UNIT - 3

13/10/18

REQUIREMENT ANALYSIS

A process for determining user expectations for a new or modified product. The features expected are to termed in the requirement.

Characteristics :

1. Correctness : Requirements must not have errors.
2. Consistent : It should maintain singular defination throughout the project. & must not contain any ambiquity.
3. Completeness : All inputs changes & constraints must be properly defined.
4. Realistic : Only possible & fesiable requirements should be analysed.
5. Verification : Requirements could be verified with different used cases.
6. Tracable : We can trace requirements from one to another which have interdependency.

Requirement Analysis Task :

1. Problem Recognisation : (i) Provides basic idea of problem. (ii) Provides overview of software (iii) User expectations are listed under
2. Problem Evaluation & Synthesis : (i) Flow & context of information. (ii) Defination & description of all software func^n's. (iii) Software behaviour & its effect : (iv) Provide design constraints.
3. Modelling : (i) Provides abstract description of the system. (ii) Makes the parts understandable for the user.

(iii) User can make modification in the model.

(4) Specification: (i) It provides final functionalities of the system those are agreed between developer & user.

(ii) SRS is built to define requirements.

(iii) It includes system's internal & external behaviour.

(5) Validation: (i) It includes information, behaviour, interfaces, & there specified validation criteria.

(6) Review: (i) It is used to create a check on the progress of the project.

ANALYSIS PRINCIPLES OF REQUIREMENT ANALYSIS

Principle 1: Identify the customers: (i) Identify the customers for the project

(ii) Define stakeholders which are the people who are directly or indirectly affected by new project.

Principle 2: Eliciting Requirements (Req. gathering): (i) Ambiguous understanding of the project. (different perception of singular thing)

(ii) Inconsistency in project by multiple customers.

(iii) Insufficient input from stakeholders.

(iv) Conflicting interest of customers.

Principle 3: Requirement Analysis & Negociation: (i) this process includes requirement gathering, categorisation, examination for consistency, necessity, ambiguity, test ability

Principle 4: Documentary Requirements: (i)

(ii) It provides high level description. (iii) It creates contact b/w developer & user (iv) It defines priorities of

(v) It makes clear abstractions b/w diff. layers of the system & it supports system design.

REQUIREMENT ANALYSIS TECHNIQUE

16/10/18

Analysis can use multiple techniques to gather requirements from the customers & stakeholders. This facilitates creation of prototype or near-exact representation of project.

Techniques are: (1) Stakeholder's Interview: A set of questionare is prepared to be asked from the stakeholders to get better idea of the requirements. (2) Joint Requirement Development Session: These sessions are used to define the info~ which is not collected by interviewing the stakeholders & are found out in this sessions with the combination of developers & users. (3) Use Cases: Functionality & performance related req. are analysed with the set of diff. use cases.

TYPES OF REQUIREMENTS

1) Architectural Requirement: Structure of the system.
2) Behavourial Requirement of system: How system behaves
3) Functional Requirement: Which func~ is performed
4) Non-Functional Requirements:
5) Performance Requirements: Shows performance: Depends on feedback
6) Design Requirements: Where user interact with system.
7) Derived Requirements: Size adjustable acc. to system used.
8) Allocated Requirements: From where allocation is done.

PROTOTYPING

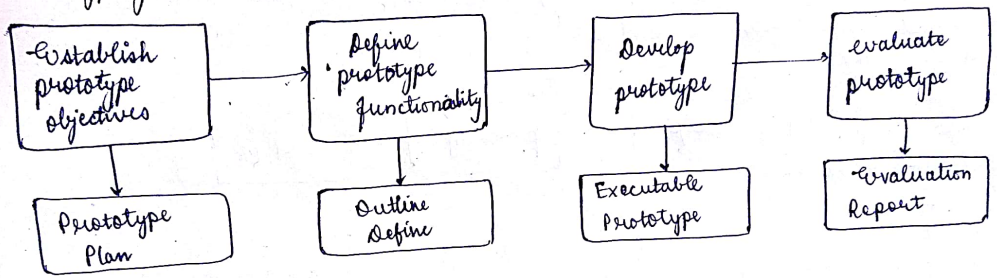
It is the rapid development of the system. Prototyping can be used in req. gathering so system could be seen by the user.

