Automation system:

→ Office automation system are amongst the newest and most rapidly expanding computer based information system.

→ They are being developed with the expectation that they will increase the efficiency and productivity of office workers administration clerk.

→ Transaction processing system:

→ Transaction processing system deals with well-structured routine process includes application keeping records , placing orders , dealing customer, detail of employee and banking information are some of the common transaction.

→ Management information system.

→ Management information system provides input to be used in the managerial decision process deals with supporting well-structured decision situation.

→ Decision supporting system:

→ Decision supporting system provides information to managers who make judgement about particular situation.

➢ What is sub-system:

→ A system that is part of some larger system. A group of related components that are parts of some system.

→ " A sub system is a set of elements which is part of that particular system".

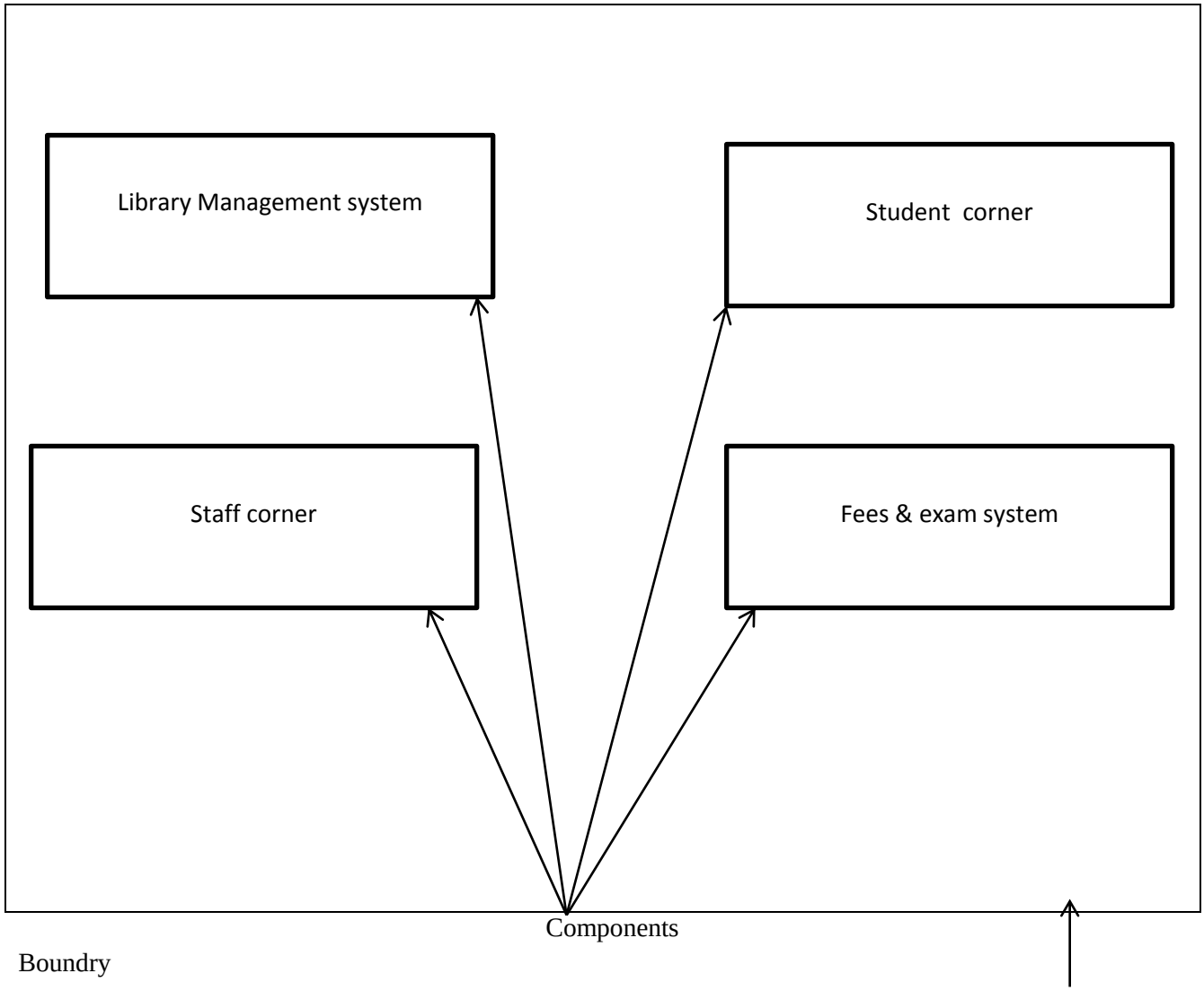


Figure:- Sub-system

→ **Information Technology /System OR Information & Communication Technology.**

- Software
- Hardware
- BPO,KPO (Business/knowledge process outsourcing)
- Mobile technology

- → "An information system is only combination of peoples activity using technology to support operations, management, decision making."
- → In very broad sence the term information system is frequently used to refer to the iteration between people, process, data and technology.
- → Information and communication technology (ICT) is used to organize with support of business system or business process.
- → Information system helps to control the performance of business processes like capturing data, storing data, retrieving data, manipulating data and displaying data.

Software Crisis & Software Failure:

- → Software crisis means some problem is arriving when our project is going on.

1) Programmers points of view:

- Problem of portability.
- Problem of documentation.
- Problem of co-ordination
- Problem of errors.
- Problem of pieracy of software
- Problem of implementation and maintenance.

2) From the users points of view.

- How to choose a software from market of availability.
- Problem of virus.
- Problem of software errors.
- Problem of protected data.
- Problem of different versions of software.

- **Roles of system Analyst:**

Que: Who is system analyst?

- → **"** The system analyst is the person who guide through the development of the information system. **"**
- → While performing this task analyst must always meet the information system objective with the goals of organizations.
- → A system analyst is a person who conduct a study, identifier of activities and objectives determination of all the activities.
- → Designing and implementation system to as organizational needs are functions of the system analyst.
- → System analyst play a major role in business benefit from computer technology and that's why the analyst has all the skill.

- **Attributes of system analyst:**
  1) **Knowledge of people.**
  2) **Knowledge of business function.**
  3) **Ability to communicate.**
  4) **Knowledge of data processing.**
  5) **An analytical skill/mind.**
  6) **Flexibility.**

1. Knowledge of people.

→ A system analysis works with other so closely he/she must under stabbed their needs and motivates them to developed system properly.

2. Knowledge of business functions.

→ A system analyst must know the environment in which he/she works.

→ He must be aware of the function of management and knowledge of accounting marking principle.

→ He must be familiar with his company management policies.

3. Ability to Communicate.

→ As a co-coordinator system analyst must communicate properly with people and different levels of origination.

→ System analyst must lisen carefully to what other say and integrate the thought of others into the system developments process.

4. Knowledge of data processing.

→ Most system today are computer based .the system analyst must potential aware about the potential and limitations of computer.

5. An Analytical Skill/Mind.

→ In organization there are so many situations that is very tough or cmlex . It will not solve by has programmer so analytical skill to solve that problem at tome of information gathering.

6.Flexibility:

→ System analyst must be flexible in their thinking and different function in an organistion have need and most system are result is to produce the system that will be best for his organistion

Software Process Development Models.

- SDLC(Software Development Life Cycle)
- Spiral Model
- Prototype Model
- Agile Model

1.SDLC(Software Development Life Cycle).

→ System development is a birth-to-mature process called **system development life cycle.**

→ A system project begins with a decision to analyze user requirements. The project is completed once a useable s/w product is produced and implemented or released. The period of time that begins when a system is fully operational for use and gets maintained is called the development life cycle of a system.

1) Preliminary investigation
(2) Feasibility Study
(3) System Analysis
(4) System Design
(5) Development of Software
(6) System Testing
(7) Implementation & Evaluation
(8) Maintenance

1. Feasibility Study

→ A feasibility study is undertaken to determine the possibility of either improving the existing system or developing a completely new system.

→ The purpose of feasibility study is to determine whether the requested project is successfully possible.
There are three aspects of feasibility study.

1) Economical Feasibility Study:
2) Technical Feasibility Study
3) Operational Feasibility Study

2. Analysis and Design:

→ Analyst gather the requirement for system, this stage includes in detail study of the business needs of organization, and for changing the business process may be consider.

→ Analyst is the main stage of the system development life cycle and we have to gather all the right information from the client side.

→ After we have to go design of the system, design consider the look of the system for example, menu design, window design, color combination, tabular format etc.

→ Analyst and design are very crucial in the whole development cycle, this phase would be very expensive. Software development must care in taken during this phase.

3. Coding:

→ In this stage, we have to writes all the code for appropriate system. In this stage programmer write the program for the particular module. In coding, stage we have to apply programming methodology for the solving complex situation and better programming.

4. Testing:

In this phase programmer or tester test all the modules one by one. Normally programs are written as a series of individual models. This subject to separate and detai.List of test .this system is tested with some testing strategy like.

- Black box testing.
- White box testing.
- Load testing
- Performance testing
- Unit testing.

5.Implementaion

→ IN this phase, the designs are translates through the testing departments than we have to go for implementation of the system.

→

→ Implementation is nothing but one kind of training to the client and suggestion to the user and operators.

→ In implementation phase we can also give the knowledge of hardware , software ,system installation etc.

6.Maintenance.

→ Any software system will need maintenance software will definitely going to change once it is delivered to the customer.

→ There are many reasons for the maintenance or changes could happen because of some unexpected input value into the software operation.

→ In addition the change in the system could directly effect the software operation.

→ There are three types of maintenance

1. corrective maintenance
2. adaptive maintenance
3. perfective maintenance

1. Corrective maintenance:

→ Corrective maintenance deals with the repair or faults or all defects found

→ A defect can result from design errors , logical errors and coding errors.

2. Adaptive Maintenance:

→ Adaptive maintenance consist of adapting software to change in the environment , such that the hardware or the operating system, business rule , government policy ,new pattern , hardware and software devices ,or platforms.

3. Perfective maintenance.

→ Perfective maintenance mainly deals with new changes or some enhance to the user requirement.

Fact finding technique.

→ The specific method analyst use for collecting data about requirements that is called fact finding technique.

→ Analyst use this technique to help accurate information from the client side.

1. Interview

1) Structured interview.

→ In structured interview we have to collet some of questions from all our respondence.

→ this technique is easy to collect and evaluate more objective from both side (client side and programmer side) with answer to question.

Limitations:

→ Cost of preparation is high.

→ Client may not accept high level of structure and mechanism of questions.

→ High level of structure may not be suitable for all situations.

2. Unstructured Interview.

- Interviewer has granter flexibility in questions for very easy to client.
- Interviewer may produce information about Ares that were overwork or most important.
- Unsterture interview has no sequence and question related to sub modules or sub system.

Limitations:-

→ Time consuming.
→ Interviewer may introduce their unused questions .
→ Unexpected information may be gathered
→ Take extra time to collect essential information or facts.

2. Questionaries :-

- Questionaries' are another way gather information where the potential user of the system are potential user of the system are given questionaries' to be filled up and return to the analyst.
- Questionaries' are useful when the analyst need to gather information from a large no of people.
- It is not possible to interview each individual also if the time is very short in that case also questionaries' are also useful.
- There are two types of questionaries'.

1.Open ended Questionaries or Open response.

- The objective of open response questionaries is to gather information and data about the essential and critical design feature of the system.
- The open ended question require to learn about the filling operon And experience of the respondent.
- This information is also used and help the making of the system very effective because the analyst can offer sub-sequent modification as per knowledge.

2) Close ended questionaries or close response

- The objective of close response questions is to collect the target information (small questions) of the system.
- It give an insight how the people deals with the system, behave with the system and how confortable are they with it?
- In this case respondent how to choose from a set of given response. The close questions can be of various types.
  - **Fill in the blanks.**
  - **Multiple choice questions.**
- **Yes or No type.**

→ **Close ended question format.**

Yes/No Questions		
Questions	**☑ Yes**	**☒ No**
Do you want reports? Do you want calculator?		

