Varun Khadayate A016 1-9-20

S.E

Assignment - III

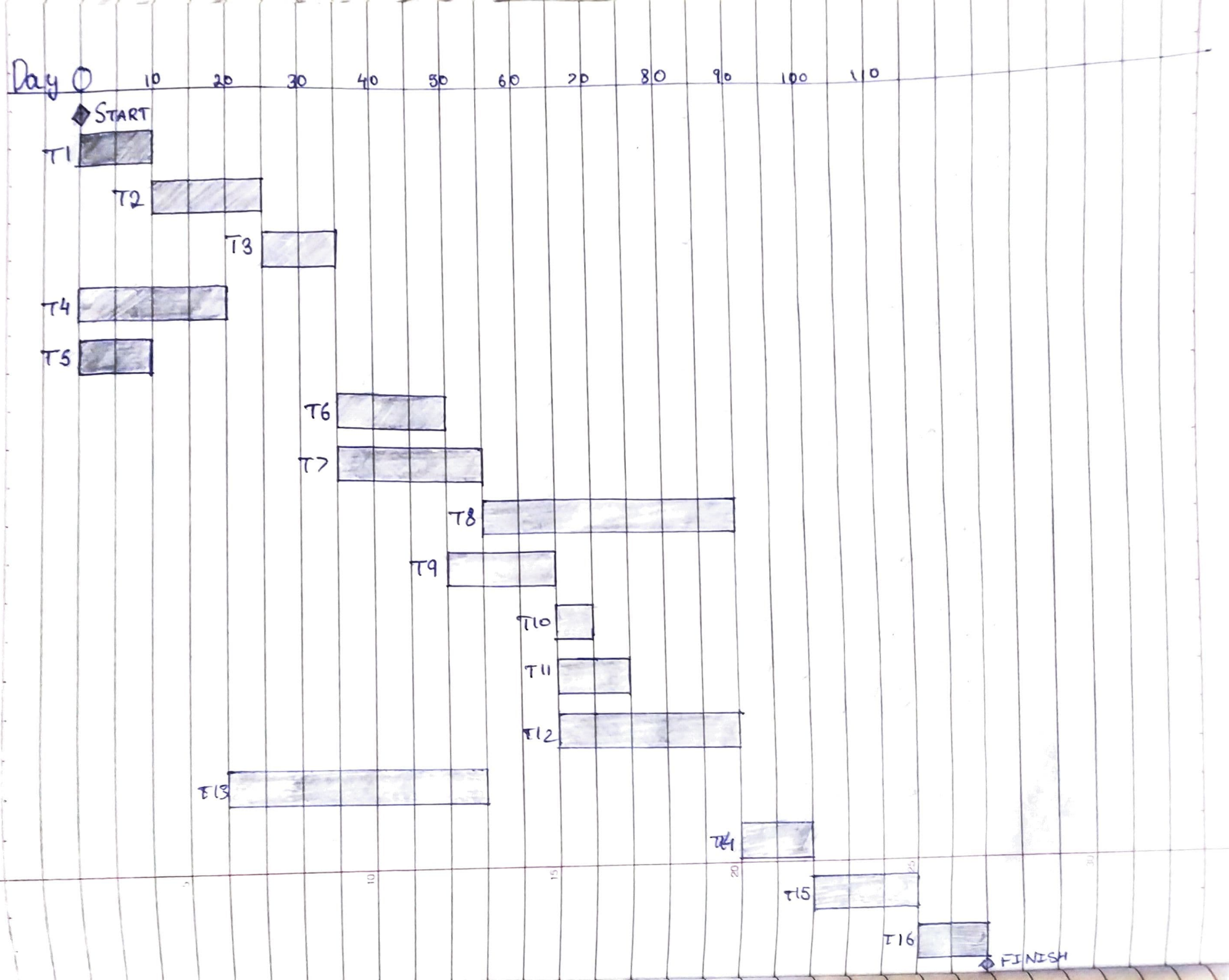
Q2.