& user can exp. on it. It could be used for requirement validation so errors in the requirements can be revealed so prototyping is a risk reduction activity.

Benefits of Prototyping :

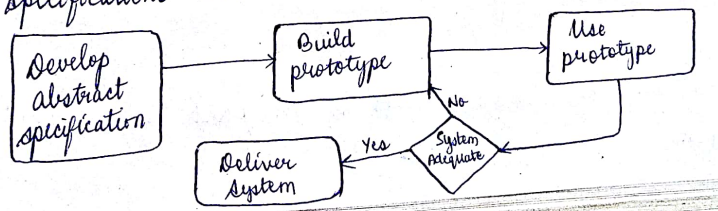
1) Misunderstandings between the user & developer can be identified.
2) Missing services & Confusing services can be identified.
3) Easy working system is available.
4) Sub-system training & testing requirements could be specified.
5) It could be used to improvise system usability & development effort

Prototyping Process :



Types of Prototypes

① EVOLUTIONARY PROTOTYPING : Initial prototype is produced & refined in no. of stages till the final system is achieved. Its objective is to deliver a working system to the end-user. It must be used where specifications cannot be developed in advanced.



Advantages of Evolutionary Prototyping.

1. This accelerates the delivery of the system.
2. User's high engagement with the system.
3. Case tools are used to define the system.

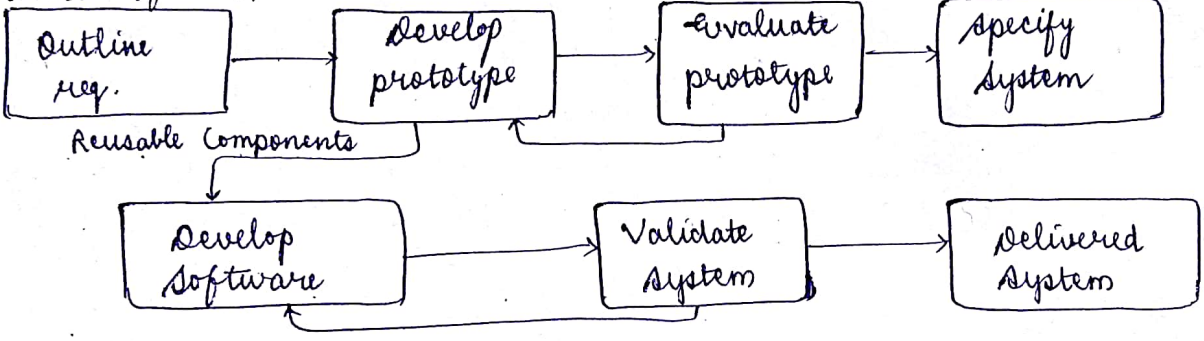
Disadvantages:

1. specialist skills are required to develop prototype.
2. Continuous changes in the trends & modification accordingly is difficult.

THROW AWAY PROTOTYPING

A practical implementation of the system is produced to help discover the req. problems & then discarded. The main objective of this prototyping is to reduce the req. stress so that the end system would be what user has expected.

Process of TAP:



DATA DICTIONARY

It is a textual list of all the concepts that are defined during the analysis. The purpose of this is to define a vocab that is common to all the members of the development team & users of the system.

1) It has the list of all the entities that is the part of object model of the system.

2) For each entity defined in data dictionary there should be a name & brief explanation of entity should be there.

3) It maintains structure defination of use of each data in the system.

4) The elements that are stored in data dictionary termed as data elements & information about it is stored in attributes.