Multiple choice Questions

How many client do you obtain in year?	**A)1-100** **B)101-500** **C)501-1000** **D)1001 above**	☐ ☐ ☐ ☐

Open ended question format.

1) **What is the salary calculation process in your company?**

ANS:_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

3: Record review

- → It allow the analyst to collect information about the various need of system from a large no of activities.
- → The use of record review technique is more reliable than the other fact finding technique.
- → In record review technique we have to collect all the reports from the client side for example
- → salary slip, salary report, attendance report, sales report, invoice, purchase report, account report, sales order report, yearly sales order report, quarterly sales orderly report, weakly

→ purchase order report yearly, yearly purchase order report, employee salary report, employee salary report.

4: Onsite observation

- → It allow analyst to gain information from the side through onsite observation.
- → Onsite observation analyst can secure the first hand activities process.
- → This method is very much useful when analyst need to actually observe that how the documents and transaction are handeled? How process are carried out and whether specification steps are actually followed or not?
- → Experience analyst can give a lot of thing from the observation.
- → Onsite observation provide close review of the working of the system. The analyst can observe people, objects, documents and events.

❖ Advantages

1: Data gathered by observation can highly reliable.

2. observation is likely to over the existing system.

3. the analyst is able to see exactly what is being done comlex tasks are sometime difficult to clerly explain in word.

➢ Simulation.

→ Simulation of the system is two way.
1.By developing a model of the system and testing it.
2. By passing dummy module through the system.

→ Simulation is very expensive and large volume of dummy data must be used to check all the requirement of the system.

Tools for Documentation procedure.

1.Decision Tree.

- → Decision tree is a graphical representation of particular condition and acions .decision tree method displys the relation of each condition and it action.
- → The Decision tree provides the graphical representation of decision logic that help the non-computer people find easy to understand.
- → Decision tree is the one of the methods for describing decisions. Decision tree are commonly used in operation research especially in decision analysis to help identify of some strategy most likely to reach a goal.

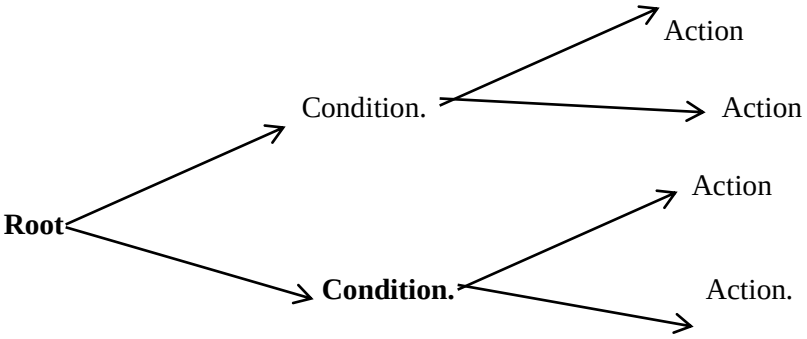
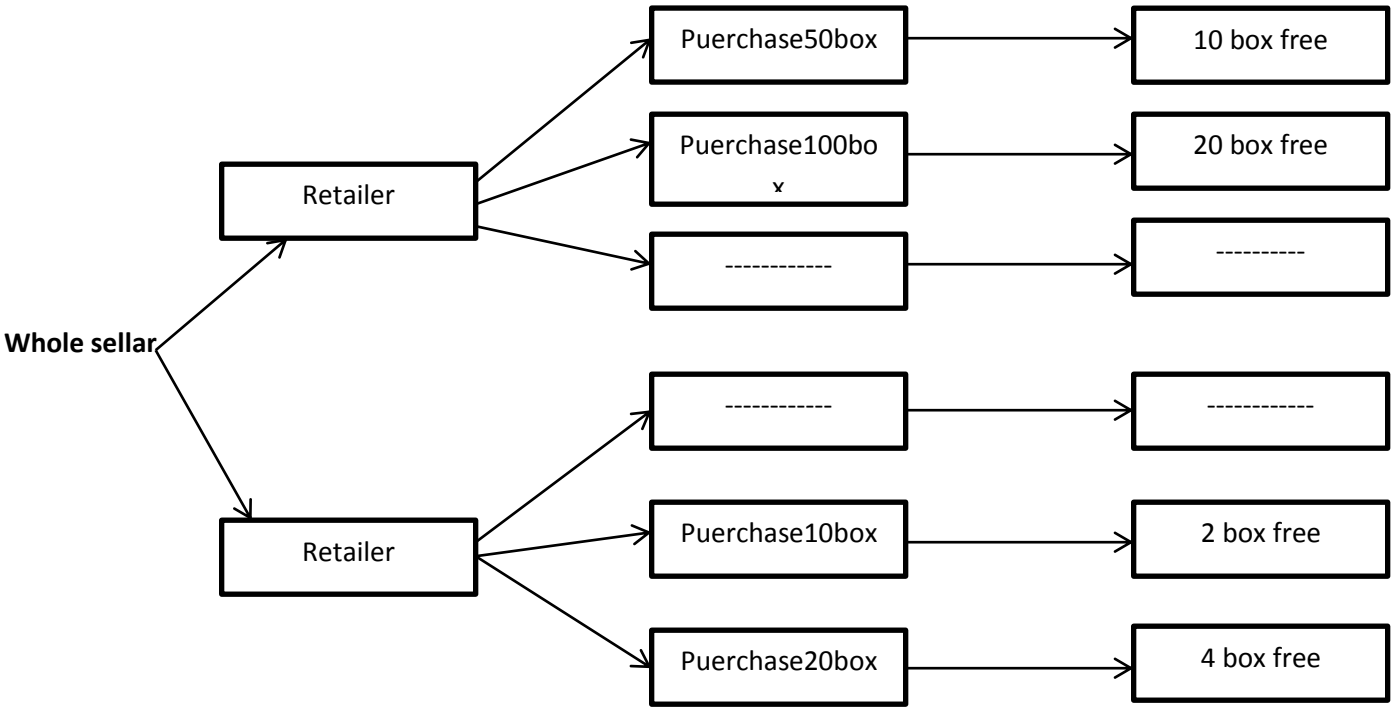


Figure:- Decision Tree Structure.

Project manager ask "how do we accomplish this , common process frame work activity.



Advantages

1. Decision tree reassembles the branches on a tree.
2. Root of the tree is a starting point of the decision tree.
3. The particular branch to the follow depends on condition that exist and decision that is to be made.
4. Decision tree gives various condition that has various actions so we can easily understand particular situation of the system.

2.Decision table:

→ Decision table is MATRIX representation (ROWS & COIUMN WISE).

→ Organization have complex policy and decision making rules in business process.

→ Decision tables are populatr tool in analysis or expeissing this requrements .
Decision table is a tabular format of representation that satisfy a set of condition of their corresponding action.

→ The general format of decision table is given below.

Condition statement	Condition Entry
Action Statements.	Action Entry.

1. Condition statement identifies relevant condition.
2. Condition entry tells which value applies for particular action.
3. Action statement list is the set of all steps that can be taken when certain condition occur.
4. Action entry display that specific action in the set to take when in the set to take when selected condition or combination of conditions are true.

Examples.

→ **A technique support company write a decision table to printer problems and we have to solve with decision table of that particular condition.**

	Rules							
Printer does not print	**Y**	**Y**	**Y**	**Y**	**N**	**N**	**N**	**N**
A red light is fleshing	**Y**	**Y**	**N**	**N**	**Y**	**Y**	**N**	**N**
Printer is reignited	**Y**	**N**	**Y**	**N**	**Y**	**N**	**Y**	**N**
→ **Check power cable?**	**x**							
→ **Check printer & wimp cable ?**		**x**						
→ **Driver or software ?**			**x**					
→ **Check printer ink**							**x**	

Advantages.

1. They are easier to draw.
2. Decision table provide a compact representation of the decision making process. A small table can replace more pages of flow charts.
3. It is also much easier to understand a particular path , discoing table , are best suited for calculation discounts tax, inventory control procedure Like time behavior , resource behavior.

Disadvantages:

- → Decision table cannot express the complete sequence of operation to solve a problem.Therefore it may be difficult for a programmer to translate a decision table directly into a computer program.
- → When there are so many alternatives decision table cannot list them all.

1) Structure English:

- → Structure English is an attempt to allow the use of natural language.
- → Structure English skip ambiguity to express action to be taken under particular conditions.

This is accomplish by:

- → Choosing a simple subset of natural language verbs.
- → Choosing a simple subset of natural language nouns.
- → Defining, expressing very sequentially.
- → Selection.
- → Iteration.

- → So complex programming require some time find out their definition and syntax or flow the process. This is almost impossible to combine knowledge of business domain and technical proficiency in the programming language in one individual.

- → The structure English is also known as " pseudocode " and structure English is used as analytical tool in software industries.
- → This "pseudocode"(Script) is passed to programmer who translate it into actual programming code.
- → We can use some common keyword like:Begin,end,stop,do,while,until,do while,if,else,elseif,between,true,false,create,open,and,or,not,get,close,with,sort etc..

➢ **Example of structure English:**

→ A blank will grant loan under the following conditions.

1. If a customer has an account with the bank and had 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 given.
3. Reset all loan application in all other cases.
4. **Ans:- Structure English of particular condition.**

— **IF customer has bank account THEN if customer has no dues from previous account THEN**

Allow loan Facility.

Else

If management approved is obtain THEN

Allow loan facility.

Else

Reject

end if.

End if.

Else

Reject.

Endif.

4. Data flow Diagram / Data Flow Analysis Tool.

→ When data flow diagram are the method of choice or technical description for some reasons.

1. Data flow diagram is easier to understand by technical and non-technical audience.
2. Data flow diagram can provides a high level of system overview, complete with boundaries and connections to the other system.
3. Data flow diagram can provide details representation of system component.

→ Data flow diagram helps system designer and others during initial analysis stages to visualize a current system or requirements.

→ Data flow diagram represents the following areas.

1. External devices and receiving data.
2. Process is that change the data.
3. Data flows
4. Data storage location.

→ Data flow diagram consists of some levels, likes context level (0 levels). 1st level, 2nd level etc…. that represent different parts of the system.

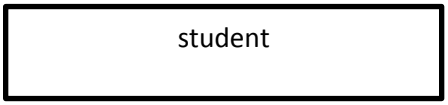
→ Data flow concept

Data flow diagram consists of four basic components like:

1) Entity
2) Process
3) Data flow
4) Data store

1Entity:

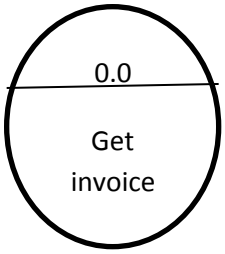
- An entity is the source or destination of data. The source in a data flow diagram represents entity that is outside the system.
- Entity provides data to the system and receives data from it.
- Entity is often represented as rectangle.
- Entity is also referred to as source ore destination.

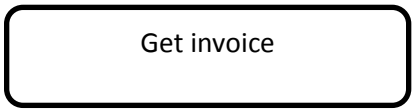


2) Process :

- The process is the manipulation or work that transfer data or perform decision on business rules.
- In other words process receives data and generates some output.

- Process name is simple verb or name of the data flow such as “payment system”, “school information”,”get invoice”.
- Process can be drawn as circle or segmented rectangle.





3) Data flow:

→ Data flow is the movement of data between the entities.

→ Data flow is the name to reflect the nature of the data used (this name shouted also be unique with in specified data flow diagram).

→ Data flow is represented by the arrow with specified data name.

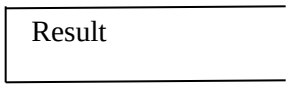
4) Data store

→ A data store is used for storing the data.

→ Data between processes is used for retrieving the information for in the future use.

→ Files and tables are considered for the storing of data.

→ Data store name are simple but meaningful such as “customer care”, “student record”,”result”.



Data dictionary:-

- Data dictionary is a method that means it will be data about data.
- Data dictionary is a store house of data giving information of data. It is a list of all items and data file of a system.
- Data dictionary is a catalog of the elements in a system.
- In a data dictionary you will find list of all the elements related to system.

