69 69

13-05-2019 CSC 413 Dr TOYIN

SOFTWARE ENGINEERING I

1. Attendance
2. Assignment
3. Formal Test
4. Brief Presentation
5. Examination
6. Class Uniform / Dress Code

COURSE OUTLINE

* Introduction to the Concept of Software Engineering ✓
* Project planning ✓
* Requirement Engineering ✓
* State transition ✓
* State Chart ✓
* E-R Models ✓
* Software Validation ?
* Software testing ?
* Software reengineering ✓
* Software patterns ✓
* Software re-use ?
* Open Source development ?
* Aspect Orientation

3-05-2019 Software Engineering

Software is considered as a collection of executable programming codes, the libraries and their documentations

When software is made up of specific requirements, we call it a Software Products

Engineering is about developing products using well stated processes and principles and methods (Scientific methods as usually contained)

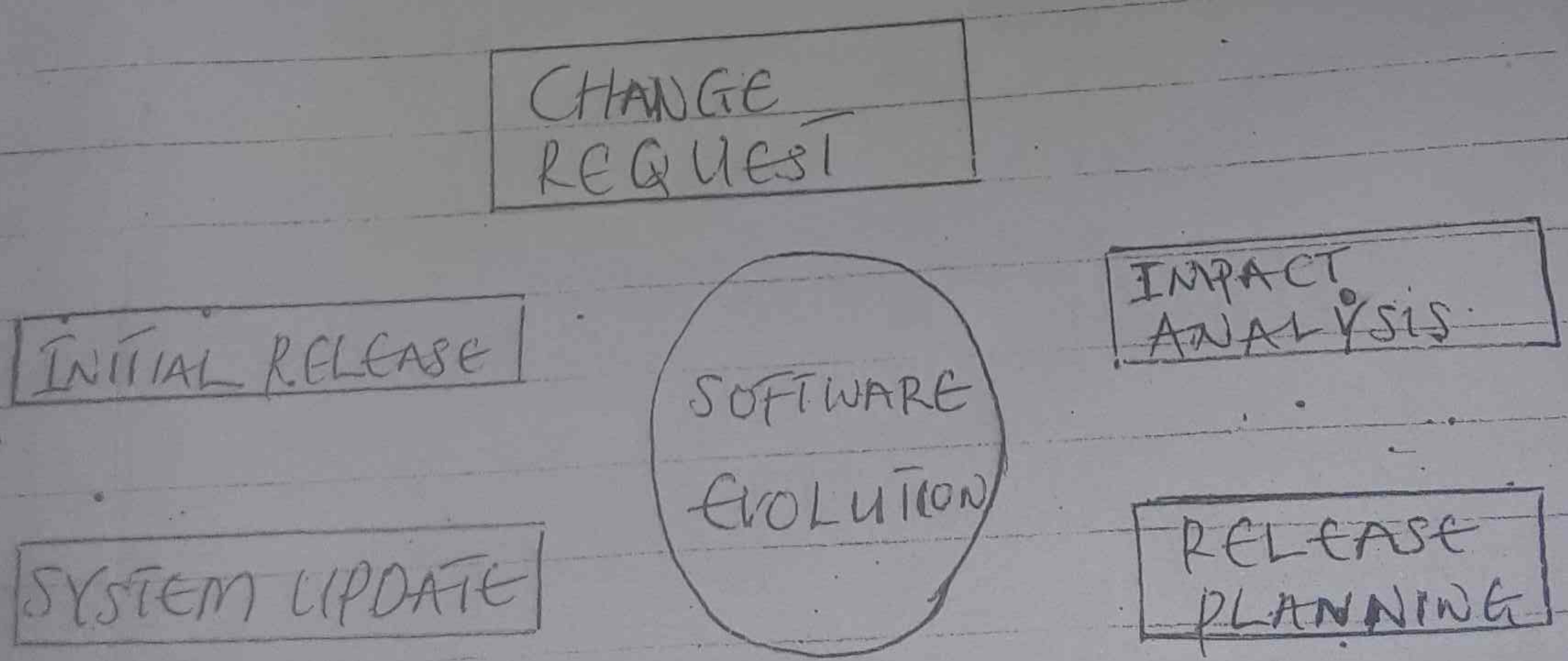
Software Engineering is the collection of executable codes, libraries and documentation as a product with specific requirement using scientific methods, principles and processes.

Software Engineering is an engineering branch concerned with the development of products

In the earlier concept of software engineering it was ~~required~~ defined by FRITZ BAUER as the establishment and use of a sound engineering principle in order to develop an economically usable software and work efficiently on machines.

A Software product can be identified when the following are possible: UPDATE, REQUIREMENTS, SYSTEM DESIGN, DEPLOYMENT, SYSTEM ANALYSIS, SYSTEM TESTING.

The process of developing a software using software engineering principles and methods is called Software Evolution. It includes the initial development of software, it's maintenance and update which satisfies the expected requirement.



- Requirement gathering

Change Request will change the Initial Request until user's satisfaction is most

Software Evolution laws

Lehman's Evolution law

1) S-type law (State-type) -

This is a software which works strictly according to defined specification and solution the solutions and the method to achieve are immediatly understood before coding.

- The S-type is subjected to changes. For example Calculator program for mathematical computation is an example of S-type.

2] P-type law (practical type)

- These are softwares that consists of collections or procedures. They are defined by exactly what procedures can do.

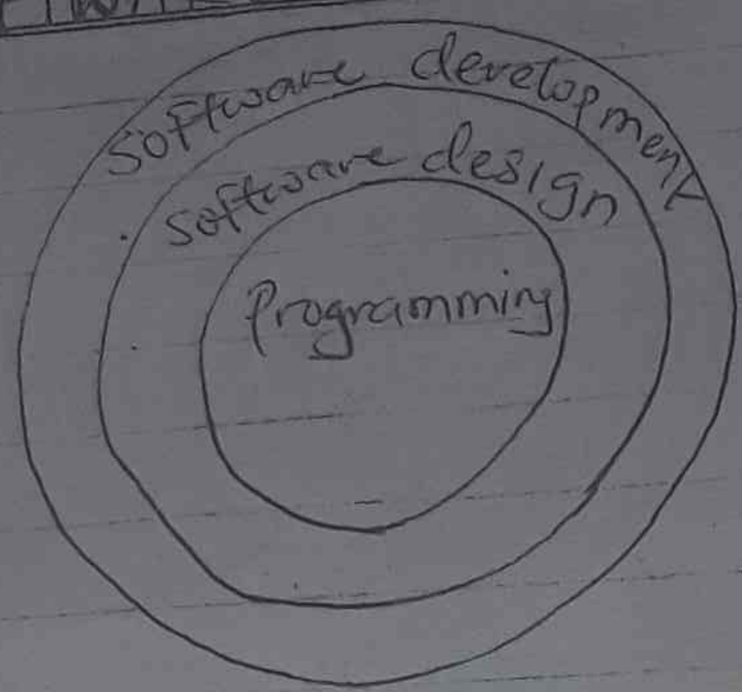
3] E-type law (Embeded type)

These are software that have a high degree of evolution as these are numerous changes just like the real world

There are 8 laws for the E-type Software Evolution

1] It must continue to change
2] It will have increasing complexity
3] Conservation of reality
4] It will continue to grow
5) E-type declines in quality except for rigorous maintenance
6) It consists of multi-loop, multi level, Feed back system
7] Self regulatory
8] It must have organisational stability.

SOFTWARE DEVELOPMENT PARADIGM



Software Engineering is essential because of the increase in software need, as well as a result of increase in software.

* **Scalability** : Building a new software by Scaling the old software

* **Cost** : Hardware manufacture increase specification

* **Documentation** : For keeping Records

* Quality Management

CHARA

1) Operational
2) Transitional
3) Maintenanc

1) operationa
- It must b
Functionali

2) Transit
- Portable

3) Mainte
- Modu

CHARACTERISTICS OF A GOOD SOFTWARE

1) Operational
2) Transitional
3) Maintenance.

1) Operational
- It must be able to be measured on budget, usability, functionality, security, dependability.

2) Transition.
- Portable, re-usable, adaptable, inter-operatability

3) Maintenance
- Modularity, Maintainability, Flexibility and Scalability

-05-2019 **SOFTWARE ENGINEERING & THE WEB**

The development of Software engineering has led to serious diversions in the way we execute our daily life processes. This has led to tremendious improvement in the method of solving our problem.

Previously, the **WEB** was a universally accepted information storage and accessment product without much intervention in software processes. The software ran on local computer and also only accessible on a local area network. Around year 2000, the web began to evolve and many other services were added to it. This means that web based system could be developed leading to the vast creation of many software domiced on the web. According to the several software running on the web have been at different times interchanged as web services. Web services includes but not limited to software running on the web.

Note ÷ A portal is not a Software but a web services

Pro
Software
run on th
several
Be
monolithic
or depend
were loc
Scratch
Price
and in
narra

1) Re
constr
assem

2) It
requ
web
ma

Principally, we may address the web services as software components that delivers specific output and run on the web. A web service is a combination of several web based software component.

Before the web, business application are mostly monolithic applications that run on the local computer or defined and restricted cluster. Then, communication were local within an organisation.

Business applications are not programmed from scratch but evolve by continuous change in components and procedure. This led to extensive re-use. The advancement and improvement in computer organisation changed the narratives.

THE NARRATIVES INCLUDE

1] Re-use has become dominant as an approach for constructing web based systems when building a system, it is assembled using pre-existing software.

2) It is obviously impractable to completely defines all the requirements of a system before it's development therefore, web based services should be developed in stages incrementally.

3] User interfaces are constrained to the web browser; Therefore, app interfaces of web based services are often poorer than the specially designed user interfaces on the PC

Software Engineering Ethics

Upholding standards by including honesty and integrity

1] Confidentiality: You must ensure the confidentiality of your client

2] Integrity: You must always show your client that you have high level of integrity

3] Competences

4] You must guide against intellectual property rights

Software En
Professional
Software
to making th
development
beneficial a
with their Co
welfare of

1. Software e
public inte

2. S.E act
employer

3. Software
products

4. Software
reputat

5. SE m
and p

Software Engineering Code of Ethics and Professional Practices

Software engineers shall commit themselves to making the analysis, specification, design development, testing and maintenance in beneficial and respected profession. In accordance with their commitment to health, safety and welfare of the public

1. Software engineers shall act consistantly to the public interest
2. S.E act consistently to the interest of client or employers
3. Software engineers shall ensure that their products meet the higher professional standard
4. Software engineers shall maintain integrity and reputation of public interest
5. SE managers and leaders shall subscribe to and promote ethical approach to the manage

6. Advance integrity and reputation of the profession

7. SE shall be fair to and supportive of their colleagues.

8. SE shall participate in life long learning in the practice of their profession

NB:- -> The institute of electronic and computer engineers

* In your own words, represent the professional code of conduct of ethics for

① ASSIGNMENT

1) For each of the clauses in ACM/IEEE code of ethics suggest an appropriate example that illustrates that clause

2) Based on your knowledge of some application types explain with example why they require specialized software engineering techniques to support their design and development

3] Explain how the Universal use of the Web has Changed the Software System

The Following is used as a generic example for the Software engineering Concept. This will Consist of the embedded system and a sensor based on data collection System.