Task	Duration (Days)	Dependencies
T1	10	
T2	15	T1
T3	10	T1, T2
T4	20	~~T1, T2~~
T5	10	
T6	15	T3, T4
T7	10	T3
T8	35	T7
T9	15	T6
T10	5	T5, T9
T11	10	T9
T12	20	T10
T13	35	T3, T4
T14	10	T8, T9
T15	20	T12, T14
T16	10	

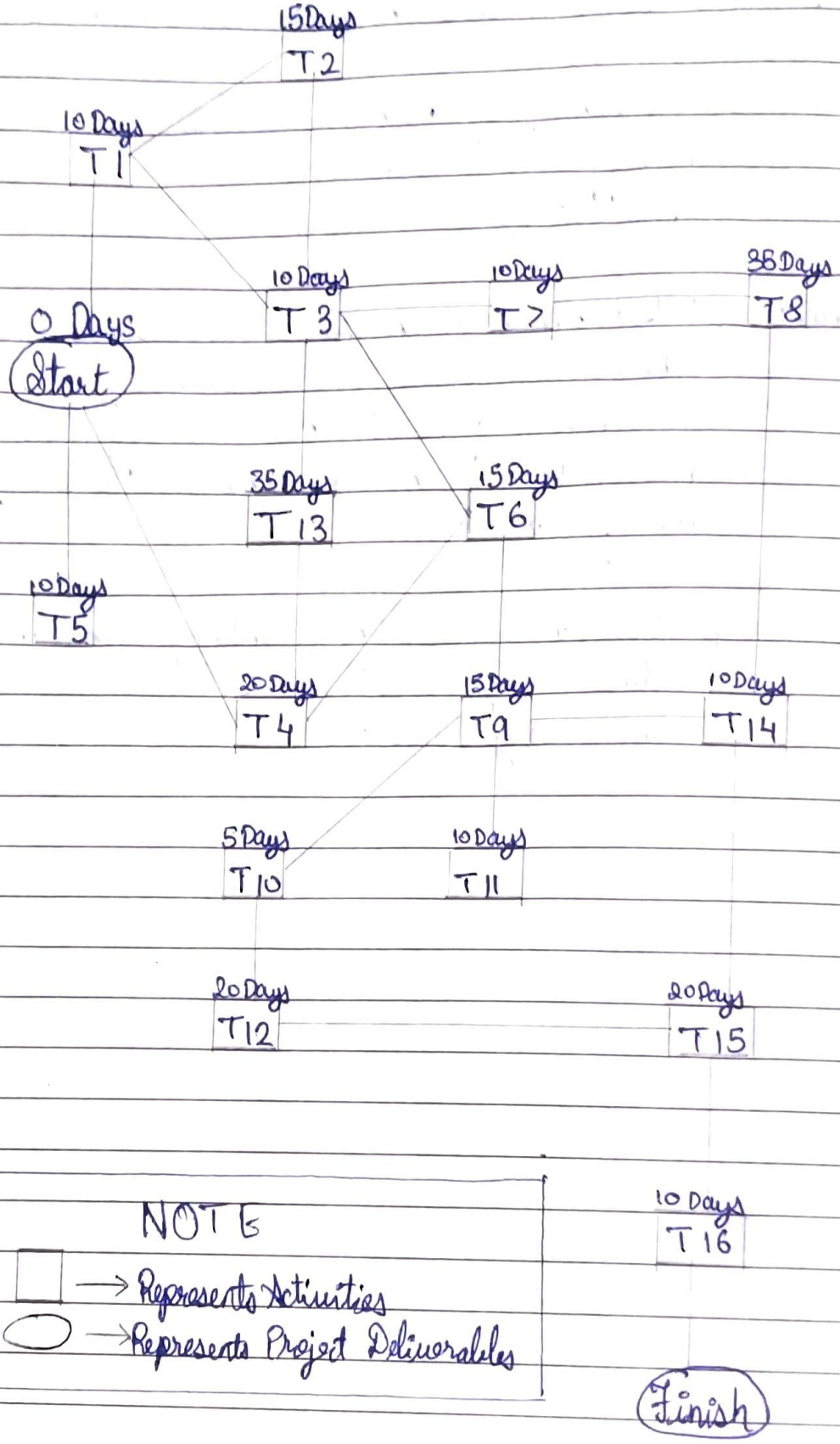
Dependencies =

Activities = 16

Duration (Days) = 250 days / 8.2 Months


Day 0
10
20
30
40
50
60
70
80
90
100
110
START
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12
T13
T14
T15
T16
FINISH

ACTIVITY CHART



The activity chart shows activities that can be carried out in parallel and the activities that must be executed in sequence because of a dependancy.