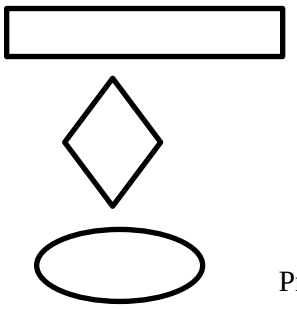
Format of data dictionary

1) Data elements
2) Data name
3) Data description
4) Data characteristic
5) Data composition

Example:Student information system:-

Data name	Size	Data type	constraints	Description
S_name	15	Character		Name
Surname	15	Character		Surname
Father name	15	Character		Father name
Address	50	Character		Address
Roll_no	5	Number	Primary key	Roll no
Contact_no	13	Number		Mobile no
Father's no	13	Number		Mobile no
Email	50	Character		E mail id
B_date	10	Number		Birth date
Enroll_no	15	Number		Enrool no
Cast	10	Character		Cast
Last_school	25	Character		Last school
Last_result	4	Character		Lasr result
Blood_group	4	Character		Blood group
Gender	5	Character		Gender
Icard_no	7	Number		Card no
D_licence_no	10	Character		Licence no

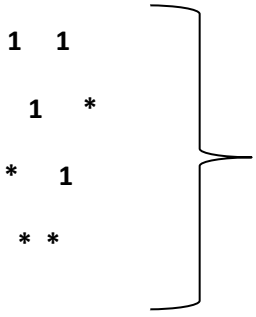
- **ER DIAGRAM (Entity-Relationship Diagram)**
  - Entity relationship diagram is a graphical representation of entities and their relationships to each other.
  - Entity relationship diagram has some notations like



Entity

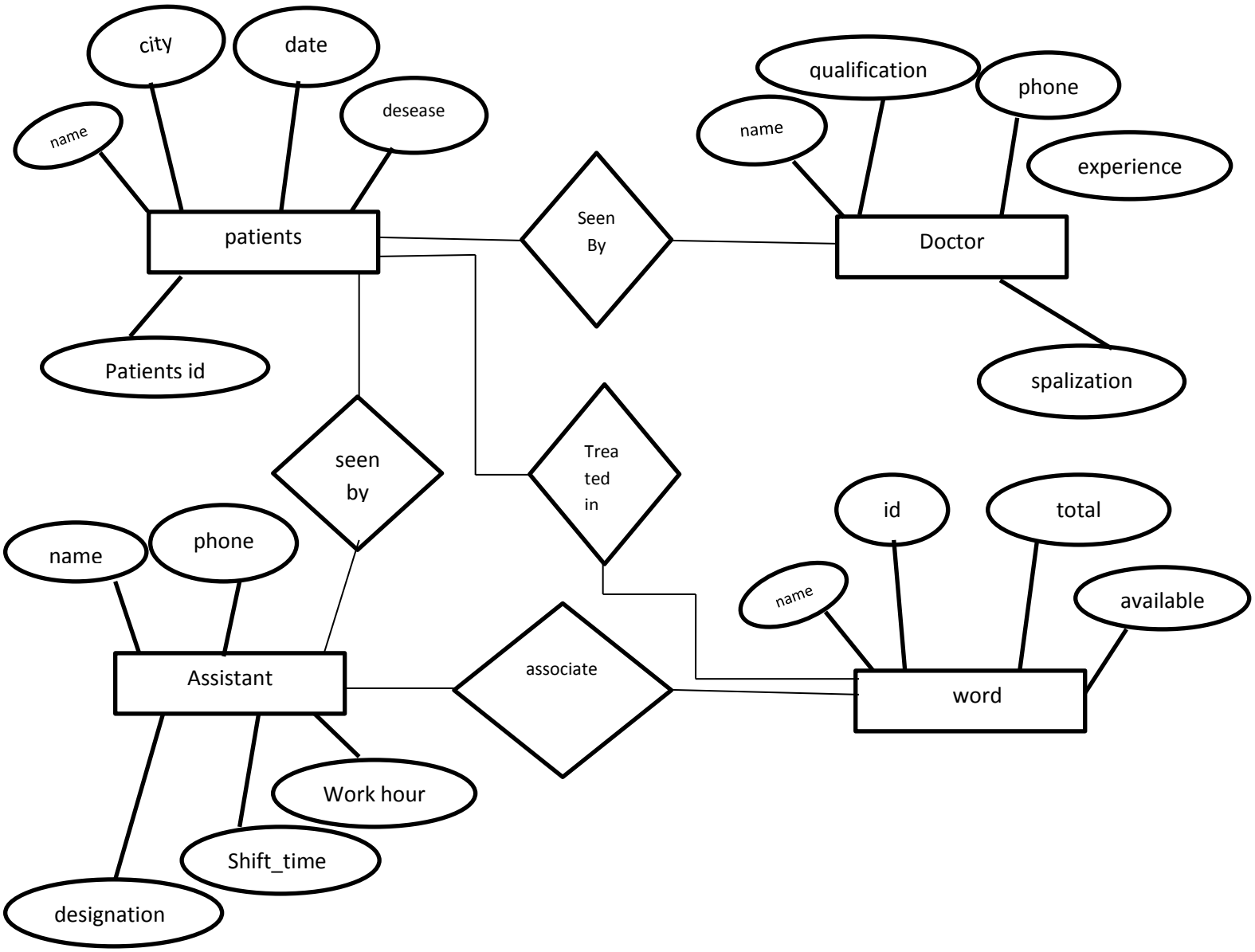
Relation

Properties (fileds)



Relation flow

Example : hospital management system.



Hospital Management System.

SE (Software engineering)

❖ What is software?

- By definition the software is collection of computer programs, procedures and documentation that perform different task on a computer system.
- The term 'software' was first used by 'John Turkey' in 1958.
- There are some types of software, system software ,programming software, utility software, data backup software and recovery software, inventory software Web Development Software.

History of Software Engineering

→ The establishment and use of some engineering principle in order to obtain economically software that is reliable and works efficiently.

→ At time professor "Frits Bayer" at the first NATO (North Atlantic Treaty Organization) conference of software engineering in 1969.

❖ Defination of Software Engineering

- The IEEE computer society "Software Engineering is the application of systemic deciplined and and quality approach to the development, operation and maintenance of software i.e software engineering."

❖ Why Software Engineering ?

- To deliver quality product.
- Increasing competation in the software industry.
- Increasing development cost.
- Time to market.
- Fast changing technology.
- Sensitivity in software products.

Goals of Software Engineering

1. Efficiency:
   The ability for software to use computing resources effectively (space and time).
2. Abstraction:
   Abstraction is nothing but mirror copy of particular information.
3. Uniformity
   Make everything look similar.

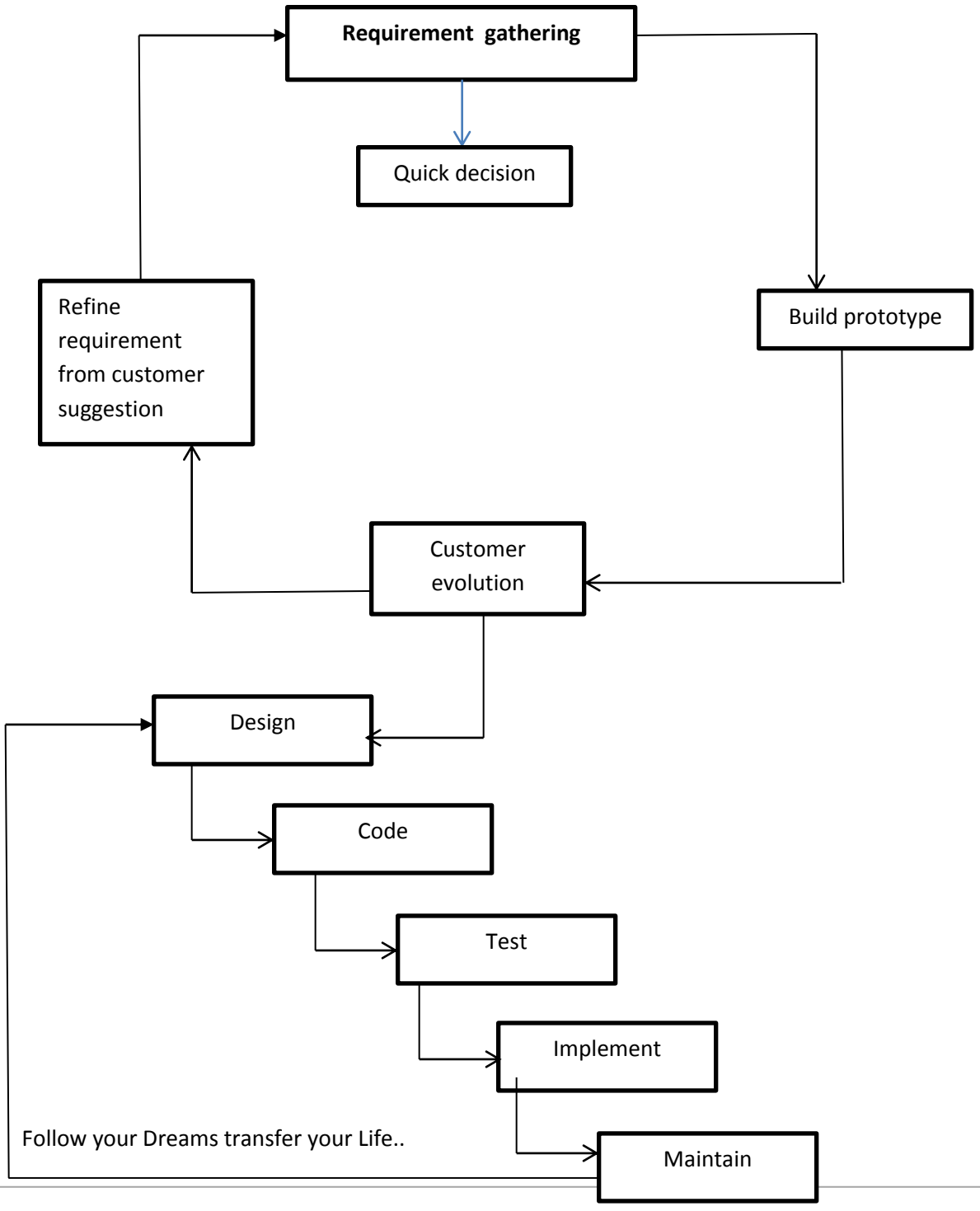
4. Modularity:

Divide data of complex program or software.

Software Development process model

- **Prototyping model**
- **Spiral model**
- **Agile model**

1) Prototyping model



- The prototyping model requires that before caring out the developing actual software.
- A working prototype of the system is a toy implementation of that system.
- A prototype is usually build using some shortcuts the shortcuts might involve using dummy function.
- A prototype usually turned-out to be a very complex version of actual system. It uses limited functional capabilities, low reliability, and inefficient performance as compared to actual system.
- In prototype model we have to develop graphical user interface (GUI) of a particular system for the user. It becomes much easier to give opinion of a user with a working model.

Advantages of prototyping Model:

- It saves time on system development.
- Encourage communication between developer and user.
- Missing customer services can be detected.
- System can be delivered within proper time therefore, delays are minimized.
- If proper tools are used low cost is required for development of prototype.