CASE STUDY IN THE INSULIN PUMP CONTROL SYSTEM

An Insulin pump is a medical system that simulates the operation of the pancreas. The pancreas is an intial organ in human. The Software Controlling this System is an embedded System which Collects information from a sensor and controls a pump that delivers a Controlled dose of Insulin to the user. People who suffer from diabettes use the System. Diabetes is a relatively common condition where the human pancreas is unable to release sufficiently amount of hormones called insulin.

Insulin metabolises glucose i.e Sugar in the blood. The conventional treatment of diabetes involves regular injection of genetically engineered insulin. Diabetics

their blood depth sugal level using an external meter and then calculate the dose of insulin that should be injected.

The problem with this treatment is that the level of insulin required does not only depend on the level of blood glucose but also on the time of last insulin injection. This can lead to low level of blood glucose if there is too much insulin or it can lead to high level of blood sugar if there is too little insulin.

Low blood glucose can result in how memory in the short time and ultimately concosiousness are depth. In the long term, continual high level of blood glucose can lead to eye damage problem, kidney-damage and heart problems.

Currently advances in developing minaturized sensors have meant that it is now possible to develop automated insulin delivery system. This system monitors blood sugar level and delivered an appropriate those of insulin when required

In the future it might be impossible to have such system permanently attached to the body. Therefore, a software controlled insulin delivery system might

work by usi
to measure
proportiona
This is
controller

Th
it
pu
n

*

l meter work by using a micro sensor embedded in the pattern
should to measure some blood parameters that is
proportional to the sugar level
t the This is then sent to a pump controller the
om controller controls the computer
of
l of The sugar level and amount of Insulin needed
can it then sends a signal to the miniaturized
pump to deliver insulin via a permanently attached
miniaturized needle
mory
ure * FOR THE ABOVE CASE STUDY, DRAW A CORRESPONDING
of Flow diagram that shows the execution of the
Insulin delivery System
ize

21-05-2019 SOFTWARE PROCESSES

Process - Series of activities involved in the generation/Creation of a product

Sequence involved in the generation of a software product is a software process

A Software process must have the following attribute:

1] It must prescribe all activities be it major or minor

2] It uses resources to produce both intermediate and final products

* A software product without documentations is an Intermediate product

3] It may include sub products that have entry and exist points.

This means that it may include sub process that has entry and exit points

4] The activities are organised in sequence

When the process involves the building of a product use refer to the process as a life cycle (software life cycle)

In the software development process; the software cycle is made up of a coherent set of activities which include

1] Specifying
2] designing
3] Implementing
4] Testing the software system

A structured set of activities required to develop a software system is called a "software process". They allow for specification, design, evaluation and evolution

A "software process model" is an abstract representation of a process; it presents the description of a process from some particular perspectives.

There are are many software process model that are generic

1] The waterfall model
It has seperate ~~distl~~ distinct phases for each of the specification and development

2] Evolutionary development
Both specification and development and validation are interleaved

3] Formal System development
A mathematical model that is transformed to an implementation eg Java system

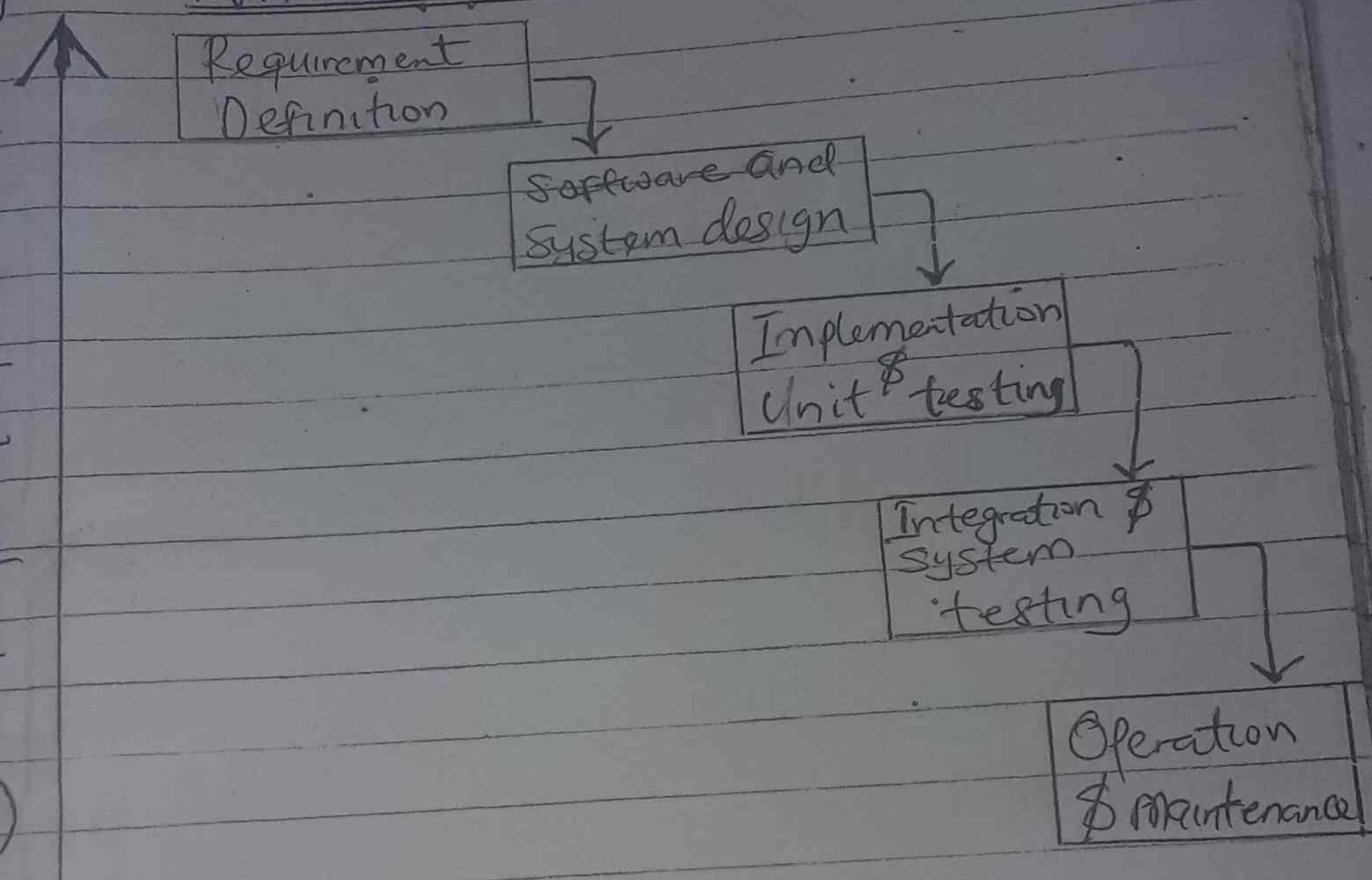
4] Re-use based
System is developed and assembled from existing component

1 R S I I O

req
def

1

WATERFALL MODEL

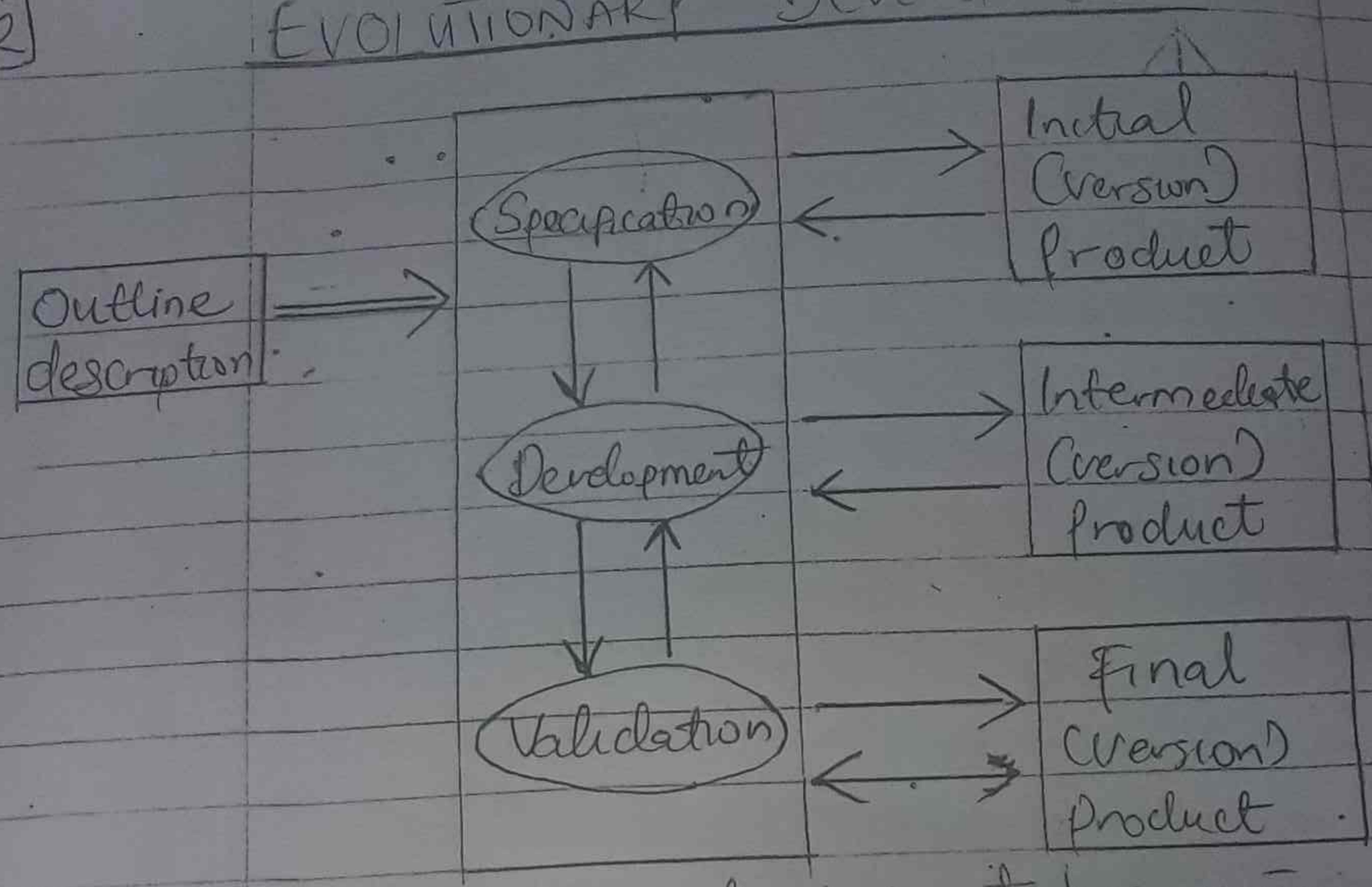


1] When a process begins, change is difficult.

2] Inflexible partitioning of projects into distinct stages.

Therefore, this model is only appropriate when the requirement are well understood. The water-fall model defines process of step wise refinement

2] EVOLUTIONARY DEVELOPMENT



The evolutionary development has a major objective to evolve a final product from an initial outline specification. It must start with well understood requirements. The system evolves by adding new features to existing outline (listed requirement) as demanded by the customer.