Things we include in Data Dictionary:

1. Name
2. Synonyms
3. Description (1-2 line)
4. Date of origin
5. Users
6. Programs in which used
7. Authorisations (who can use this)
8. Data Type
9. Length
10. Unit (string : bits)
11. range of values
12. Frequency (how many times system can be used)
13. Input - Output
14. Conditional Values
15. Record Files (In which file this is stored)

Use of Data Dictionary

1. It keeps documentation of data & its flow
2. Ensures consistancy of elements b/w the systems.
3. It will be used for developing database.

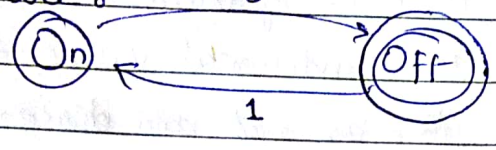
Eg: Data Dictionary for access control system

Noun	Relation	Verb
→ Card : have unique code	→ Register file	→ Connection is lost
→ Card Reader : hardware	→ Card entity	→ Entry in office
→ Door : control equipment	→ Door opening & closing	→ Exit in office
→ Employee : user		→ Validate card
→ Display : Interface		

Unit, Range, Data Type

Finite state machine - finite state machine is a multi facet model which has descriptive graphics and provied clear information. The major advantage of Finite state machine are :-

ex :-



1. Simplification → It achieves simplicity by diviiding a complex dynamic program into a set of state of static conditions. It has two parts
   1. states 2. Dynamics of moving between them

2. Model varification → Rules are formulated for FSM model movements against rules are form accepted under errors & exceptions. It gives the opportunity to correct & improvise requirments.

3. Derivation → It make transition from requirment to design and design to implementation

4. Testing → Test scenarios can be generated from model which can simplify the varification process.

FSM model specifications

1. Name → Label a state with name which must defines condition properly to eliminate diffrent interpretitions.

2. Description → Conditions are supplied with one or two sentences of discription to eliminate confusion.

3. Action → It specifies entry action and exit from a particular state.

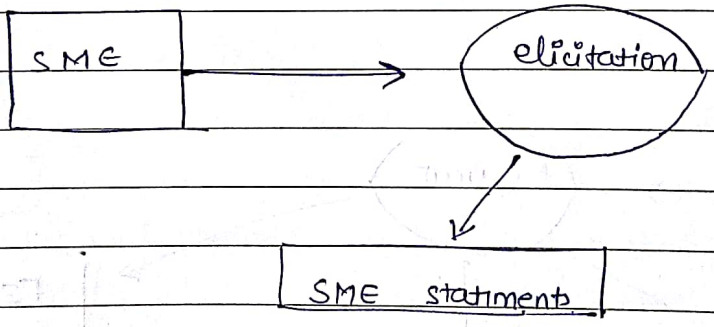
Requirment Specification with FSM model

Requirment specifications with FSM are provided in three variants

1. Requirment statment → is choosen from project or product
2. control variable values → control structure is chosen control variables are difines and there values are specified.
3. State in variant → Logical expression over control structure that evaluats to true in a state is used to control transition and implied constraint.