Tools available for prototyping:

1) screen generator
2) Report generator
3) Application generator

2. **Spiral Model:-**

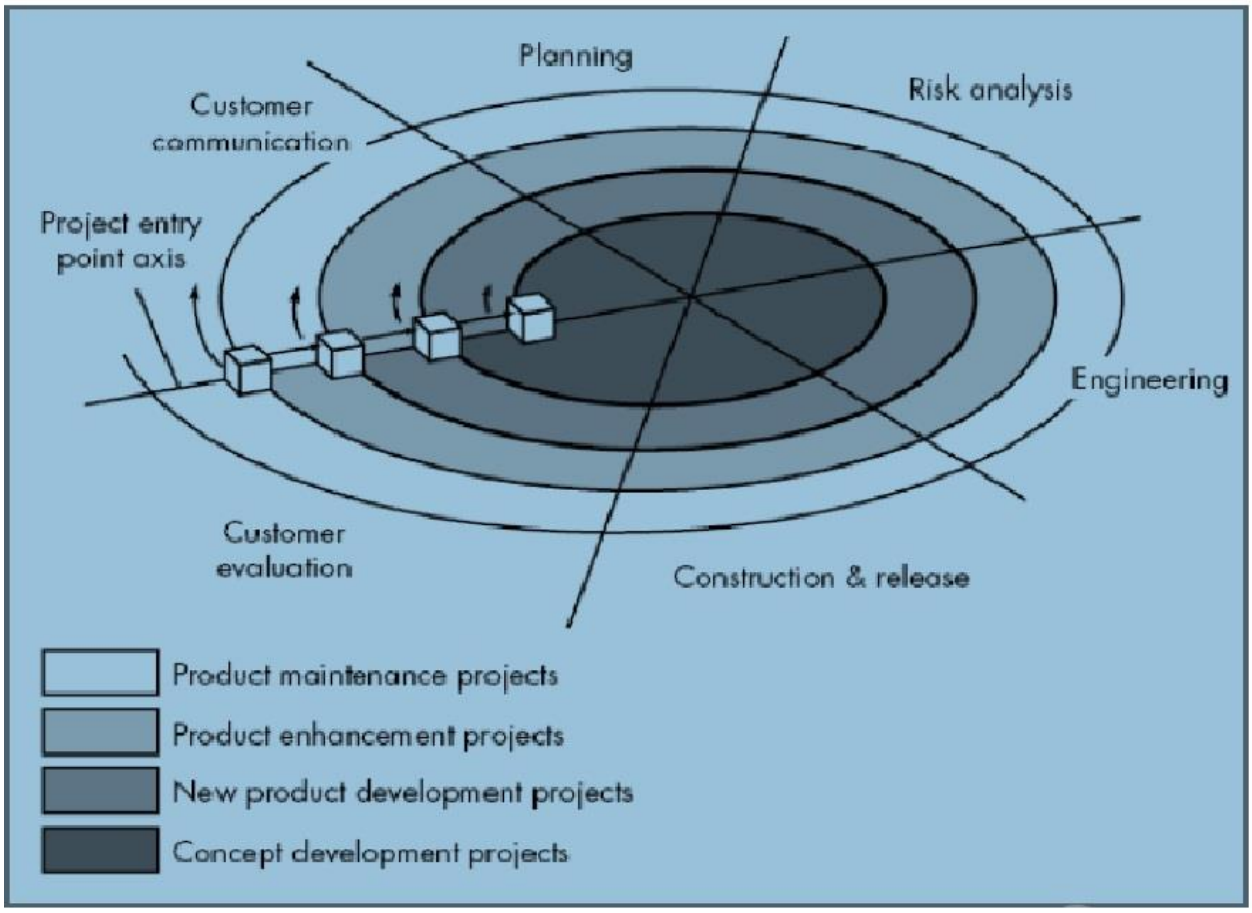


Figure:- Spiral Model.

→ A spiral model is a software development process combining which elements of both design and prototyping.
→ The concept of this model is derived from waterfall model and concept of prototyping and complied project.
→ The spiral model is only for large expensive and complicated project.
→ The spiral model was derived by "*Barry Bohem "in 1986 in his article* , " A spiral model for software development and enhancement."
→ A preliminary design is created for the new system.
→ First phase is most important part of the spiral model.in this phase all provide alternative which can help in a cost effective project are analyzed.
→ If risk identify and indicates and indicates any kind of uncertainly in requirement, prototyping may be used to process with the available data and find out possible situation in order to deal with potential changes in the requirement.

Application of Spiral Model.

1. Game development is a main area where the spiral model is used.
2. The Us Military has adopted the spiral model for Future Combat System program(FCS) in united states of America.

3. **Agile Model.**
   - → Agile software development refers to a group of software development.
   - → Agile software development method is based on iterative development where requirements and situation involve though the combine bet the organization.
   - → Agile method generally promotes a project management process that encourages frequent inspections and adaption with a leadership philosophy that encourage work with co-ordination.
   - → Consumer satisfaction is main objective of our project, and agile model give all the satisfaction like time to market, quality,communication ,daily co-ordination,co-operation between business people and develpers, face to face conversation is the best of quality product.

Software Testing

Q. **What is software bugs & Faults ?**

- → Software bugs are an error mistake , failure , or fault in a computer program or system that producer an incorrect or unexpected result , or cause it to behave in unintended ways.
- → Most bugs arise from mistake and made by people in either a program source code or its design, and compilers producing incorrect code.
- → A program that contains a large no of bugs that bugs seriously unfear with its functionality that is called **Buggy.**
- → Reports detailing bugs in a program are commonly known as bug Reports, fault Reports, problem Reports ,trouble reports change request.

Defects and failure

→Not all software defects are caused by coding errors. One common source of expensive defect is caused by requirement gaps, for example unrecognized requirement.

→A common source of requirement gap is testability, maintainability, usability performance and security.

➢ Test Case

→ A test case is a set of conditions of variable and inputs that are developed for a particular goal of objective to be achieved on a certain application to judge its capability or features.

Test case structure

→A formal test case written of three parts

1. Information
2. Activity
3. Result

1.**Information**

→ Information consist of general guidelines about the test case.

→ Information identifier

→ Test case create name
→ Test case version
→ Name of the test case
→ Price description.

2. **Activity**

→Activity consist of the actual test case activities. Activity contains information about the test case environment, activity to be done, a test case initialization, step by step activity.

3. **Result**

→Result are outcomes from the performed test case.
→Result data consist of information about expected result and the actual result.

Example: of Test Case

Test	Test case Description	Expected Result	Pass/ Fail	Actual Result	Comment	Bug id
Versionre 101 function login/out.						
(a)	Open up a browser to http:\\url & click on go button	Make sure login page should display	Pass	-	-	-
(b)	Enter wrong password and wrong username and click on ok button.	Make sure error message should display	Pass	-	-	-
(c)	Enter valid password and invalid username and press ok button	Make sure error message display as “please enter correct pass”.	Pass	-	-	-
(d)	Enter valid password and invalid usernames and press ok.	Make sure error message displays as “Please enter correct user”.	Fail	-	-	B1
(e)	Enter only correct password click on ok button. Enter only invalid password and click on ok button	Display message “please give username”.	Pass	-	-	-
(f)	Enter only vaild	Make sure ”please enter correct password”	Pass	-	-	-

(g) (h)	username and click on ok button Enter only invalid username and click on it.	Display message "pleas enter correct username" Display message "please enter correct username"	Pass pass	- -	- -	- -

➢ Test Script

→ A test script in software testing is a set of instructions that will be performed on the system under test to system function as excepted.

→ There are various means for exerting test script.

1. Manual testing
2. Automated testing

1. Manual testing

→ Manual testing is the process of an individual testing. Manual software testing done with some tester software engineers.

→ Manually testing need not any kind of testing software.

2. Automated testing

→ Automated software testing is the process of creating.test script that can be run automatically though some software.

→ Automated testing need not neet human interaction.

→ Eg. Applictions of Automated Testing.

- Winrunner
- Loadrunner
- QTP
- Rational

➢ Test Harness.

→ In software testing a test harness is generated by automated software with collection of errors and data.

→ It is configure to test a program unit by running it under various conditions and manitering it's has behavior and outputs it has two main parts.

1) Test Executed Time.
2) Test Script Repository.

Sodtware Testing.

Intoduction.

It refers to tests that verify the functionality of a specific section of code,usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

→ These types of tests are usually written by developers as they work on code(white-box style), to ensure that the specific function is working as expected.
→ One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other.
→ Unit testing is also called component testing.

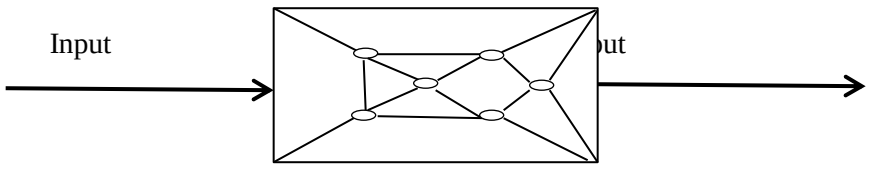
Integration testing

→ Is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be localized more quickly and fixed.
→ Integration testing works to expose defects in the interfaces and interactionbetween integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system

System testing

Tests a completely integrated system to verify that it meets its requirements

White box Testing:-



White box

→ White box testing (clear box testing, glass box testing , transparent box testing , structural testing) use an internal perspective of the system to design test case based on internal structure.

→ It requires programming skills to identify all paths through the software.

→ The tester chooses test case inputs to exercise path through the code and determine the appropriate outputs.

For Example:-

→ White box testing is applicable at the unit testing levels of software testing process, it is applied to the unit testing.

Black box

→ Testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, modelbased testing, exploratory testing and specification-based testing.

Specification-based testing:

Specification-based testing aims to testthe functionality of software according to the applicable requirements.[Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requiresthorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.

Specification-based testing is necessary, but it is insufficient to guard against certain risks.

Advantages and disadvantages:

— The black box tester has no"bonds" with the code, and a tester's perception is very simple: a codemust have bugs.

— Using the principle, "Ask and you shall receive," black box testers find bugs where programmers do not.

— On the other hand,black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. As a result, there are situations

(1) a tester writes many test cases to check something that could have been tested by only one test case, and/or

(2) some parts of the back-end are not tested at all.

Therefore, black box testing has the advantage of "an unaffiliated opinion", on the one hand, and the disadvantage of "blind exploring", on the other.

➢ Software Project Management

→ Management involves the activities and task undertaken by one or more person for the purpose of planning and controlling the activities of the others in order to achieve objectives that could not be achieved by the others acting alone.

→ Management functions can be categorized as planning, organizing , staffing , and controlling.

→ Project management is a system of management procedure practice, technologies, experience , that are necessary to manage successfully an engineering project management.

→The manager of a software engineering project is called a project manager a project head.

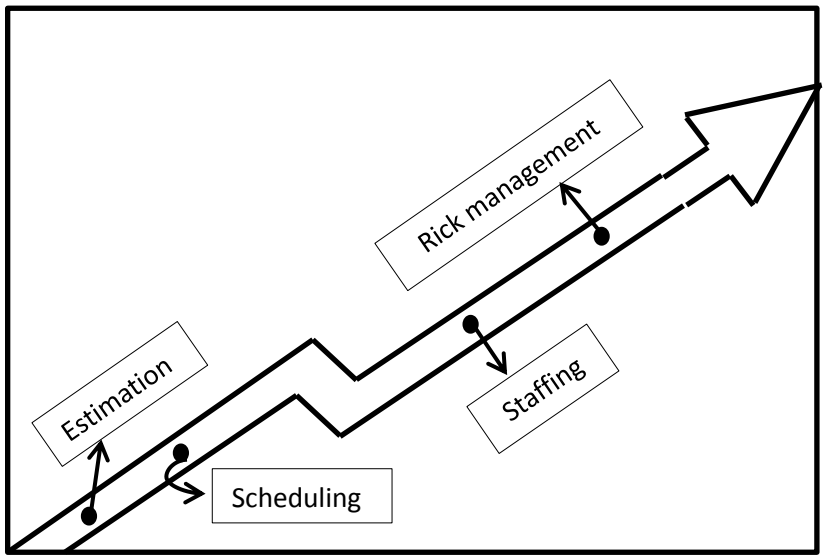


Fig: Software Project Management Process.

1.Estimation :-

- Cost

→ How much is it going to develop the software ?

- Duration

→ How long is it going to develop a project ?

- Efforts

→ How much effort would be required in developing the product.

→ The effectiveness of the all other planning activities such as scheduling and staffing we based on the accuracy of the software cost estimation.