Throw-away prototyping is an other method in evolutionary development. It's main objectives is to understand the system requirement and deal with the poorly understood requirement

Advan
clarity Sy

1] LACK of Pro
2] Systems a
3] Special S

3] FORM
This
representatio
executable
Th
for
from requir
formal t

Require
definit

Advantage is that we can develop a quick and dirty system i.e WORST CASE SCENERIO

PROBLEMS

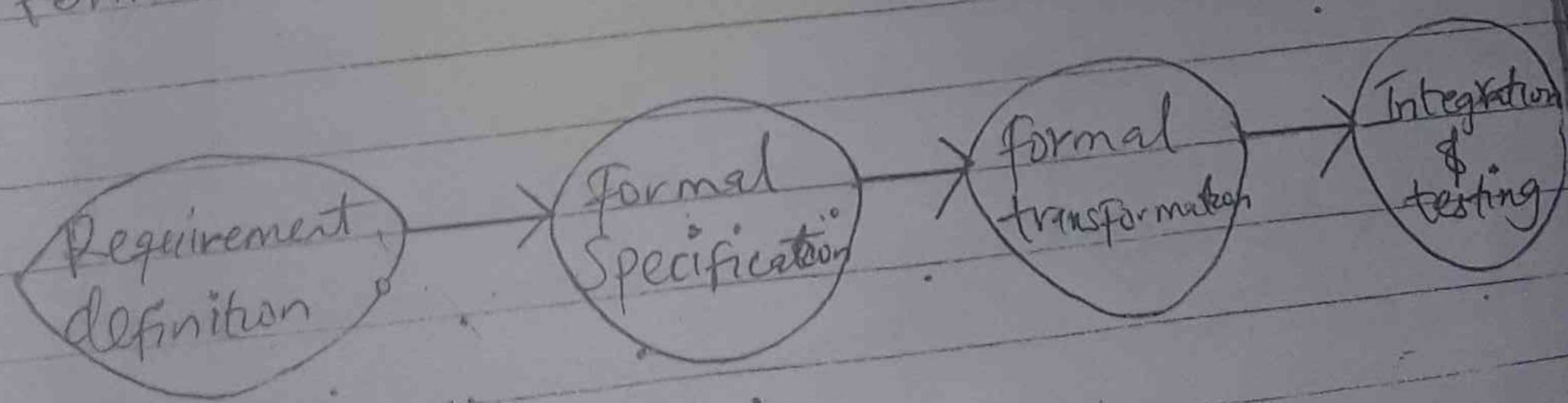
1] LACK of process. Feasibility
2] Systems are poorly structured
3) Special skills may be required.

3] FORMAL SYSTEM DEVELOPMENT

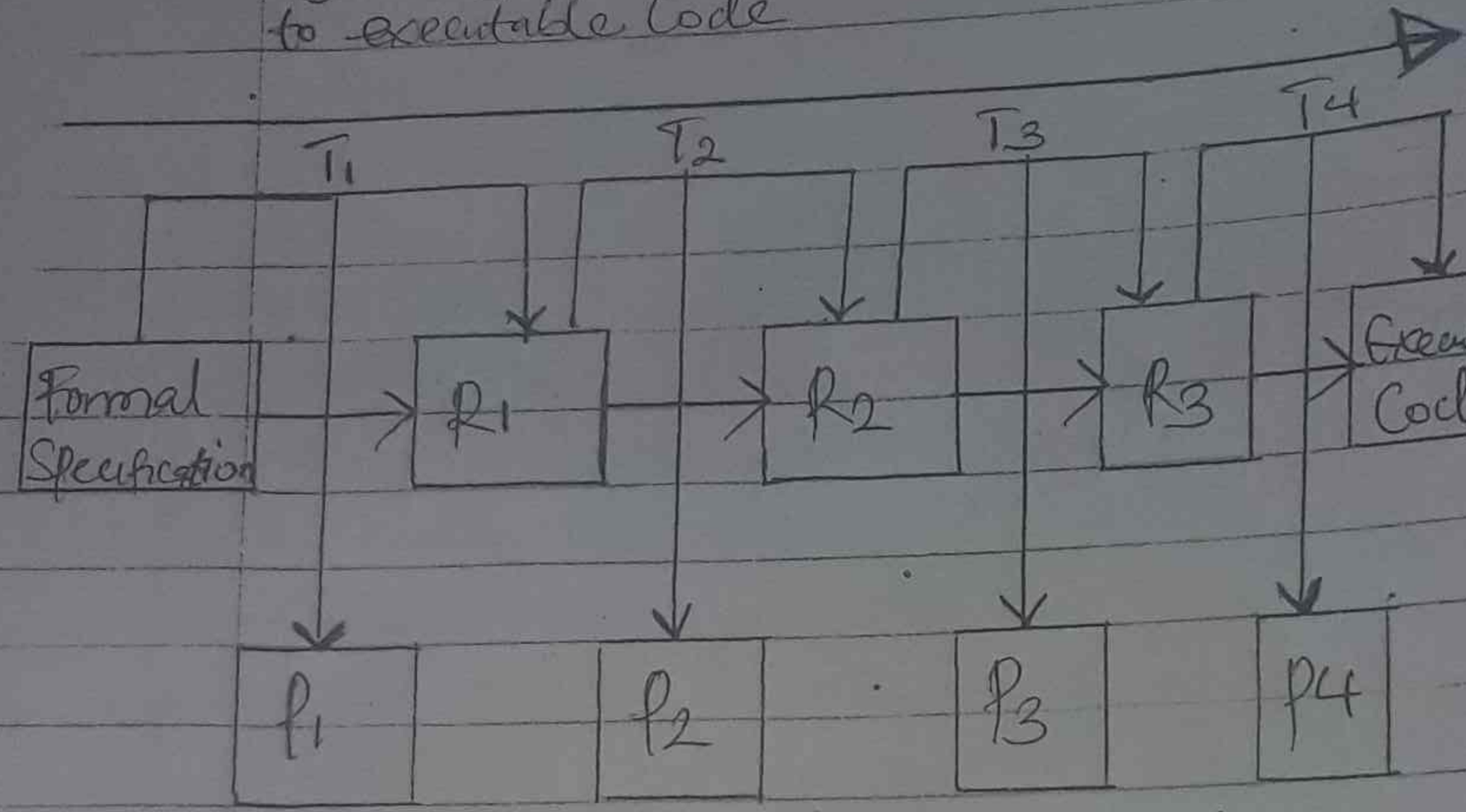
This is based on the transformation of a mathematical representation through different representation to an executable program.

The transformation are "Correctness preserving"

formal specification. System is a linear flow from requirement definition to formal specification to formal transformation to Integration & testing



This linear flow can be proven for correctness by having a linear flow from formal specification to executable code



key: T → Transformation
R → Requirement
P → Representation

Problem

1) Need for Specialised Skills

Re-use

This approach is becoming important but limited in experience. Whose Sytem are integrated from existing components or system, It is also a linear representation starting from.

Requirement → Component analysis → Requirement modification

→ System design and Re-use → development

Code and Integration → System Validation

PROCESS ITERATION

Modern development processes take iteration as important and tend to ~~risk~~ find ways of managing rather than ignoring the risk. System requirement always evolve at the _ _ _ _ of a project

1) Increamental development
2) Spiral development

1] **Incremental Development**

Rather than delivering the system as a Produ
the development and delivery is broken down
into components and deliver them in incremental
order. User requirement are characterized by
the highest priority.

Once the development OR an increment begins
the requirements are frozen

2] **SPIRAL DEVELOPMENT**

Process is represented as a spiral rather than a
sequence of activities with back tracking. Each
loop in a spiral represents a phase in the process,
there is no fixed phase such as specification and
design.

i.e the loops are chosen depending on what is required

Here risk are explicitly accessed and
resolved throughout the process

② . ASSIC
. * Use
SPIRA
mac
— OI
— R
—
— (

a Produ
down
emental
sed by

ent begins

than a
. Each
process,
tion and

required

and

② . ASSIGNMENT

. * USE A WELL DETAILED DIAGRAM to DESCRIBE SPIRAL MODEL of the Software development model

- Objecture, Setting
- Risk assessment
- planning
- development and validation

This is the process of converting the system Specification to an executable System.
Software design involves the design of a flow structure that realizes the specification
Design is about specification.

- Implementation is the translation of structure into an executable program

The activities of design and implementation are closely related and deeply interleaved

Design Process

- Architectural Design
- Abstract Specification
- Interface design
- Component design
- data structure design
- Algorithm design

③ ASSIGNMENT

TERM PAPER – Identify an organisation and do a heuristic report on how software engineering has affected the productivity of the organisation

GROUP ① Group leader → Michael Lieul

27-05-2019

Creo

The
descr
mean
The
for a
placin
a cl

doc
che
REQ

NB:- The

I
abst
proc

defi

is b

27-05-2019 SOFTWARE REQUIREMENT ENGINEERING

(REQUIREMENT ENGINEERING)

The requirement of a system are the description of what the system should do. This means the services provided and constraints, These requirements reflects the need of customer for a system that serves a purpose such as placing an order, finding an information controlling a device.

The process of finding out, analyzing, documenting and checking these documenting and checking these constraints and services is called REQUIREMENT ENGINEERING

NB:- The meaning of requirement is not consistent

In some cases requirement are high level abstract statement of service that a system should provide or a constraint on a system.

Requirement is a detailed function and definition of a system function

Requirement definition is the requirement that is being defined by a software developer

The extent to which a requirement can be defined

SOFTWARE SYSTEM REQUIREMENT are generally classified into two Categories

1] FUNCTIONAL REQUIREMENT
2] Non-FUNCTIONAL REQUIREMENT

1] FUNCTIONAL - REQUIREMENT :- Statement of Service that the system should provide it includes

a] How system should react to a particular input and
b] How system should behave in some situation
c] What services are required
d] Sometimes dictate what the system should not do

2] Non-Functional Requirement

These are majorly constraints on the system

- Development Constraints
- testing Constraints

- timing constraints
- Constraints improved by standards

Non-functional requirement often apply to the system as a whole rather than individual system features.

The distinction between the different types of requirement is not very clear sometimes and most usable during implementation

The functional requirement of a system describes what to do, which depends on the software developed and also depends on the general approach of the organisation that owns the software.

When expressed as user requirement, fundamental requirement can be explained in an abstract way, that can be understood by the users. However, more specific requirement describe the system function, input and output expressively

DOMAIN REQUIREMENT :- These are derived from a particular domain rather than specific need of a user

PROBLEM

Software Engineering may not understand the characteristics of this domain.

CASE STUDY

Why will an attendance System in LASU not be usable in LASUTH

ANSWER :- Time difference in lectures

i.e LASU - DAY (Taken attendance during the day When taking lectures but not at night)

LASUTH - DAY & NIGHT (Taken attendance during the day and also at night)

Non-functional requirement are not concerned directly with the system, they may relate to the emergent properties like

reliability
define co
in the int

Non-fu
architec
A Sin
may dem
Non
budget c
software

* RE

reliability, responce time, Automatively, the may define constraints. eg. Capability of I/O devices used in the interference with other system.