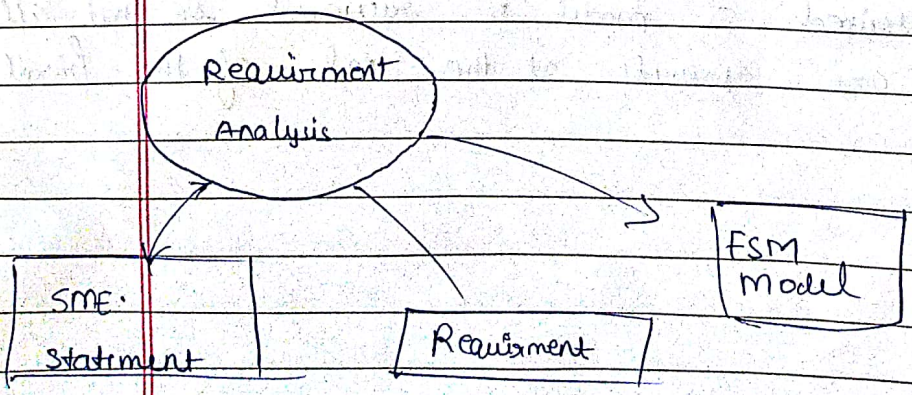
Constructing a requirment FSM model

step. 1. Elicitation



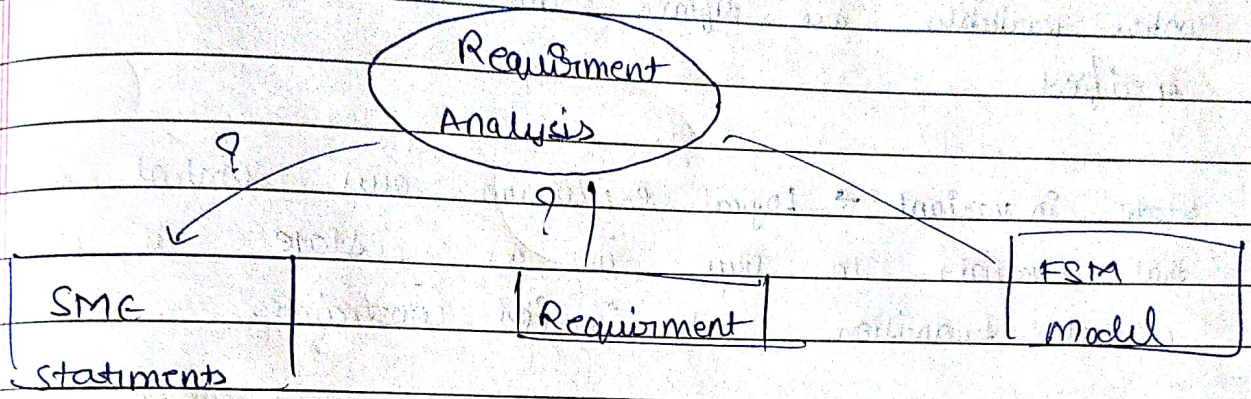
Information is recieved from SME & recorded as SME statment

step 2. Req. Analysis



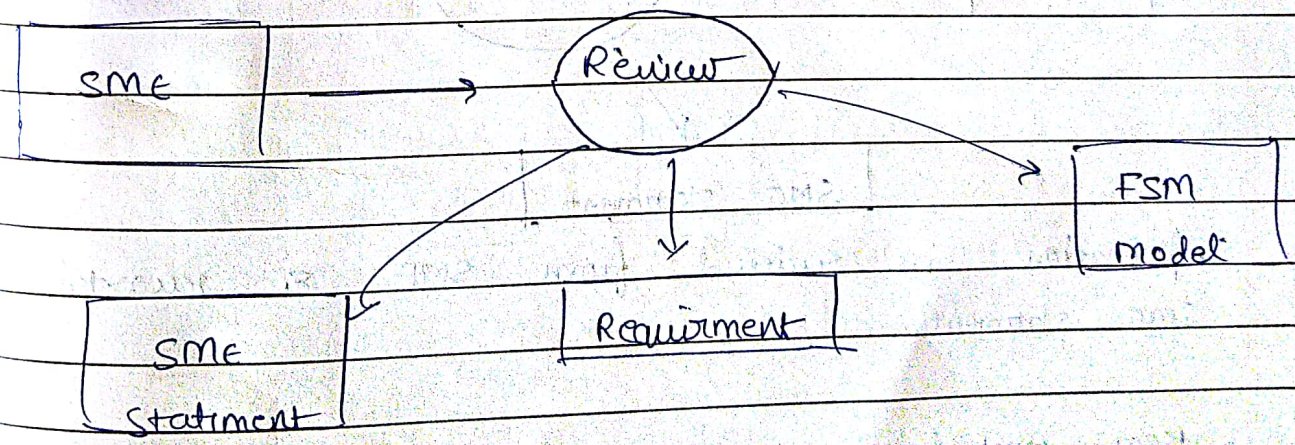
writing requirments & constructing a model that supports SME statements & can done in requirment analysis.