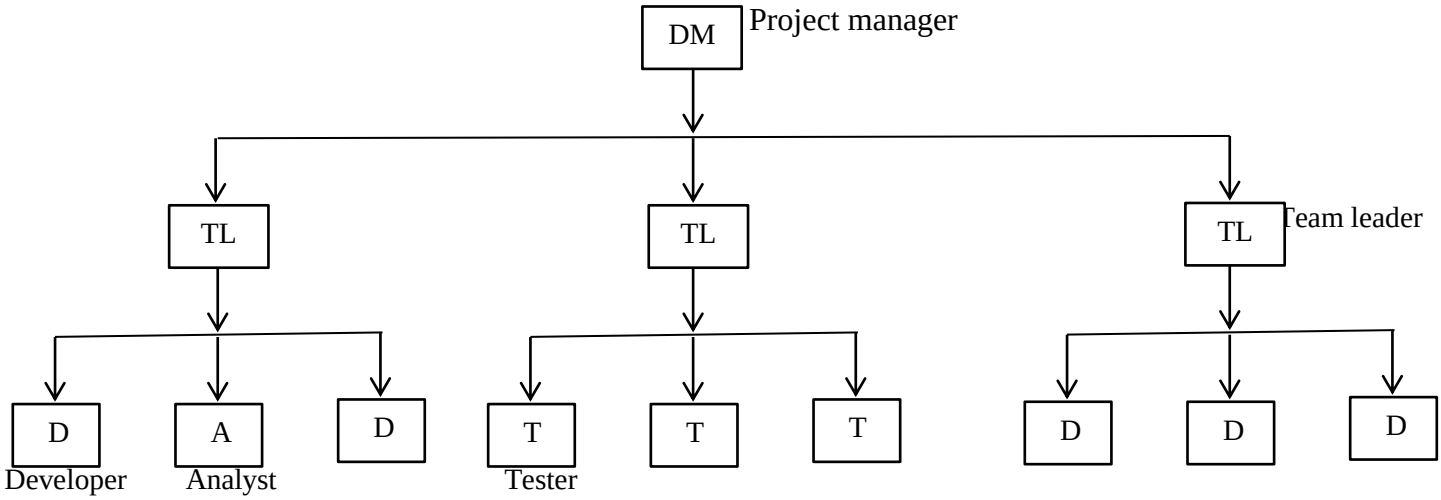
2 Scheduling :-

→ After estimation is made the schedules for men power and other resources have to be developed.

→ This scheduling display the project status and project schedule.

3 Staffing:- (Team Structure)

Staff organization plans have to be made.



4.Risk management:-

⟶ Risk identification risk analysis and planning have to be made.

There are so many type of risk like technique risk, programing risk, design risk, management risk, social risk.

5. Other

→Some plans such as Quality assurance, configuration management plans have to be made.

➢ Software Project Planning

1. **Quality Plan:-**
→Describes the quality procedures and standards that will be user in a project.

2. **Validation plan:-**
→ Describes the approach resources and schedule user for system validation.

3. **Configuration Management Plan:-**
→ Describes the configuration management procedures and structure to be user.

4. **Maintenance Plan:-**
→ Predicts the maintenance requirement of the system maintenance cost and effort required.

5. **Staff Development Plan:-**
→ Describe how the skills and experience of the project team members will be developed.

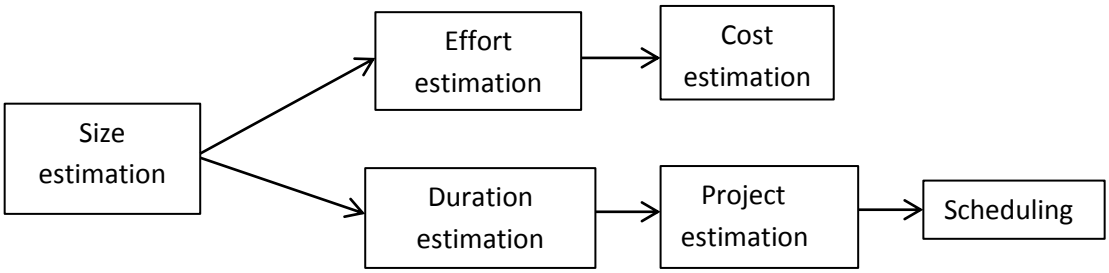


Fig: Software Project Planning.

➢ Software Project Estimation (Cost Estimation)

→ Introduction

→Software project estimation focus on cost and effort required to develop the software
→Different types of cost estimation techniques such as Delphi cost analysis and **COCOMO** model can be used.

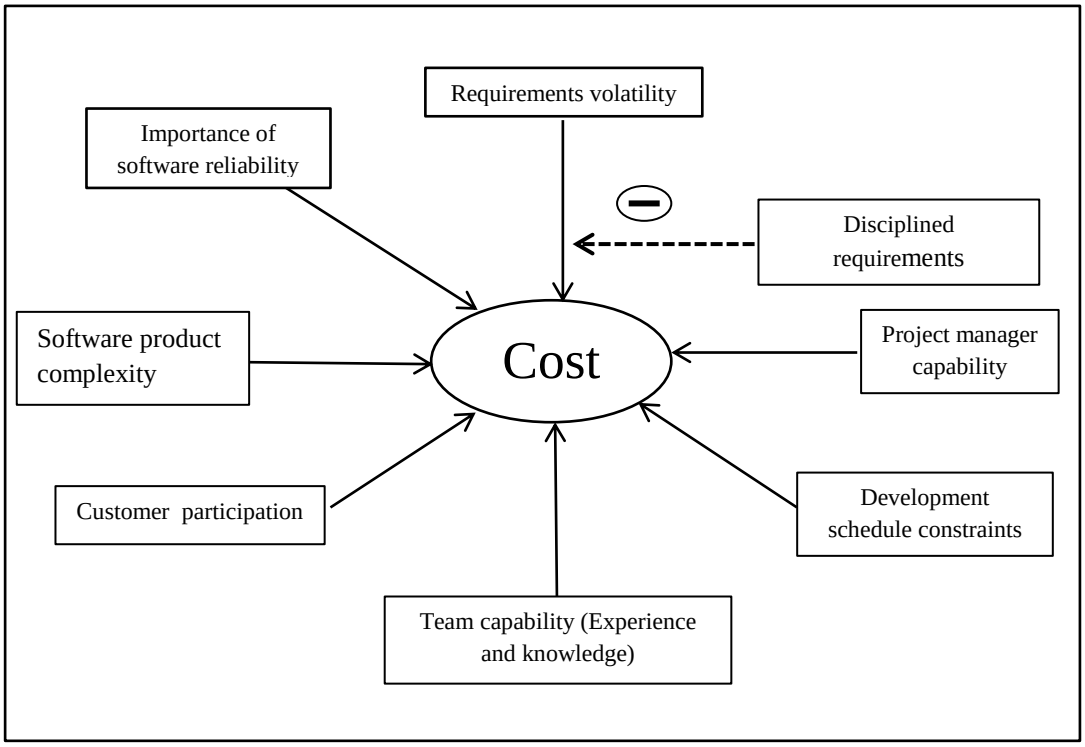
→ Project Estimation

→In early way of computing software cost of a small percentage of the overall computer based system cost.

→Today software is the most expensive development of computer based system.

→Software cost will never be an exact science.

→However to many variables like human, technique, environmental, political can affect the ultimate cost of the software and effort applied to develop it.



Different Methods of Estimation

1. Delphi Cost Estimation:

→The Delphi cost technique was developed by Rend corporation in 1948, to gain expert without introducing the adverse side effect of group meeting.

→The Delphi technique can be adapted to software cost estimation in the following manner:

1. A Coordinator provides each estimator with the system definition document and a form for recording a cost estimate.
2. Estimators study the definition and complete their estimates itself and they may ask questions to the coordinator but they don't discuss their estimator with one another.
3. The coordinate prepares and distributes a summery of the estimators response, and includes any unusual rational noted by the estimators.

4. Estimators complete another estimate, again itself , using the result from the previous estimates. Estimators whose estimate differs sharply from a group may be asked, itself, to provide justification for their estimates.
5. The process is iterated for as many rounds as required . No group discussionism allowed during the entire process.

2.COCOMO Model (Cost Constructive Model).

→ The COCOMO is a software cost estimation model , developed by 'Barry W. Boehm'.
→ The model uses a basic recreation formula , with parameters that are derived from historical project data and account project characteristics.
→ COCOMO was first published in 1981 by Barry W Bohem book software engineering economics as a model for estimating effort cost and scheduled for software projects.
→ This model can calculate the range of project size from 20000 to 10,000 lines of code.
→ Inn 1997 COCOMO 2 was developed and finally published in 2000 in the book "Software cost Estimation with COCOMO-2"

COCOMO Hierarchy.

There were three types of COCOMO Model.

1) Basic COCOMO.
2) Intermediate COCOMO.
3) Advance COCOMO.

1. **Basic COCOMO (Model-1)**.

The basic COCOMO model computer development effort as a function of program size expressed in estimated line of code.

2.**Intermediate COCOMO (Model-2).**

The intermediate cocomo model computer software development effots as a function of program size and a set of cost drivers(attributes) that include accesment of products, hardware , personal and project attributes.

3. Advanced COCOMO (Model 3).

→ The advanced cocomo model in corporate all characteristics of the interface version and assignment of cost drives impact on each step of the software engineering process.

- **Basic COCOMO Model.**

→ Basic COCOMO software development effort and cost has a function of program size.
→ Program size is expressed in estimated thousands of line of code (KLOC).
→ COCOMO applies three classes for software project.

1 Organic project.
2 Semi-detached project.
3 Embedded project.

→ The Basic formula.
→ **Effort applied :-** $a_b(KLOC)b_b[(men)(Months)]$
→ **Development time:-** C_b(effort applied) $d_d(Months)$
→ **People required: -** effort applied/development time.

Software Project Scheduling Management.

→ Spilt project into task and estimate time and resource required to complete each task.
→ Organize tasks concurrently to make optimum use of work force.
→ Dependents on project managers experience.
→ Estimating the difficulty of problems and productivity is no of people working on a task.
→ The unexpected always happens always allow schedule plan to the project.