Non-functional requirement may affect the overall architecture of a system

A Single non-functional requirement such as Security may derive a number or functional requirement

Non-FR arise through User needs, because of budget constraints and Ability to interact with other sopturires, the organisational policy

* READ FOUR JOURNALS AND EXTRACT A SUMMARY Including : INTRODUCTION, literature, DATA, Experiment, Conclusion

3-06-2019 TYPES OF NON-FUNCTIONAL REQUIREMENT

1] PRODUCT REQUIREMENT:- Exist as requirement that discusses the constraints that exist in the behaviour of a software Examples:- Update Constraints (youtube)

2] ORGANIZATIONAL REQUIREMENT :- These are requirement that are derived from policies and procedures that comes with the relationship, between Customer and developer manufacturer

Example:- In a class where the lecturer is the developer and the student is the Customer the organizational requirement are, putting uniform in class

3) External requirements:- These are requirements that are derived from factors that are external to the system Example:- A quiet environment, Sanitation of toilet close to the Classroom. This may include Regulatory Requirement that set out what must be done for the item to be approved.

SOFTWARE REQUIREMENT DOCUMENT (SRD)

SRD Sometimes also called software requirement specification is a well specified list that shows what the developer is expected to implement. Generally, this document should contain the user requirement and all the specifications required of the systems.

Sometimes, the User requirement and specification of system as integrated into one & represented into the same case scenarious. If their large number of requirements a detailed requirement are presented in seperate document.

Requirement document is essential when a contractor is developing a software.

However, some developmental model argue that the developmental requirement changes so rapidly and requirement document may be out of it.

The requirement document has a device set of Users ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ to the engineers responsible for managing the system.

Requirement specification is the process of writing down system and user requirement in a requirement document.

The User and System requirement should be clear, unambiguous, easy to understand, complete and consistent

REQUIREMENT VALIDATION

RV is the process of checking that the document contains what the customer really wants

Requirements for a software is what the system should do and the constraints of Implementation

Functional Requirement are statement of services that the system must provide, description of how the system are carried out

Non-functional Requirement constraints that exist as the constraint system is being designed The might be organisation, product or external requirement.

The often relate to the emergent part of the system and apply the system

Software Requirement Document (SRD) is an agreed statement of the system requirement. It should be organised so that the system customer and developer can use

Software validation
Checking the system for validity, consistency, reliability

(4) ASSIGNMENT

I] Identify and briefly describe 3 types of requirement that may be given for a computer based system

II] DRAW a process model showing how a requirement review can be used in a bank ATM case study.

10-06-2019 SOFTWARE PROJECT PLANNING

PROJECT PLANNING

Project planning is a crucial part of software development life cycle

In every step or a SDLC there are several situation to make a decision

Feasibility Study - First level of SDLC

Feasibility study will identify the cost of producing a software

Project planning involves risk analysis and uncertainty analysis, analysis of current economies

In S.E decisions must be taken under uncertainties => risk is always involved

Interestingly, decisions taken are used to formulate a prototype.

In conclusion, the software engineering process begins with planning. These processes are crucial to the success of a software project

This is aimed at providing a frame work. This is aimed at helping project managers make estimate about cost, schedules and resources.

These three ingredients are required for the successful deployment of a Software

The resources include: the Software developer

PROJECT ESTIMATION: Is the first step in project planning: It helps to know the total amount of resources required for the project i.e Cost, Schedule

WAY TO MAKE ESTIMATE

- Collection of relevant historical data
- Experience

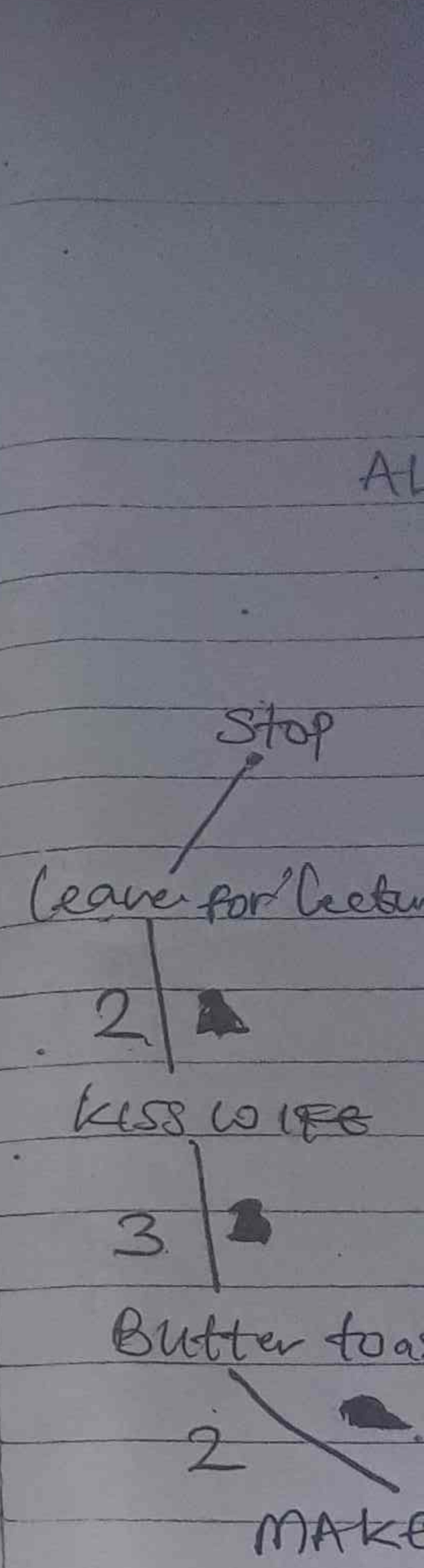
Estimation includes lots of Uncertainty that is encompassed in risk. There are so many activities associated with project planning

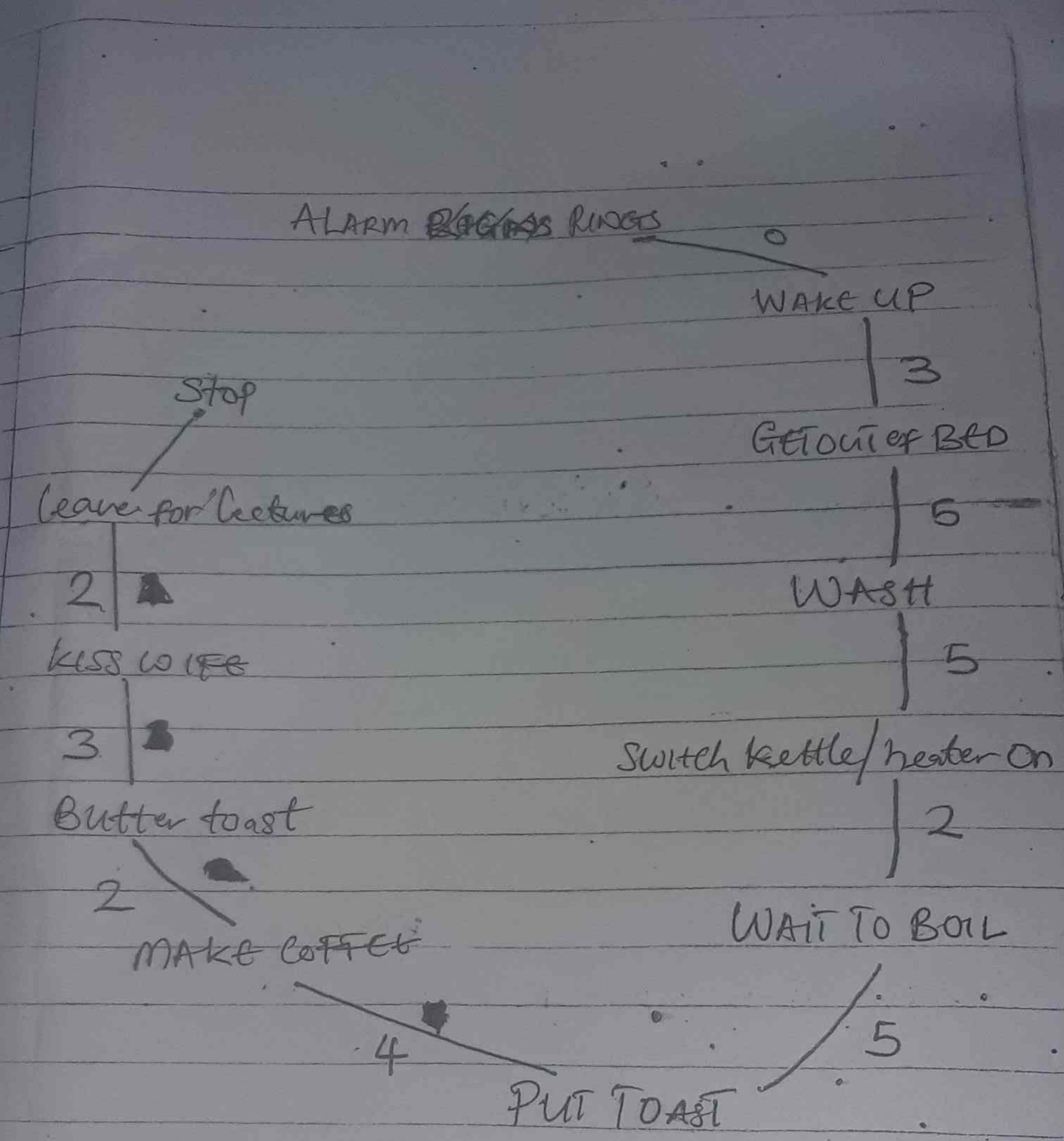
Processes involved in PLANNING

TASK	DURATION
1 Alarm Rings	0
2 Wake up	3
3 Get out of Bed	5
4 WASH	5
5 Kettle/Heater on	2
6 Wait to boil	5
7 Put toast	4
8 Make Coffee	2
9 Butter toast	3
10 Kiss wife	2
11 Leave for lectures	

This will be put in a "PERT CHART"

PERT CHART : A graphical network model used to depict the Interdependences between project tasks.




ALARM ~~BEGINS~~ RINGS
0
WAKE UP
3
GET OUT of BED
5
WASH
5
Switch kettle/heater on
2
WAIT TO BOIL
5
PUT TOAST
4
MAKE COFFEE
2
Butter toast
3
KISS WIFE
2
Leave for Lectures
Stop

Activities Associate with Project Planning

1) Software Scope.
2) Resources Estimation
3) Project Estimation
4) Decomposition.

SYSTEM MODEL

17-06-2019

System models are abstractions which describes a system whose requirements are being analyzed. System models are abstract description of a system.

The objective of learning system model include:

1] To explain why the system must be modelled as part of the Requirement (R.E) process i.e (Requirement engineering process)

2] To describe behavioural modelling.

3] To describe data modelling.

4] To describe object modelling.

System modelling helps the analyst to understand the basic functionality of the system in a way the customers. It is a form

Project Planning

of representation for the system output. This is essential because the customer is interested in the output and not the method of arriving at the output

Different models represents different perspective

The "external perspective" shows the system context via it's environment

"Behavioural Perspective" shows the behaviour of the system. "Structural perspective" shows the data architecture of the system.

All of these are due to have a distinct life cycle of the system

Disadvantages of SYSTEM MODEL.

1) It doesn't model the "non-functional requirement" i.e It can't give the general representation of the system

2) They will not tell when the model isn't suitable

3) System model can produce to much documentation

4) System Model leads to the generation of several and complex documentation that seemes too difficult for users to understand or comprehend.