Step3 Model analysis



It does model verification for integrity & complitness using specific rules.

Step4 Review



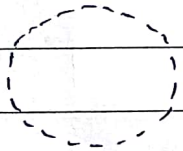
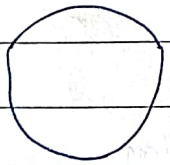
Questions are raised & model is reviewed so that all the requirments are aproved at the end of the final model.

Ans refer from unit 6 DFD

Data & control flow diagram

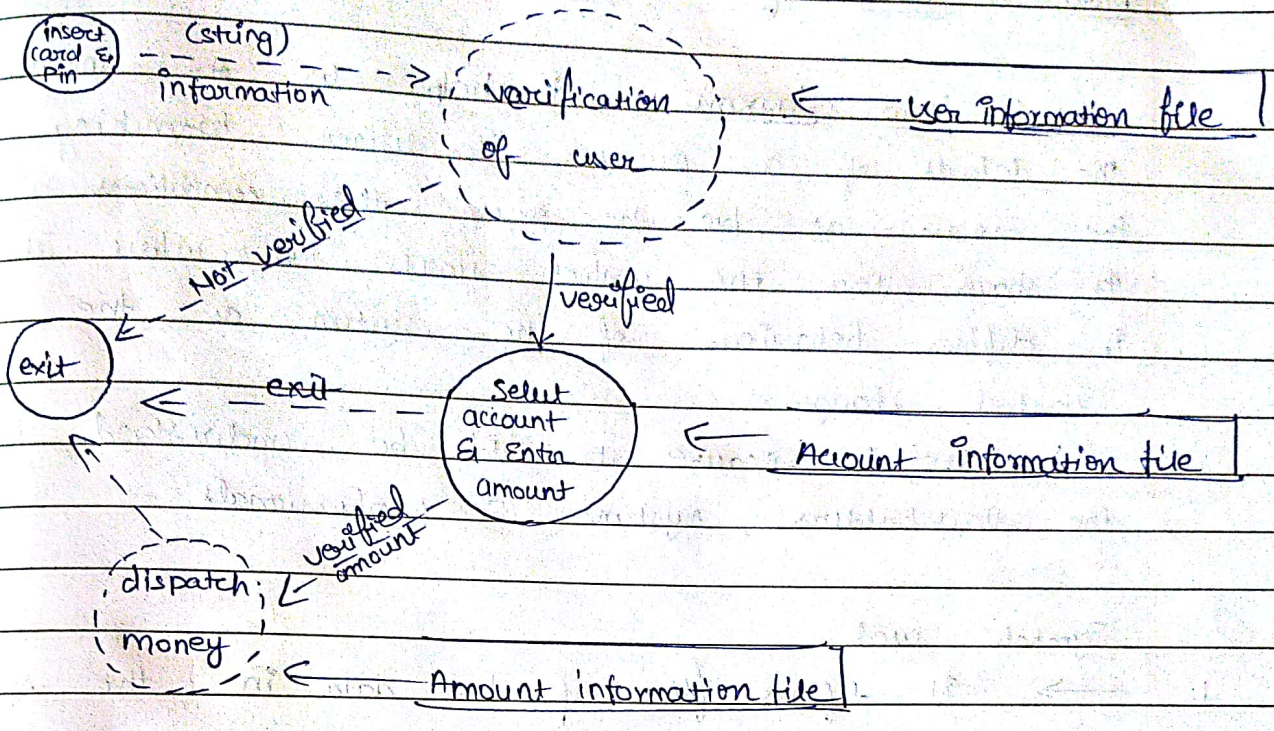
control flow diagram → It helps us to understand the details of a process. It defines branching of the process as per the given the conditions. It shows where the control starts & where it ends. It defines behaviours of the system as the control changes.
It is useful because it could be understood by the stakholders, systems & professionals.