- **PERT.**
- **Bar chart & Activity Network.**

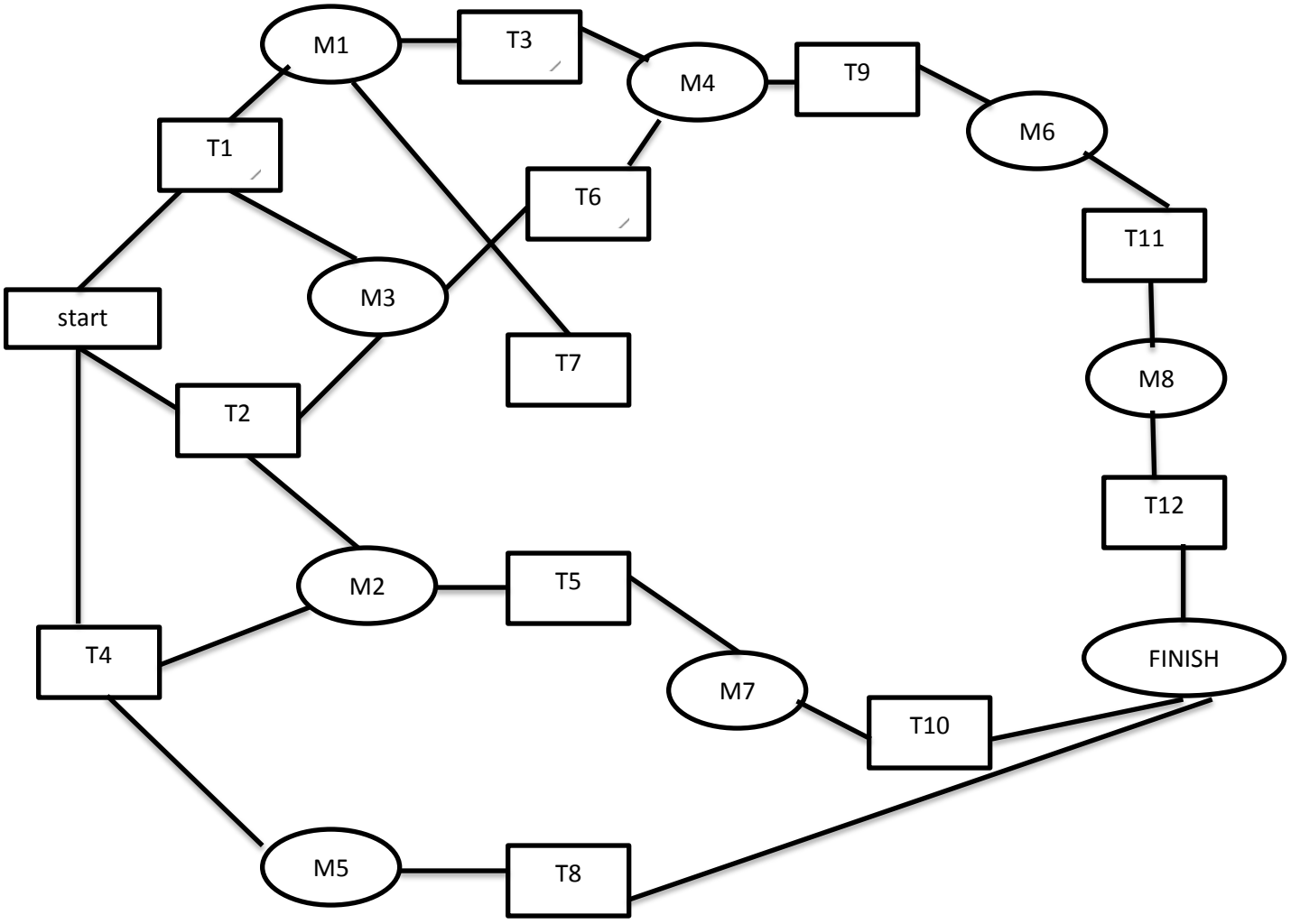
PERT (Program evaluation & review technique):

- Pert is a model for project management design to analyze and represent the task invoke in completing the given project. It is commonly used with critical path method (CPM).
- A pert chart is a project management tool used to schedule, organize co-ordinate task within a project.
- PERT stands for program Evaluation & Review Technique in 1958 to manage the Polaris submarine missile project.
- A similar methodology the critical path method CPM which is developed for project management in the private sector.
- **Activity network:**

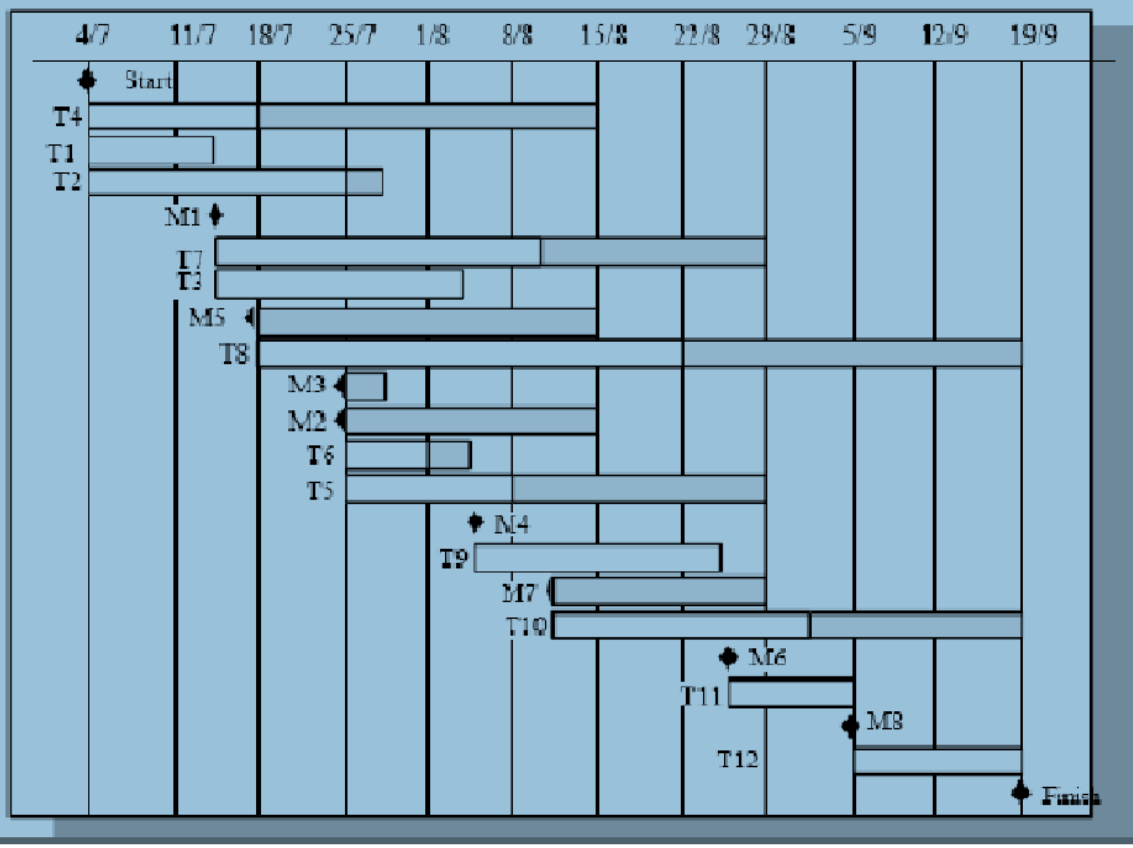
→ **Task duration and dependencies**

Task	Duration (Days)	Dependencies
T1	8	-
T2	15	-
T3	15	T1(m1)
T4	10	-
T5	10	T2,T4(m2)
T6	5	T1,T2(m3)
T7	20	T1(m1)
T8	25	T4(m5)
T9	15	T3,T6(m4)
T10	15	T5,T7(m7)
T11	7	T9(m6)
T12	10	T11(m8)

ACTIVITY NETWORK DIAGRAM



Activity timeline



Risk management

- Risk management is concerned with identi risks and drawing up plans to minimise the effect on a project.
- A risk is a probability that some adverse circumstance will occur.
  - Project risks affect schedule or resources
  - Product risks affect the quality or performance of the being developed
  - Business risks affect the organisation developing or pı the software

Software Risk

Risk	Risk type	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organisational management with different priorities.
Hardware unavailability	Project	Hardware which is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule
Size underestimate	Project and product	The size of the system has been underestimated.
CASE tool under-performance	Product	CASE tools which support the project do not perform as anticipated
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

Quality Assurance: -

• What is Software Quality:

Quality:

The American Heritage Dictionary Quality as “a characteristic or attribute to something”.

Quality Assurance:

Quality assurance is the process of defining how software quality can be achieved and how the development organization knows that the software has the required levelof quality.

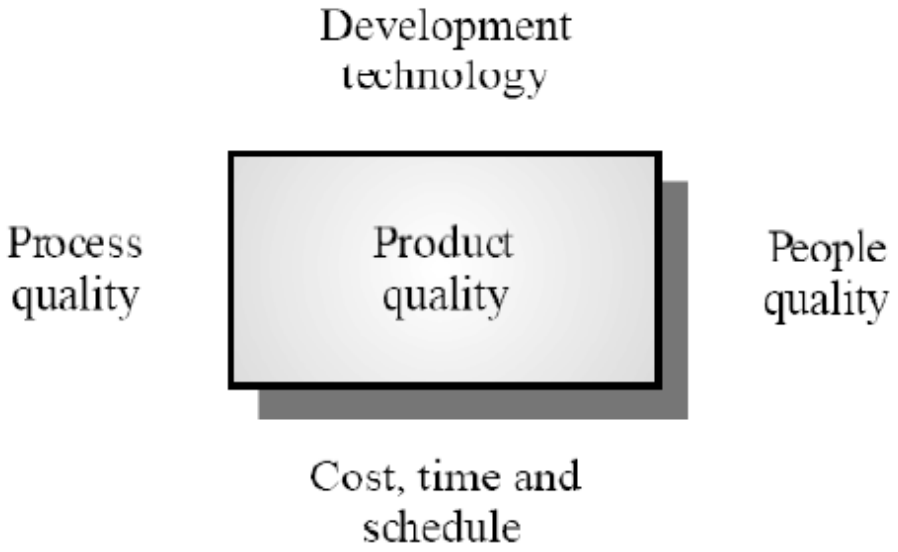
- Quality assurance refers to a program for the systematic monitoring and evaluation of thevarious process of the project, service or facility to the standards of quality.
- Two key principals characteristic quality assurance “Fit For Purpose” (the product should besuitable for particular purpose) and “Right First Time” (Mistake should be misplace and(handled), quality assurance includes regulation of the quality of the raw materials, assemblies,products and components service related to the production and management, production andinspection process.

Software Quality:

- Quality is a customer's determination not an engineer's determination, not a marketing determination.
- It is a based on the customer's actual experience with the product or services, measure against
his or her requirements.
- Product and service quality can be defined as "The total composite product and service characteristic of marketing, engineering, manufacture and maintenance through which theproduct & service and use in meet the acceptance of customers.
- Product can be designed & turned out to give satisfaction at a prize that user will pay. This is not easy & as soon as one fact feels fairly successful in the activities, he finds that the needs of the consumer have changed.
- Qualities that is defined as a matter of products & services whose measurable character satisfya fixed specification.
- Quality that is identifying independent of any measurable characteristic, that is quality define as the products or services capability to meet customer's acceptance.

→ Principle product quality factors.

Principles product Quality Factors



- **Software Quality Models.**

1. Six Sigma:

→ It is the most widely used strategy for statistical quality assurance in industry today. Originally

→ popularized by Motorola in the 1980's, the Six Sigma strategy "is a rigorous and disciplinedmethodology that used data and statistical analysis to measure and improve a company's

→ operational performance by identifying and eliminating 'detects' in manufacturing and servicerelated processes. The term "six sigma" is derived from six per million occurrences implying an extremely high quality standard.

1. Define.
2. Measure
3. Analyze
4. Improve
5. Control
6. Design
7. Verify

The SEI CMM MODEL:

- The software engineering institute (SEI) has developed a model that is predicted on a set of software engineering capabilities that should be present as organizations reached different levels of process maturity.
- LEVELS OF CMM (capability maturity model)
  1) Initial
  2) Repeatable
  3) Define
  4) Managed
  5) Optimized

1) Initial (Level 1):

→ The software process is characterized as adhoc,occasionally created.
→ Few processers are defined are defined, and success depends on individual efforts.
→ Initial level is essentially uncontrolled .

2) Repeatable(Level 2):

→ Basic project management processes are established to track post schedule and functionality.
→ The necessary process decipline is inplace to repeat earlier successes or project with similar application.

3) Define(Level 3):

→ The software process for both management and engineering activities is documented, standardized and integrated into and organization wide software process.
→ All projects used a documented and approved version of process for developing and maintain software.
→ This level include all characteristics defined for level 2.

4) Managed(Level 4):

→ Detail measure of the software process and product quality are collected.
→ Both the software process and products are understood and controlled using detail measures.
→ This level include all characteristics defined for level 3.

5) Optimized(Level 5):

→ A continuous process improvement is enabled by quantitative feedback from process and from testing innovative ideas and technology.

→ This level include all characteristics defined for level 4.

❖ IOS(International Standard Organization)

→ It is also called organization for standardization.

— **History:**

Formation	23,Feb,1947
Type	NGO(Non-Government Organization)
Purpose	International Standardization
Head Quarters	Geneva, Switzerland
Membership	165 Member(Contries)
Official Language	English,Russian,French.

→ **ISO 9126 Quality Factor**

→ The ISO 9126 standards where developed in attempt to identity quality attributes for computer software. The Standard identified six key attributes.

1) Functionality

→ The degree to which the software statics to need as indicated by the following sub attributes like suitable accuracy security.

2) Reliability

→ The amount of time the software is available for use as indicated by the following sub attributes.