TYPES OF SYSTEM MODEL

1) Data Processing Model :- It shows how the data is being processed

2) Composition model :- It shows how the component of our entity react with the component of another entity

3) Architectural model :- It basically shows the system

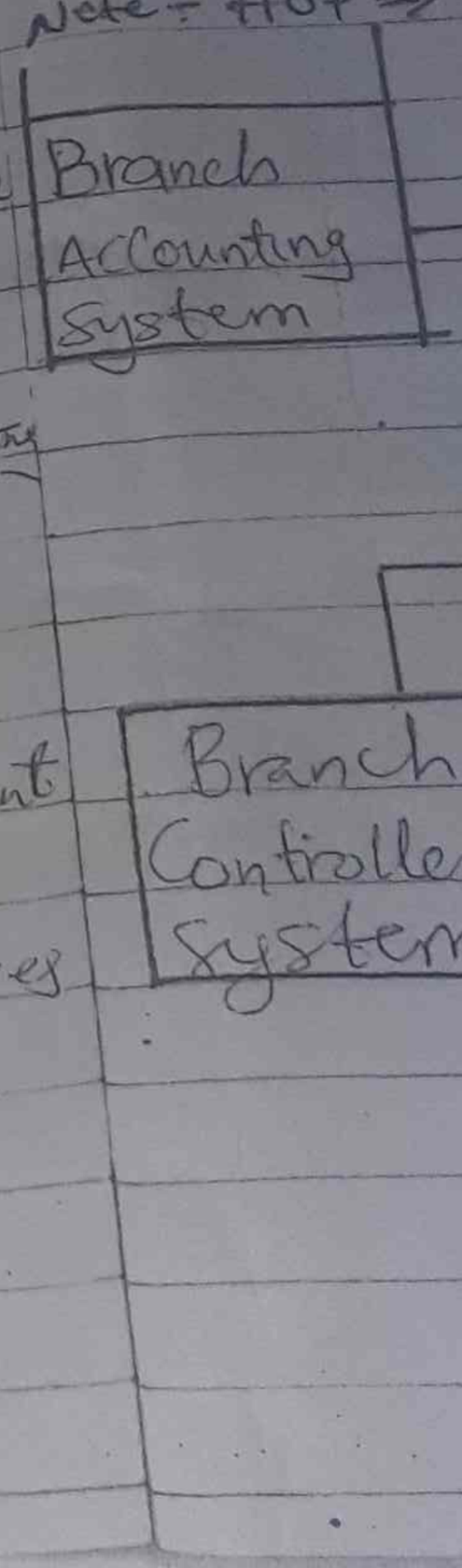
4) Classification Model :- This shows the ~~existing~~ entities having the same characteristics.

5) Stimulus or Response Model :- It shows the reaction of the system when there is an event

6) Context Model :- Illustrates the boundaries of the system.

Social and
effect
boundar
Arc
and it's
Conte

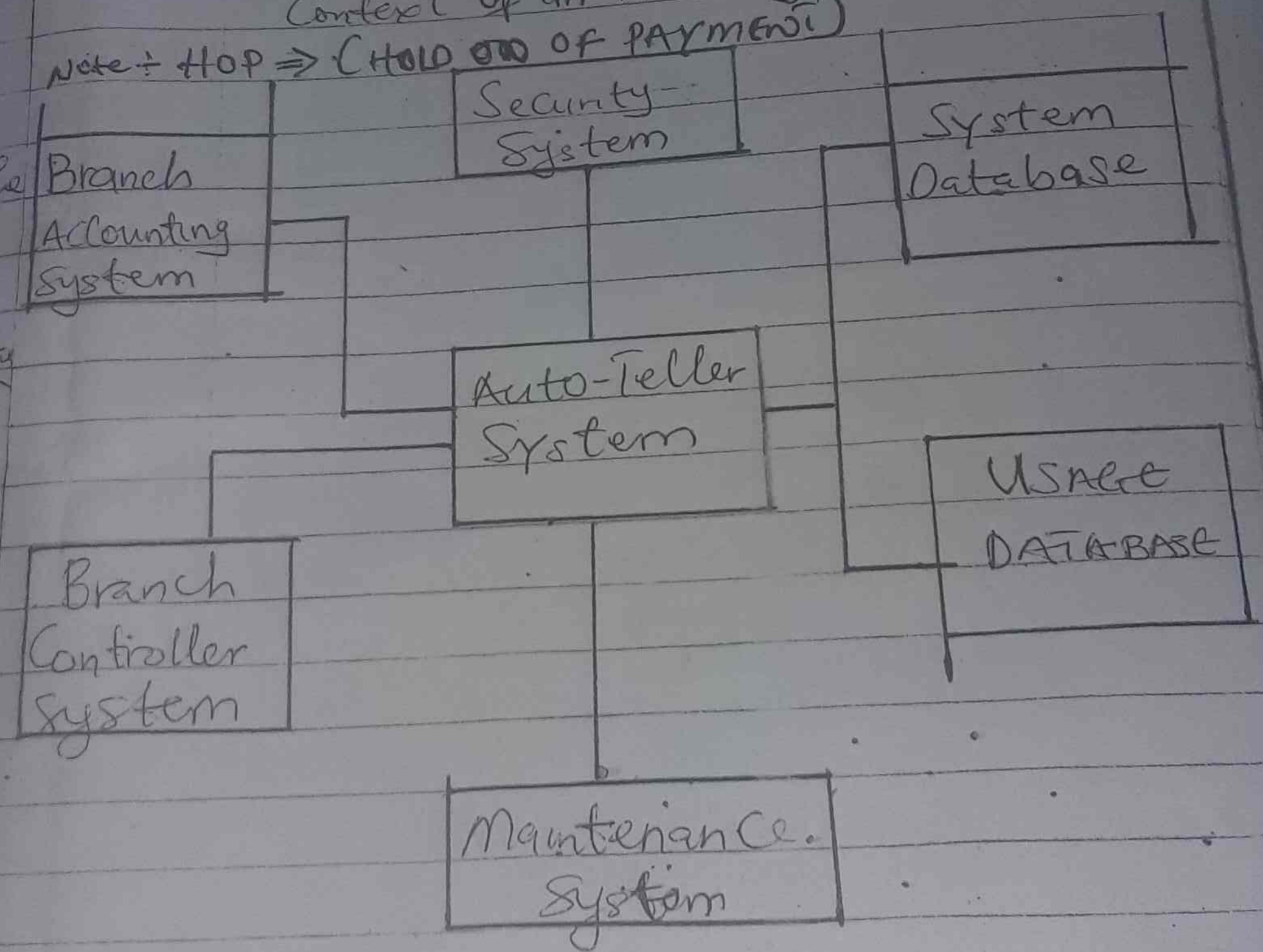
Note :- HOP ⇒ Ct



Social and organisational Concerns may affect the decision of what to put in boundary

Architectural model show the system and it's Relationship with other model Context of an ATM System

Note :- HOP ⇒ (HOLD ON OF PAYMENT)



Security System ÷ User Validation

Branch Accounting System ÷ Type of account [HOP]

USAGE DATABASE ÷ Log i.e activities of User

Branch Controller System ÷ Maximum amount of withdrawal maintenance.

Maintenance System Database ÷ Maintenance of the database of User

Auto Teller System ÷ Controls the whole process of the ATM model

The process model shows the overall process and supported by the system. The data flow model may be used to show the process of the flow of termination and the flow of operation from one place to another

Behavioural Model

1] Data Processing model ÷ Shows how data is processed as it moves through the system

2] State Mac
System

System

Test Ques
wr
op
C

19-06-

2) State machine system:- that shows the system response to event.
They are both required for the system behaviour

TEST QUESTION:- WITHIN THE BODY OF SYSTEM MODELS
Write short note on the conceptual framework of non-functional requirements as regard this body.

19-06-2019 DATA FLOW DIAGRAM

Data flow diagram are called DFD. The DFD model the system from a functional perspective, the track and document how the data is associated with a process
With the view to developed an overall understanding or the system. DFD shows the data exchange within system and other system in the environment.

STAR MACHINE MODEL.

SM models the behaviour of the system on response to external and internal events. They show the system's response to stimuli and are therefore used for modelling real time system

SMM shows system status as nodes and the event that occur between those two nodes as an "ARC"

When an event occurs, the system moves from one state to the other.

STATE CHARTS

SC are integral part of the UML.

Note that SMM doesn't show flow of data within the system.

STIMULUS AND DESCRIPTION FOR A MICROWAVE OVER

If stimulus is half-power user has pressed half-power on the microwave, If stimulus is full-power user has pressed Full power

TYPES

1) FINITE S
mach
Cuto
beha
with
or n
with

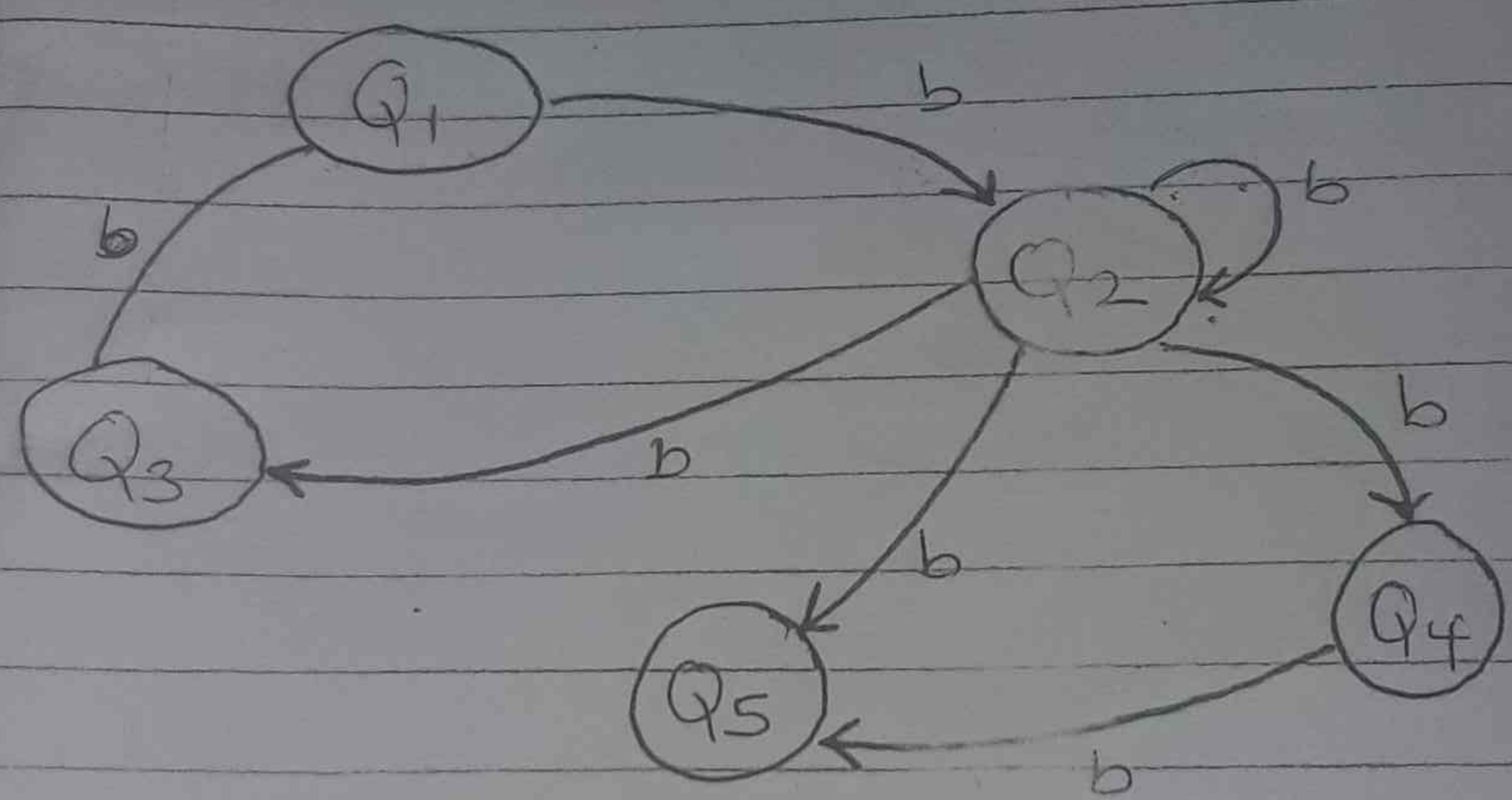
b

Q3

(a)
(b)