Symbols used

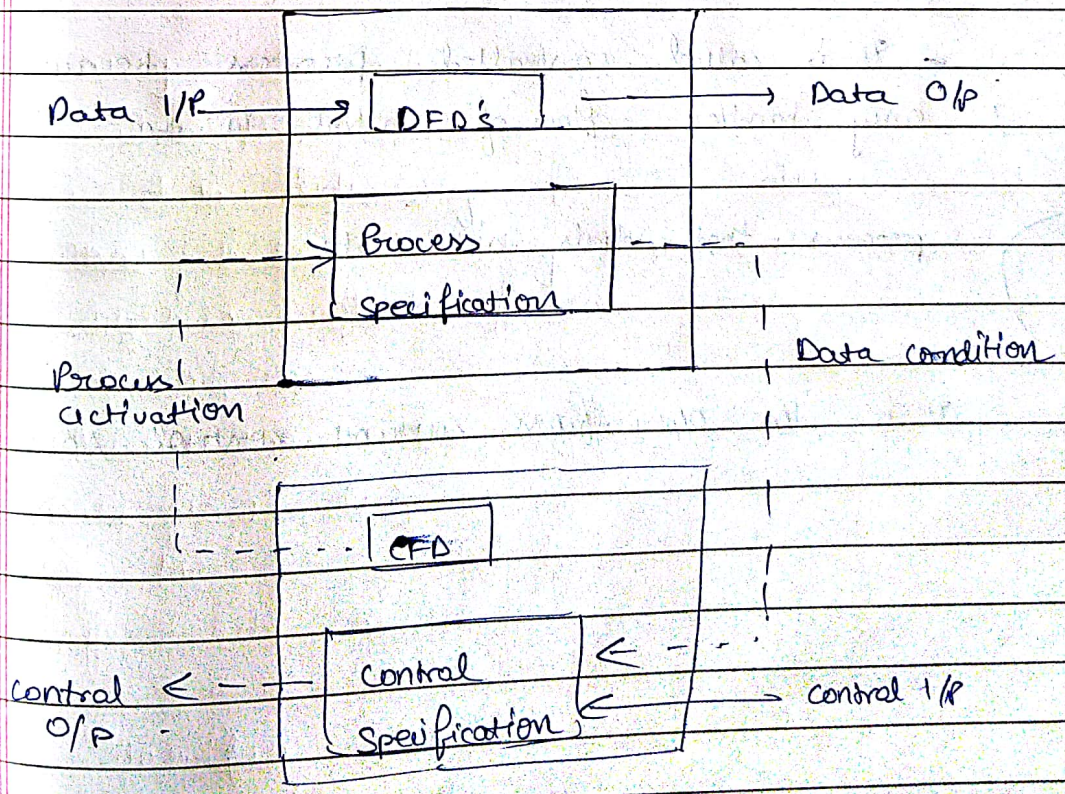
1. ⟶ It defines flow of the data in the system
2. ----> It defines the flow of control in the system
3.  → It is called controlled process because that only handles flow of control in the system.
4.  processus that deals with data
5. | It is I/p O/p from current control specification