- 1) Maturity
- 2) Fault tolerance
- 3) Recoverybility

3) Usability

→ The degree to which software is easy to use as indicated by the following sub attributes like operability, learn ability.

4) Efficiency

→ The degree to which the software make optimum use of system resources as indicated by the following sub attributes like time behavior resource behhaviaors.

5) Maintainability:

→ It is easy with which software may be repaired as indicated by following sub attributes like maintenance, stability, testing.

6) Portability.

→ The asy eith which software can be transformed from one environment to other has indicates by following sub-attributes like adapatability,replacebility,installability.

➢ **4P's project Managements.**

1. people
2. product
3. process
4. project.

1**.People.**

→ The first P stands for 'people '.it is a universal trouth engineer is making software for some other people to use it.

→ As we know there are various types of users for example,Manager,senior manager,hands on manager,indirect user,end user.

→ Every software project is populated by people who fall within this chassification.

→ Project team must be organization in a way that maximize each persons skill.'

→ And abilities ,the job of team leaders

→ The people management maturity model define the following key practice areas for software people like recruiting , selection , performance management ,carrier development program ,training work design, cultural development

2) <u>Product:</u>

→ The second p stands for 'product'. It indicates the software scope in terms of context, information, function used, and performance that finalized the products need to be fulfilled.

→ Problem decomposition or problem identification focus is on functionality to be delivered and the process used to deliver it.

3) process:

→ The third P stands for 'process'.

→ Process model must be appropriate for the developers and customers characteristics of product & project is based on process

→ There is a wide range of process models ,like

1) Prototype model
2) Agile model
3) Spiral model
4) Waterfall model
5) Iterative model (incremental model)
6) SDLC model (system development life cycle)
7) RAD (rapid application development model)
8) JAD model (Joint Application Development or Agile model).

→ Once the process model has been chosen then we have to work with that particular model, and we cannot change very easily.

→ Project manager ask "how do we accomplish this comman process frame work activity."

4.Project:-

→ The fourth p stands for project to manage a successfulk software project we must.

1. Start on the right tool.
2. Maintain tasks.
3. Track progress.
4. Make Smart decisions.
5. Conduct a re-checking analysis of the project.

→ The W_5HH Principle.

→ W5HH PRINCIPLE WEREV INTRODUCED BY "Berry W Bohem" according to him you need an organizing principle that have to provide3 simple plan for simple projcets.

→ Thus, by following this prinicile we can know many facts about projct and charcterriststis.

1. **WHY IS THE SYSTEM BEING DEVELOPED ?**

→ DOES the business purpose justify the expenditure of people time & money?

2. **WHAT WILL BE DONE?**

→ Identify the functionality to be build.

3. **WHEN IT WILL BE ACOMPLISHED?**

→ Establish a work flow and timeline for key project task and identify milestones required by the customer.

4.Who is responsible for a function?

Define member's role and responsibility.

5. Where are they located? (organizationally)

→ Customer also have responsibilities and positions for example, in company there are so many location department wise like sales department, purchase department, logistic department, hr department, marketing department, quality department, finance department.

→ Employee has post for their particular work like CEO, MD, ZM, PM, PL and analyst, designer, programmer, tester etc.

6: How will the job be done techniqually and managerially ?

→ Once a scope is defined a technical strategy must be also defind.

7. How much of each resorce is needed ?

→ The answer is derived developing estimates on earlier questions.

Unified Modeling Language (UML)

→ UML, as the name implies, is a modelling language. It may be used to visualize, specify, construct,
→ and document the artefacts of a software system. It provides a set of notations (e.g. rectangles,
→ lines, ellipses, etc.) to create a visual model of the system. Like any other language, UML has its
→ own syntax (symbols and sentence formation rules) and semantics (meanings of symbols and
→ sentences). Also, we should clearly understand that UML is not a system design or development
→ methodology, but can be used to document object-oriented and analysis results obtained using
→ some methodology.

→ **Origin of UML**

→ their object-oriented designs
→ In the late 1980s and early 1990s, there was a proliferation of object-oriented design techniques
→ and notations. Different software development houses were using different notations to document. These diverse notations used to give rise to a lot of confusion. UML
→ was developed to standardize the large number of object-oriented modelling notations that existed
→ and were used extensively in the early 1990s. The principles ones in use were:
→ • Object Management Technology [Rumbaugh 1991]
→ • Booch's methodology [Booch 1991]
→ • Object-Oriented Software Engineering [Jacobson 1992]

→ • Odell's methodology [Odell 1992]
→ • Shaler and Mellor methodology [Shaler 1992].

2.**UML Diagrams.**

→ UML diagrams represent two different view of system model.

1.Static view:-(Structural view)

- Emphasized static structure of the system using objects attributes operations and relationships.
- The structure view includes class diagram and composite structure diagram.

2.Dynamic View:-(Behavioral View)

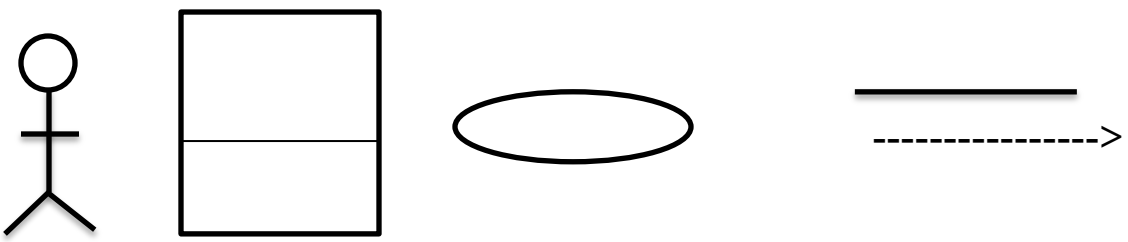
- Emphasize the dynamic behaviors of the system by displaying collaborations among objects and changes to the interstates of the objects.
- This view include sequence diagrams , Activity diagram state machine diagram.

→ UML Diagrams

1.Use case diagram:-

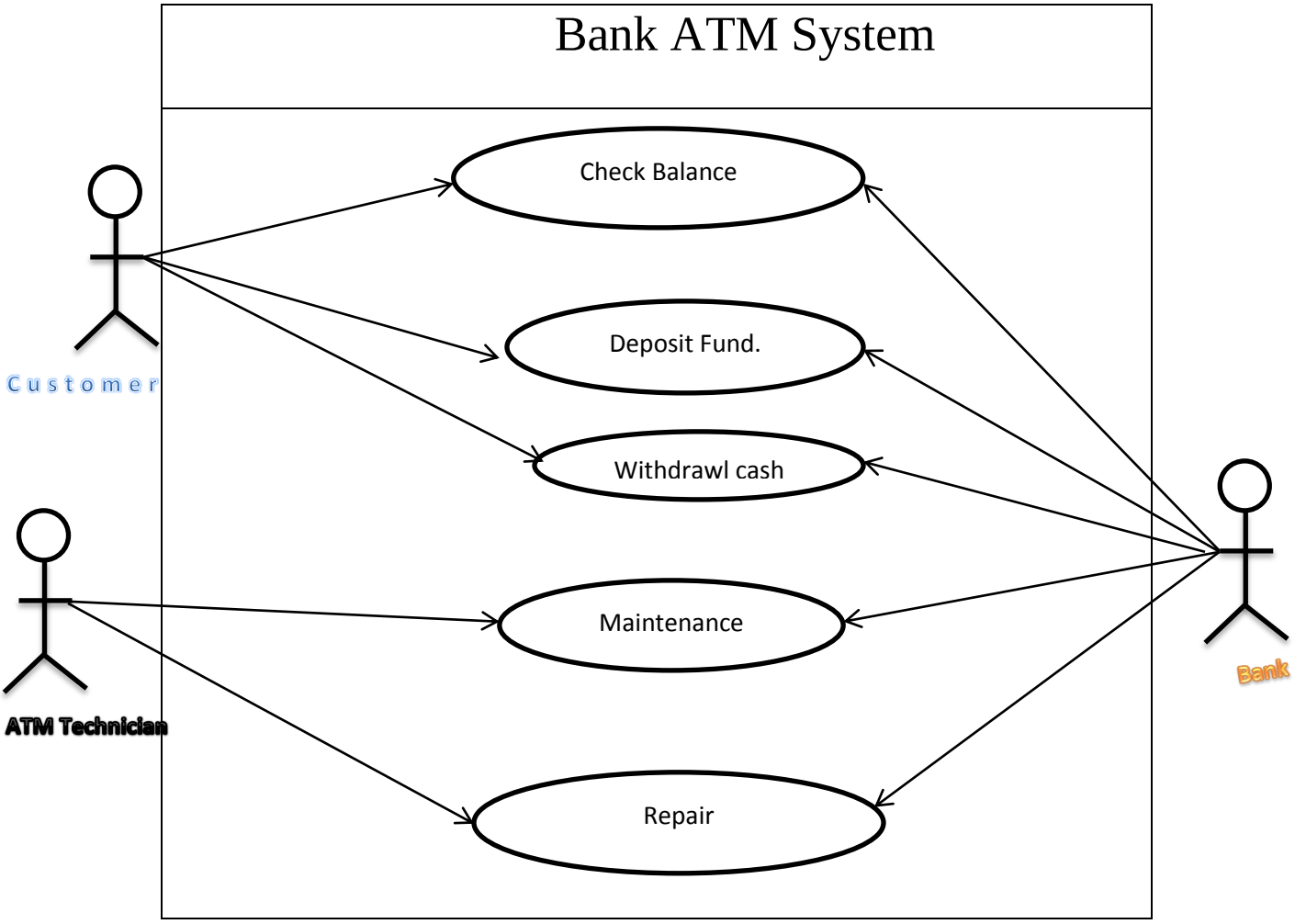
→ Behavior diagram include the use case diagram .(used by some methodologies during requirements gathering).
→ Use case diagram display a set of use cases, actors & their relationship.
→ This diagram should be used to model the context or the requirement of a system
→ It contents use case actors dependency association , relationship, generalization roles constraints , packing and instances.
→ sThe use case diagram make systems , sub-systems and classes by presenting an outside view of how the elements.May be used in context.