Types of SM

1) FINITE STATE MACHINES ÷ The finite state machine (FSM) is also known as finite state automata (FSA) They are model of the behaviour of a system or complex object with a limited number of defined conditions or modes where "mode transition" changes with circumstances



Complexity determines the number of modes Finite State Machine can be fromally defined as a model of computing consisting of

(a) A set of states
(b) A start state

(c) It must consist of an Input alphabet
(d) It must consist of an alphabet that maps
(e) ~~It must consist of an~~ activities of the current state to the next state

Computation begins in the Start state with an input string which then changes as the state changes. state change is determined by functionality

STATE — TRANSITION — movement of one state to another Rules and conditions must be met to allow a state transition
Input events are either internally or externally generated which will trigger rules of the state machine

1) Machines having output associated with output are called mealy/moore machines.
2) Machine with multiple start state Transitions are conditioned on input symbol with more than 1 transition of a given symbol

(5) ASSIGNMENT

USING AN Appropriate diagram, discuss the Mealy Machine as it produces output on a transition instead of on entry to a state.

Transitions are labelled i or O while i is the character in input alphabet while O is the character in output alphabet

PETRINETS

PETRINETS were developed by Carl Adam Petri in 1962, this concept has been extended and applied in multiple areas. It is a collection of different "arcs" connecting places and transitions. Places may hold tokens, the place of, not is it assignment of tokens to places

Every arc has capacity 1 as default

Places have infinite capacity as default

Transition has no capacity at default

Arcs can only connect places to transition and vice versa

(6) ASSIGNMENT 1
Show the mathematical properties of petrinet

24-06-2019 UML STATE CHARTS
Note: for every state there is a life cycle

State diagram is a diagram around a state
The state diagram depicts the data and the behaviour of a state throughout it's life cycle
In a basketball scenerio,

For a given set of states there must be an initial state.

1) A set of states (Initial state)
2) State transitions. (transition between states)
3) The diagram is drawn from an object perspecture.

Note → State, activity, sequence

Pseudostate

roperties of

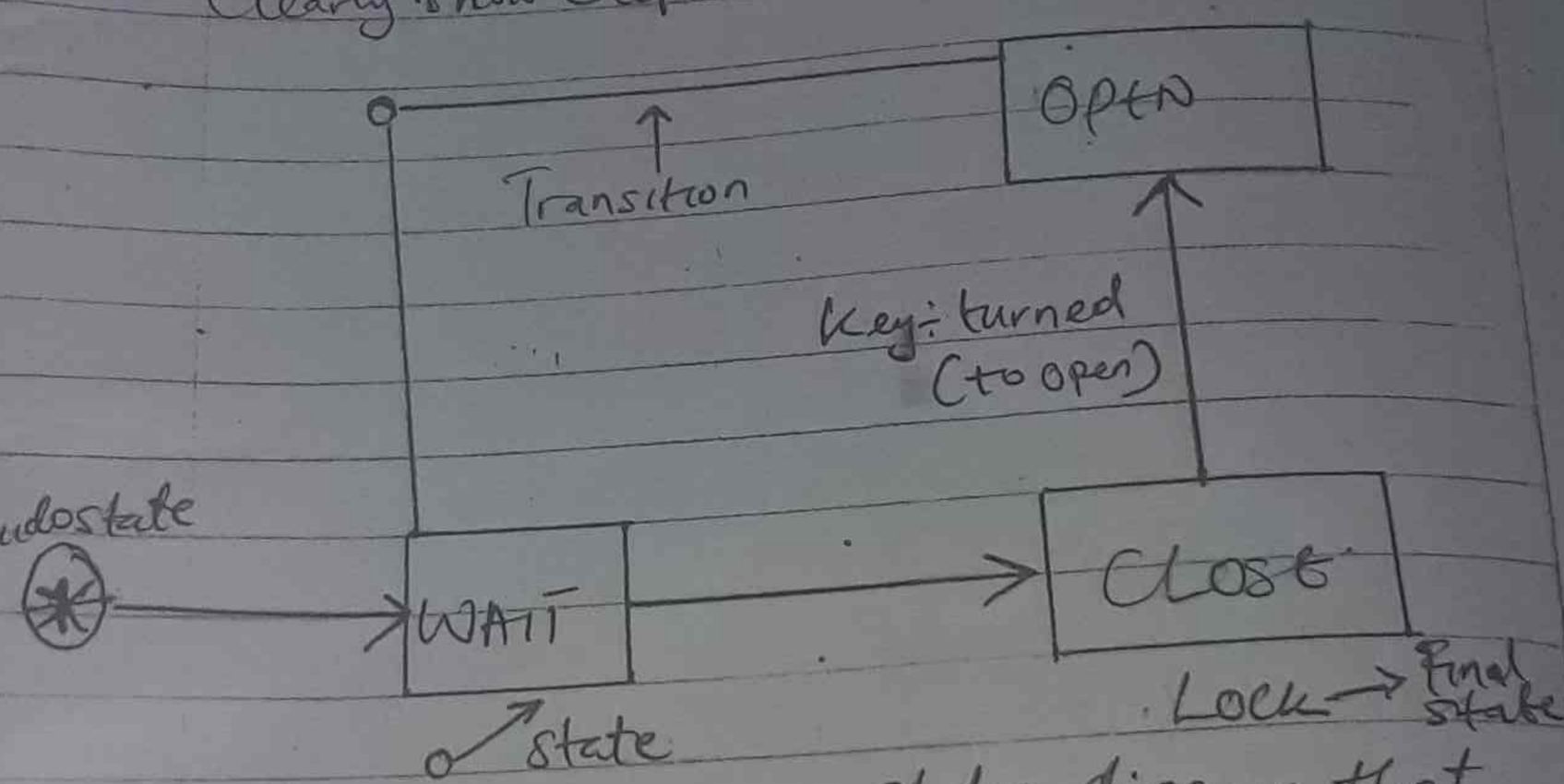
de

ound a state
and the
's life cycle

ust be

en states)
t

finite state machines: are machines that clearly show definiteness of states



Representation of a state diagram that shows when a safe is opened and locked

A state is a conceptual description of the data in the object, it is represented by object field values. The entire diagram is done for the central object perspective it only includes states

It doesn't include every possible value for the field only those which are conceptually

TRANSITION — Means movement from one state to the other

An Activity — Is defined as any behaviour executed during transition

Transition must be mutually exclusive. i.e it must be deterministic. It must be clear what transition takes to the event

Internal Activity — Actions that internal object takes on itself

Exit and Entry Activities — Are activities that cause the state to start or stop an action. When one state is complex, you can include a "sub-state" in it which is drawn as a nested rounded rectangle.

Note

STATE PATTERN

- A State pattern is an object whose sole purpose is to represent it's current state or it's configuration. Recall that in finite state machine, set of strings are the "nodes", set of edges are the "mutation" that causes a state to change.

STATE MACHINE DIAGRAM

A state machine diagram is a representation and a model that describes a scenerio of state diagrams. It represents the behaviour of a single object specifying the sequence of event that an object goes through throughout it's lifetime.

A door has three states * Open, Close, Lock * it can respond to four events

- Events of a door are ÷ OPEN, CLOSE, LOCK & UNLOCK

Note ÷ All events are valid in all states

A state transition can have a guard condition if the door is open it can only respond to close event

Note:- Unified modelling language

01-02-2019 DATA MODELS

* Importance of DM (Data Model)
* Building block of DM
* How major DM evolves
* Classification of DM

Data Models:- Data models are simple representation especially in graphical form, of complex real world system.

These real world systems are denoted by their data structures. Therefore, data models are representation of complex real world data structure. It encourages or facilitates relationship and interaction between "the system, the developer and the user.

Data
have different
will organi
There a
used to ge
Concepts
1] An Ent
to be Coll

2] An At
Data
entity.

"Relat
"Associa
relatio
— One
—. One

Examp
* Pas
* Re
* A

Data models are important because "End Users have different need for data. therefore, a data model will organize a "data based" on a specific need.

There are some building blocks that can be used to generate a data model. These building Concepts must be known

1] An Entity: Is anything about the data that is to be collected and stored.

2] An Attribute: Is also a basic building block of a Data model. It is simply the characteristics of an entity.

"Relationship": describes association between entities. "Associations" are crucial in determining the type of relationship that exist.

- One to many Relationship
- One to One Relationship

Examples of one to many Relationship

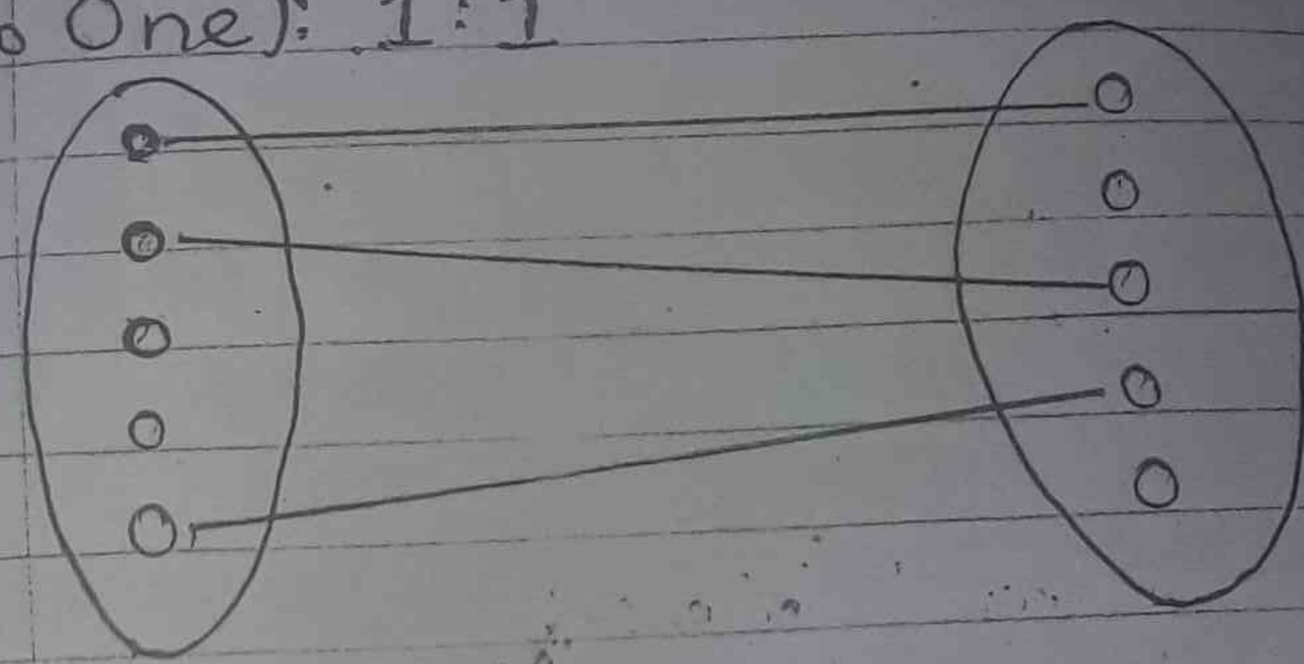
* Pastor to Congregation *
* Router to nodes
* A king to his people