classmate
Date
Page

1. Draw a control flow diagram for ATM Transaction system



Relationship between data & control model



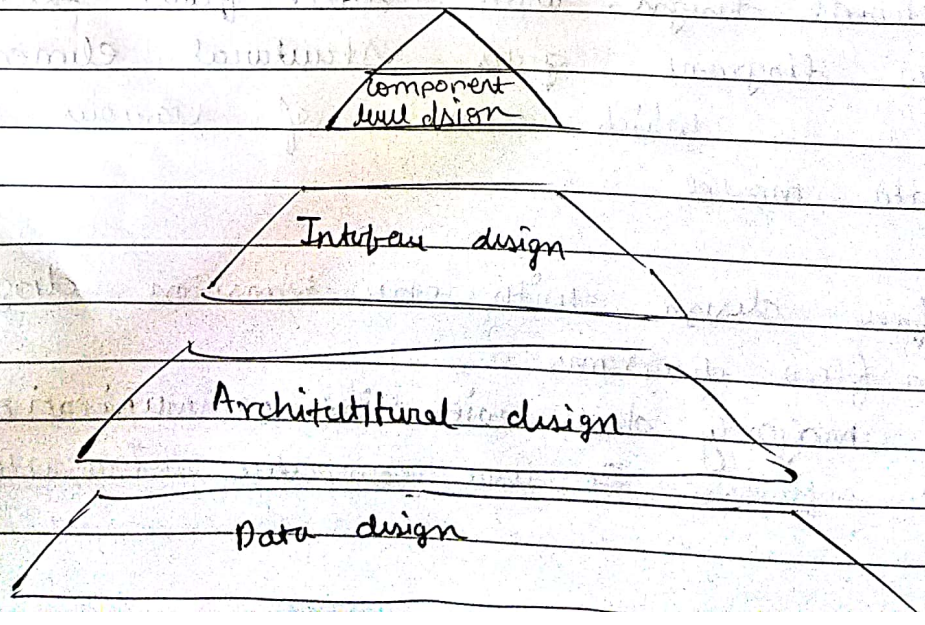
Control specification → It contains the sequential representation of behaviour of the system. It contains a process activation table that contains the information of the

process like event of appearance & condition for invocation.

Process specification → It is used to describe all flow model process that is design during analytical model. It includes information about programming language mathematical equation & algorithm at on which it works.

Behaviour modelling for control & process #4.22 (Book)
You need to explain state chart, seq. dig process, control spects.

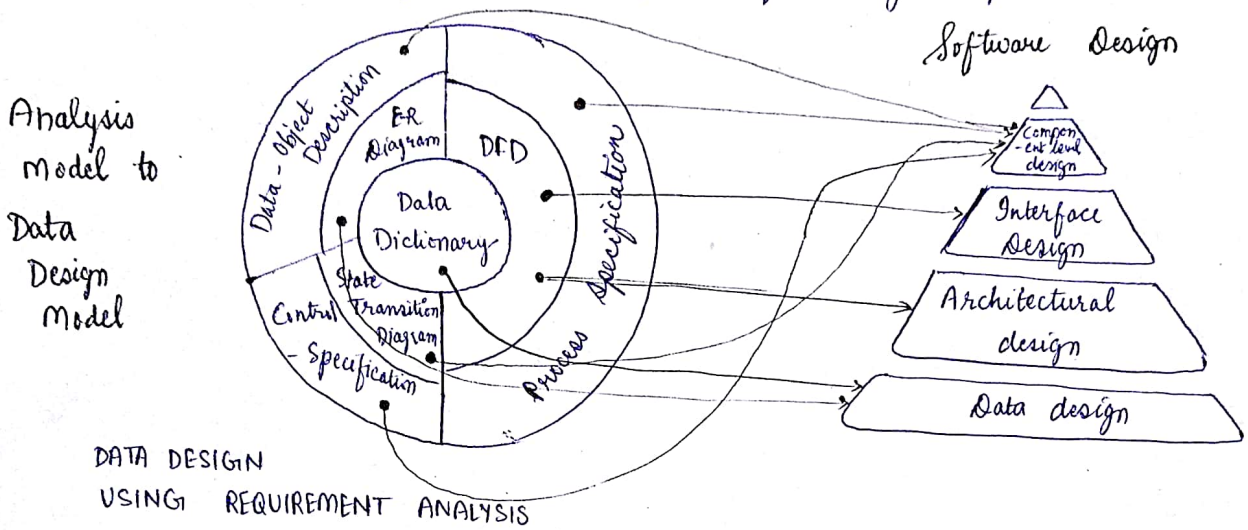
Software design → of applying various techniques design is a process for the purpose of designing a process or a system which could permit the development of the system: when the software is decomposed into number of modules with specific relationship design is produced to ~~describe~~ define architecture of system.
Analysis model is converted into design model by the following steps