→ The notation for use case diagram is given below

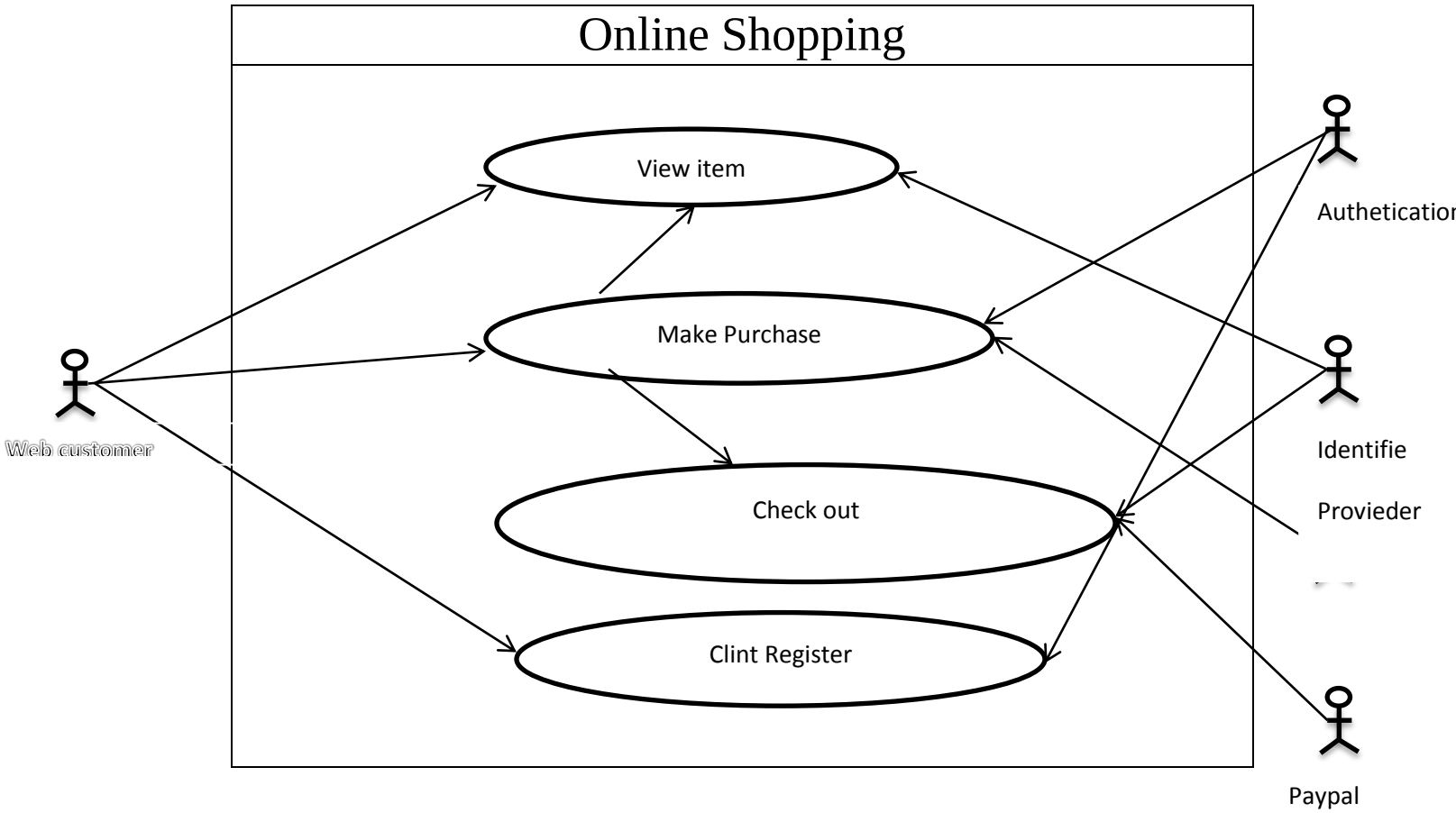


Actor System use case <<extend >> flow lines

→ Example: - Bank ATM System.



Online Shopping.



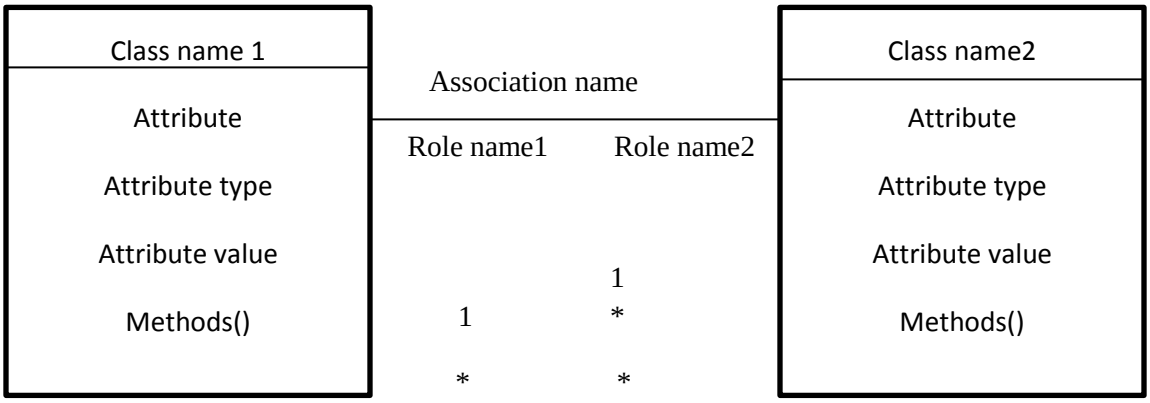
2.Class diagram.

→ Class diagrm include in structure diagram .object diagram diagram , package diagram and deployment diagram include in the structure diagram.

→ Class diagram describes the structure of a system by displaying the system class , their attributes and the relationship amount the class.

→ Functional requirements of system are demonstrated through this system. It contents set of classes interphase collaborations , dependency ,generalization and association relationship.

→ The notation for classes and the relationship between class is shown below.



Notations:

Generalization:- A extends B , A is derived from B.

Realization:-

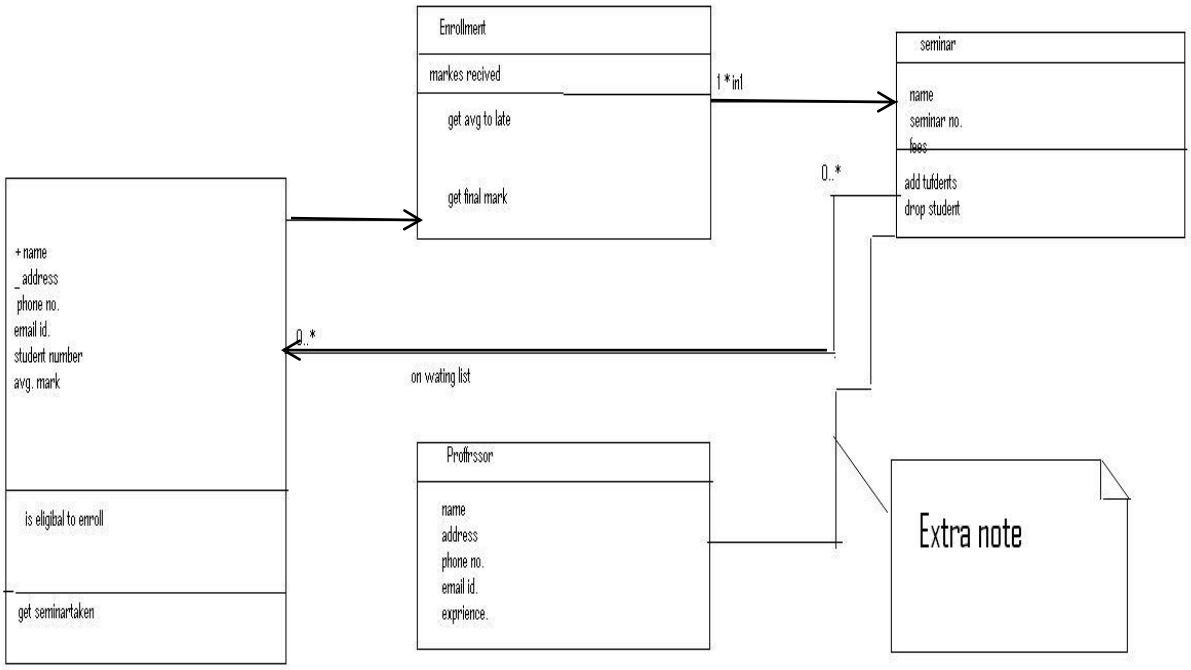
A implementation B, A is derived from abstract B, B is the definition for implementation A, B is an interphase & A provides the operations in B.

Dependancy:-

Association:-

A has access to some B , but B does not have access to A.

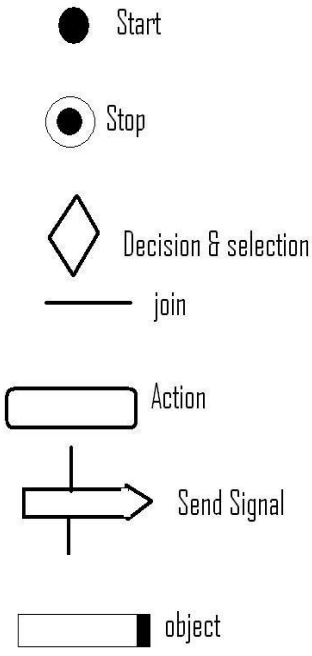
Example: - STUDENT MANAGEMENT SYSYTEM.

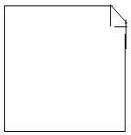


3. Activity diagram:

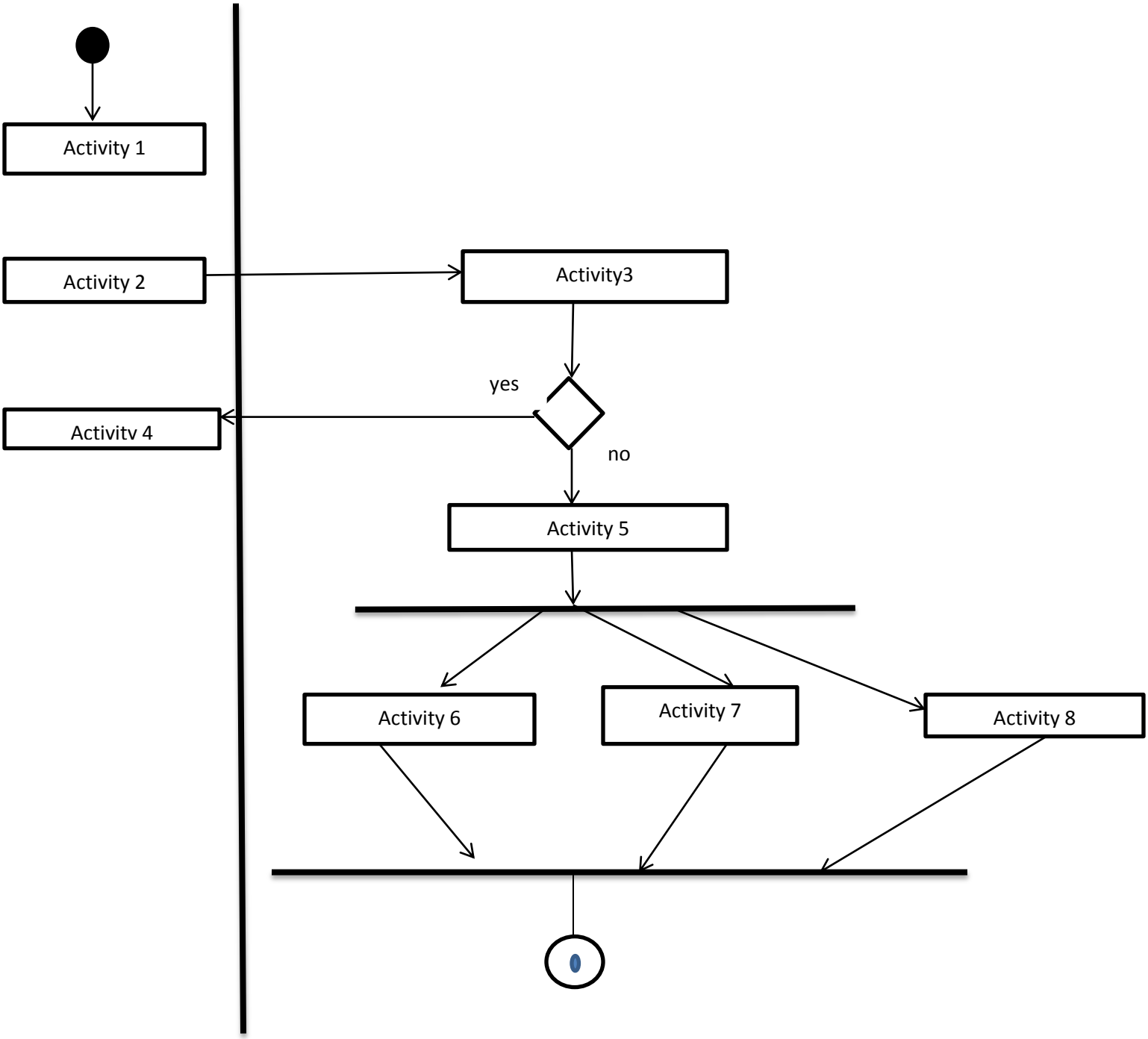
- → Activity diagram display the flow from one activity to another and activity is on going non-atomic execution within a state machine.
- → Activity ultimately results in some action which is made up of executable comutations that result in a change in state of the system or the return of a value.
- → It want activity sates, action states, object.

❖ **Notations:**





Example:-


Activity 1
Activity 2
Activity3
yes
Activitv 4
no
Activity 5
Activity 6
Activity 7
Activity 8