Examples of One to One
* student to Supervisor
* VC to University

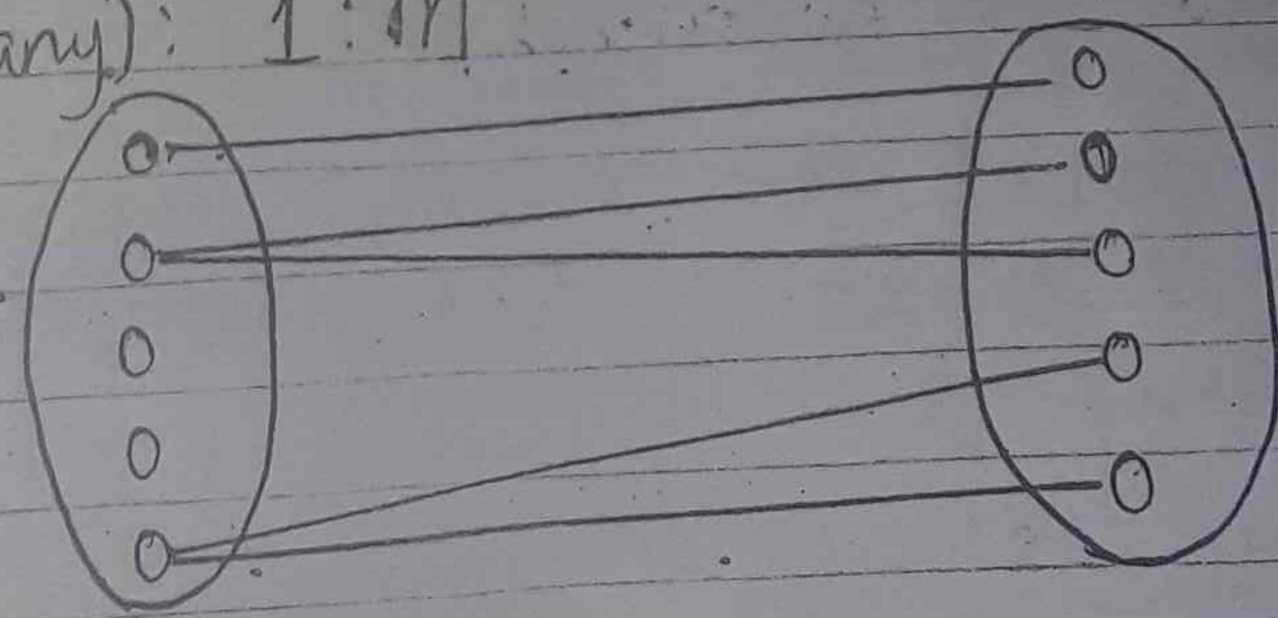
Examples of Many to many
* Books to Authors

DIAGRAM OF RELATIONSHIPS

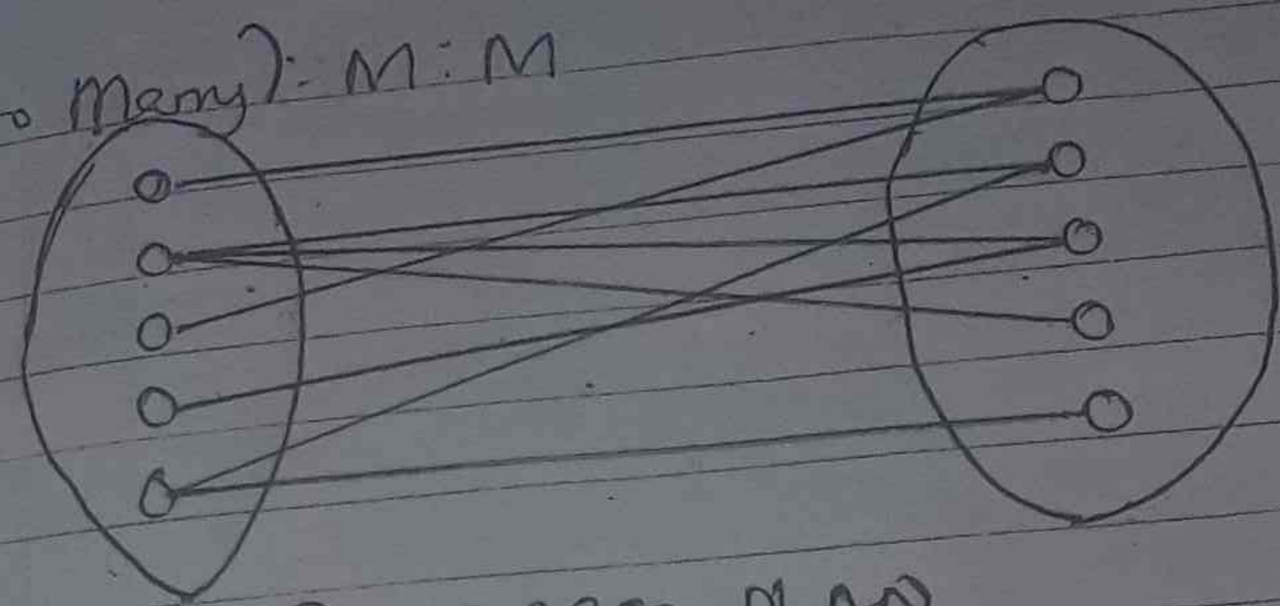
(One to One): 1 : 1



(One to Many): 1 : M



(Many to Many): M : M



BUSINESS PLAN

A "business rule" must be a brief, precise and unambiguous description of policies, procedures and principles within an organisation.

Business rule applies to any organisation that stores and uses data to generate information

Description of information that can help to generate actions

* Business rule must be rendered in writing.
* It must be kept updated
* It must be easy to understand and widely disseminated
* It must be clearly

HOW TO TRANSLATE BUSINESS RULES TO DATA MODEL COMPONENT

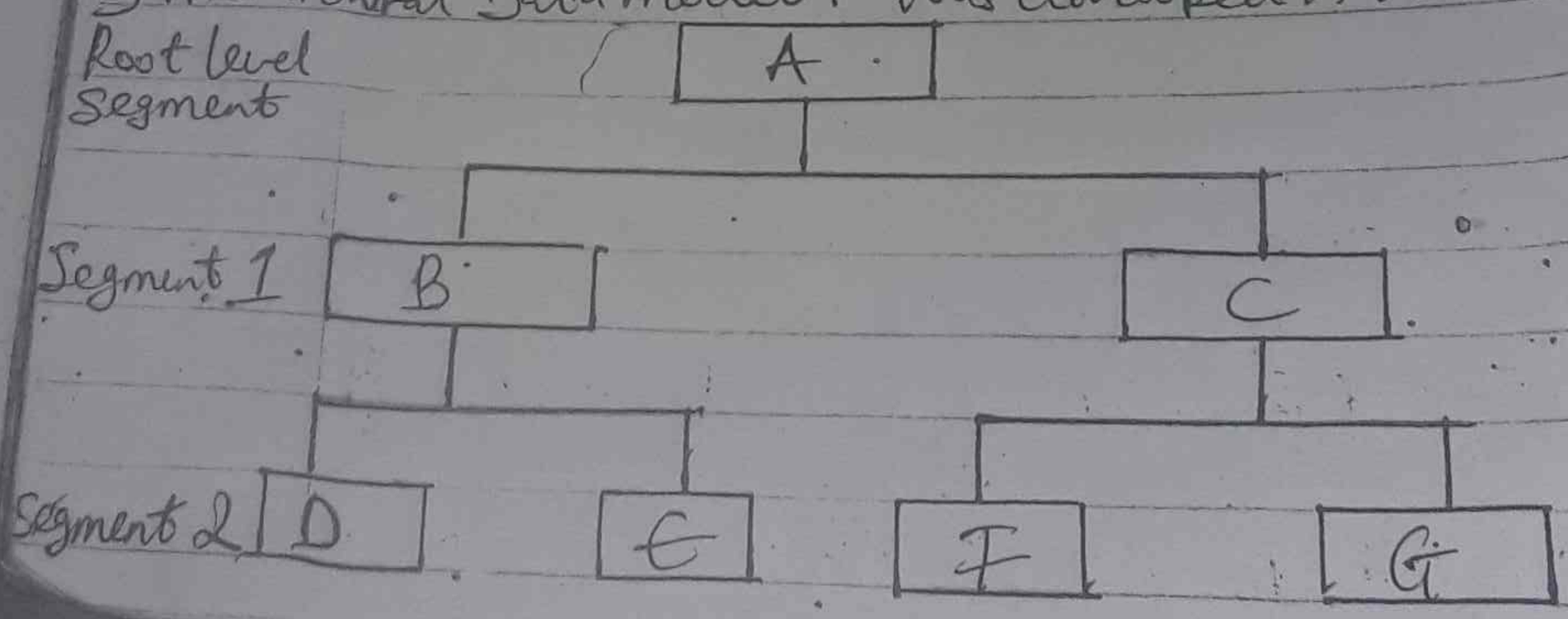
* By having a standardized Company's View of the data and allowing the designers to Understand the nature, role and scope of the data

Furthermore, It must allow designers to develop appropriate relationship between entities.

Evolution of Data Model

1] Hierarchical Data Model (HDM)
2] Network Data Model (NDM)
3] Relational Data Model (RDM)
4] Entity-Relationship Data Model (ERDM)
5] Object-Oriented Data Model (OODM)

1] Hierarchical Data Model :- Was developed in 1960



2] Network D
more complex r
It improves da
Codasy

The netw
language and
components a

D

1.] Too m
day framew

Relational
Conceptual
database
ability to
The "Re
and this
Consisting
RDB
is relate
Commonwe

2] Network Data model :- Is created to represent a more complex relationship in a more effective way. It improves database performance and was framework Conference on Data System languages

The network model has a schema definition language and sub-schema. dl that allows data components and that will be used effectively

Disadvantage

1.] Too cumbersome and not compatible with modern day framework.

Relational Model was developed by "E f Codd" Rm is conceptually simple and transformed into Relational database management System. The advantage is it's ability to hide the complexity from users

The "Relational data model" is made up of "tables" and this forms the Relation. The tables are matrices consisting of row, columns intersection.

RDBMS is a popular data Management System which is related to each other by enabling tables to share common entity characteristics.