SOFTWARE DESIGN

27/10/18

Design is a technique process of applying des various techniques for the purpose of designing. A process or a system which could permit development of system When the software is decomposed into no. of modules with specific relationships design is produced to define architecture of the system. Analysis model is converted into design model by the following steps.



DATA DESIGN USING REQUIREMENT ANALYSIS

1) **Data Design with data dictionary & entity relationship Diagrams:**
E-R Diagram gives data objects & there relationships & data dictionary provides details about the data that helps creating the data model of a system which further creates data structures for the system.

2) **Architectural Design with Data Flow Diagrams:**
Data flow diagram gives structural elements of the system which consist of various components of data model.

3) **Interface Designs with State Transition Diagrams & Data Flow Diagrams**
Interface majorly deals with inter-comm^{n} path of the software & user computer interaction dynamics

4) **Component Level Design:**
It includes control specifications, process specifications & data & object design to create a component of the system further integration of which provides the complete system to the user.

DESIGN FUNDAMENTALS:

① **Decomposition & Modularity**: Decomposition is a technique which breaks the system as per functional similarities in the system. Higher level description of the functions are used to implement & build lower level explanation of component execution & relation with other components.

Types of Decomposition:

① Data Oriented Decomposition: Data with similar data structures will be grouped under similar module.

② Event Oriented Decomposition: The events that will be happening in the system which will produce similar impact on the system will be grouped under similar module.

③ Object Oriented Decomposition: objects that are functioning with the similar class & similar inter-relations will be defined under similar module.

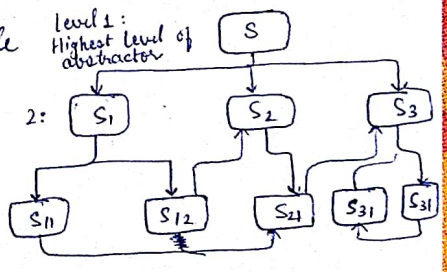
④ User Based Decomposition: This is the highest level of decomposition majorly depends on all possible users & there inputs which will be served to the system.

→ Modularity is dividing system into smaller modules such that when the system is re-integrated it will produce results similar to sequential execution of the system.

◉ Need for Modularity:

① If the system is consists of P_n processes such that ~~execution~~ of $E(P_1 + P_2 + \dots + P_n)$ $> E(P_1) + E(P_2) + \dots + E(P_n)$ then it is better to produce the ~~cost~~ processes as the individual entity & then integrating into the system.

② With the help of modularity abstraction is well defined where the sub-systems can be used further for multiple applications.
If S_{11} could be used in other application, it is reusable without any derived complexity.



③ Easy data structure modification is possible in modular elements of the system. Changes can be reflected easily when global data is present in the system.

◉ Dependancy of Cost In Modularity:

① $E(P_1+P_2+\dots+P_n) < E(P_1)+E(P_2)+\dots+E(P_n)$: When the modular decomposibility is

taking greater amount of time & effort it is not efficient to use modularity as an approach.

(2) Modular Composibility : Re-Integration of all the components is taking a larger amount of time then it is not useful to use modularity.

(3) Modular Understandibility : When complex co-relations exist b/w the sub-systems it impacts the cost higher.

(4) Modular Security : When ↑sing module ↑se the security req. modularity should not be used.