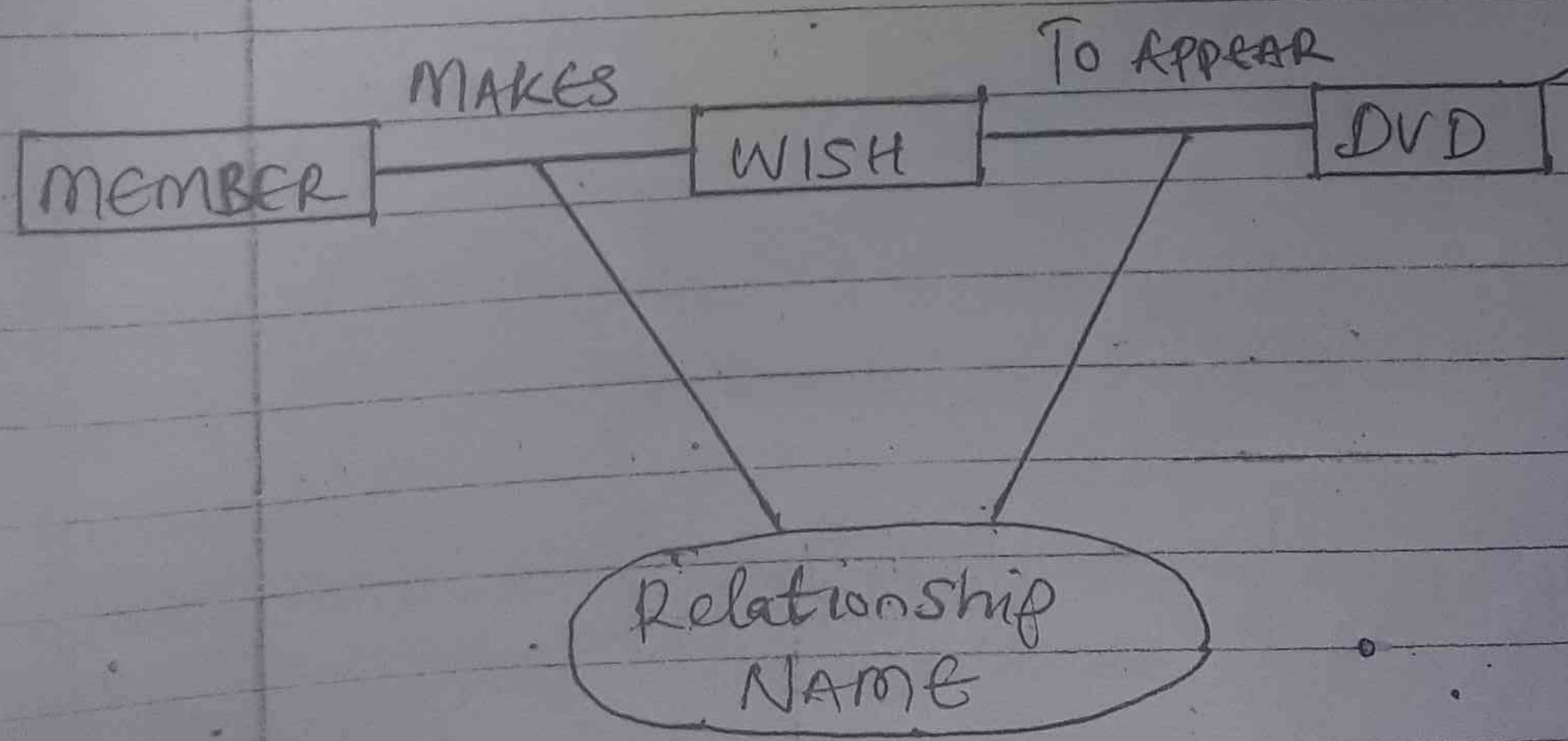
(E F Codd (IBM 1970))

"Relationship" is a set of meaningful association among entities. Relationship occurence is the uniquely identifiable relationship within a set

"Degree of relationship" is the number of participating entity

DVD
Printed Co

Recursive relationship are relationships where one entity participate more than once in different roles. Relationship roles may be given more than one name to indicate the participation of



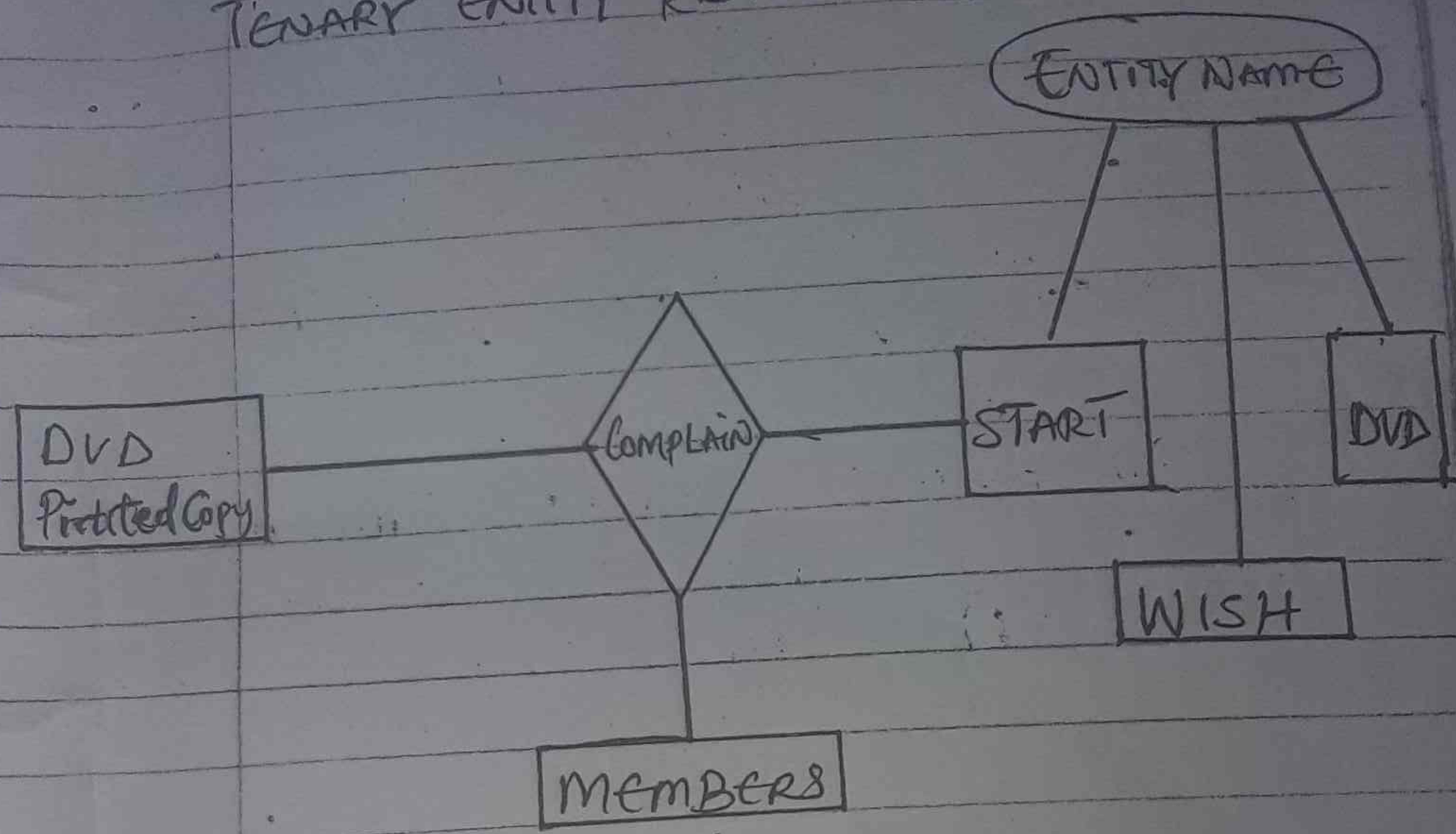
Key
PRIMA
to lo
key

Ar
depe
prim
Whe
"Ren
with

sociation
is the
a set
of

there
ent roles
one

TENARY ENTITY RELATIONSHIP DIAGRAM



KEYS

PRIMARY KEYS :- Is the Candidate key that is selected to identify each entity occurence. The Candidate key that are not selected as the "primary key"

An "Entity" is said to be strong if it doesn't depend on the existence on another entity for it's Primary key. The partial or weak entities partially / wholly depend on another entity for it's "PRIMARY KEY". These entities have constraints associated with them depending on the Business policy.

MULTIPLICITY CONSTRAINTS : Represents the number of entity occurence of one entity that may relate to a single occurence of an associated entity.

The most common degree of relationship is "BINARY"

Complex RELATIONSHIP MULTIPLICITY : Is the possible number of occurrences of entity type in an n array relationship of entity when the other (n-1) array are fixed

PROBLEM OF E-R MODEL INCLUDE

* Contain occurences that are related and have no means of connection
* Chas trap occurs between related entities that are not directly connected and the indirect pathway that connects them include partial participation.

08-07-20

Software
of ebu

1] SPEC
2] DE
3] VA
4] fo

Verifica
that ass
need. I

08-07-2019 Software Concept of Verification And Validation

Software process include a set of related activities that leads to the production of the software. These activities may involve the development of the software from the scratch, or, modifying an existing system

These are consistent processes in the software lifecycle. software process include a set of activities to develop a software

Activities of a software process

1] SPECIFICATION
2] DESIGN
3] VALIDATION
4] ~~EVALUATION~~ EVOLUTION

It is a description of the set of activities that make up the software

Verification & Validation (V & V) :- Is a process that assumes that the software meets the user's need. In verification, we are asking a question such as

Are we building the product right because we must confirm the software

Knowledge Meaning (information System)

Validation asks the question are we building the right product.

V & V process must be applied

1] To discover the defect in a system

2] To access whether it is usable for the

Verification can be static or Dynamic

Software testing is concerned with observation dynamic Verification

The goals of V & V are as follows

1] Verification & validation should exhibit confidence. It doesn't mean that the system is free of defects

we must

e building

m

for the

ic

n observation

llows

habit

ystem is

V & V planning :- Careful planning is to get the most of

Planning should start the development process

Software Inspection :- Observation for anomies and defect

~~Write~~

Assignment

(1) Write a short report on static Verification of the SRS

(2) With a well labelled diagram describe the V model of development

KEY POINT

Verification & VALIDATION DIFFERS

Verification shows conformance with specification

Validation shows that the software meets customer need

Static verification - involve examination analysis for error detection.
Program Code is respective

10-07-2019

CLEANROOM PROCESSES IN S.E UNDER V&V

The name is derived from the clean room process in semi conductors during the fabrication. The clean room is a technique that avoids defect other than removal of defects.

The Software development process based on environmental development. formal specification static verification and statistical testing.

In the clean room process there are some characteristics:

1. The former specification using a state transition model.
2. The structure program has limited control and abstraction that are used in their construct.

Static verification uses rigorous inspection, Statistical testing involves mathematical assignment that are used to increase confidence

In the inspection process
The clean room has a theme
This team implements stage of the clean room.

1) Specification team :- provide specification for developing and maintaining system specification

2) Development Team :- Responsible for development and verification of software that is needed by the user

3) Certification Team :- Responsible for developing a set of statistical test to exercise a software

Reliability growth models are commonly used to ensure that the softwares reliable.