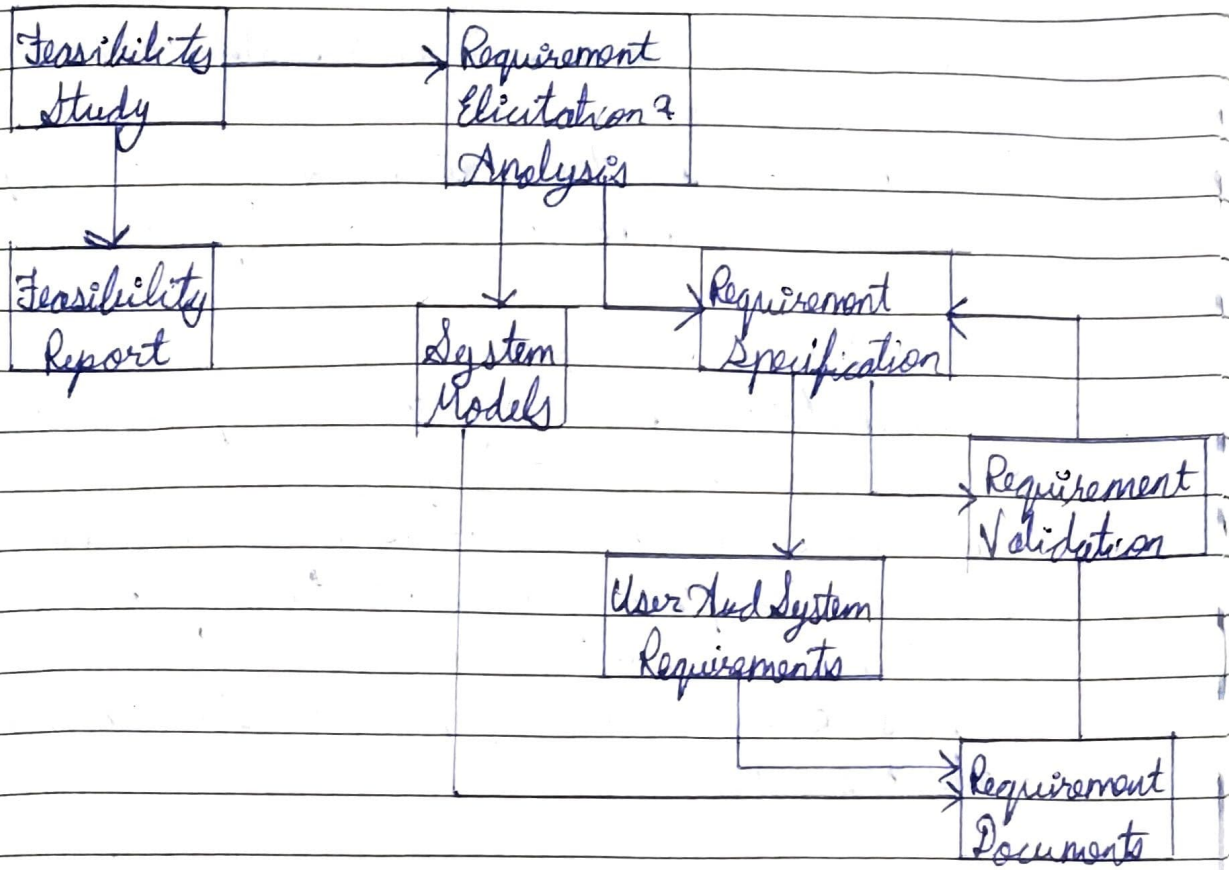
Q1. Process Activities are the activities that belong to a particular process. It defines the smallest measurable amount of work to be performed to convert some portion of process activities inputs into desired outputs. The 4 basic process activities of specification, development, validation, and evolution are organised differently in different development processes. In the waterfall model, they are organised in sequence whereas in evolutionary development they are interleaved. How these activities are carried out depend on the type of software, people and organisational structures involved.

The 4 process activities are:-

- Software Specification
- Software Design and Implementation
- Software Validation
- Software Evolution

1. Software Specification

A software requirement is defined as a condition to which a system must comply. Software specification or requirement management is the process of understanding and defining what functional and non-functional requirements are required for the system and identifying the constraints on the system operation and development. The requirement engineering process results in the production of a software requirement document that is the specification for the system.



There are 4 main phases of requirement engineering process:-

1. Feasibility Study

In this study an estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies. It considers whether the proposed system will be cost effective from a business point of view and ~~whether~~ whether it can be developed within existing budgetary constraints

2. Requirement Elicitation And Analysis

This is the process of ~~discovering~~ deriving the system requirements through observation of existing systems, discussion with potential users, requirement workshops etc.

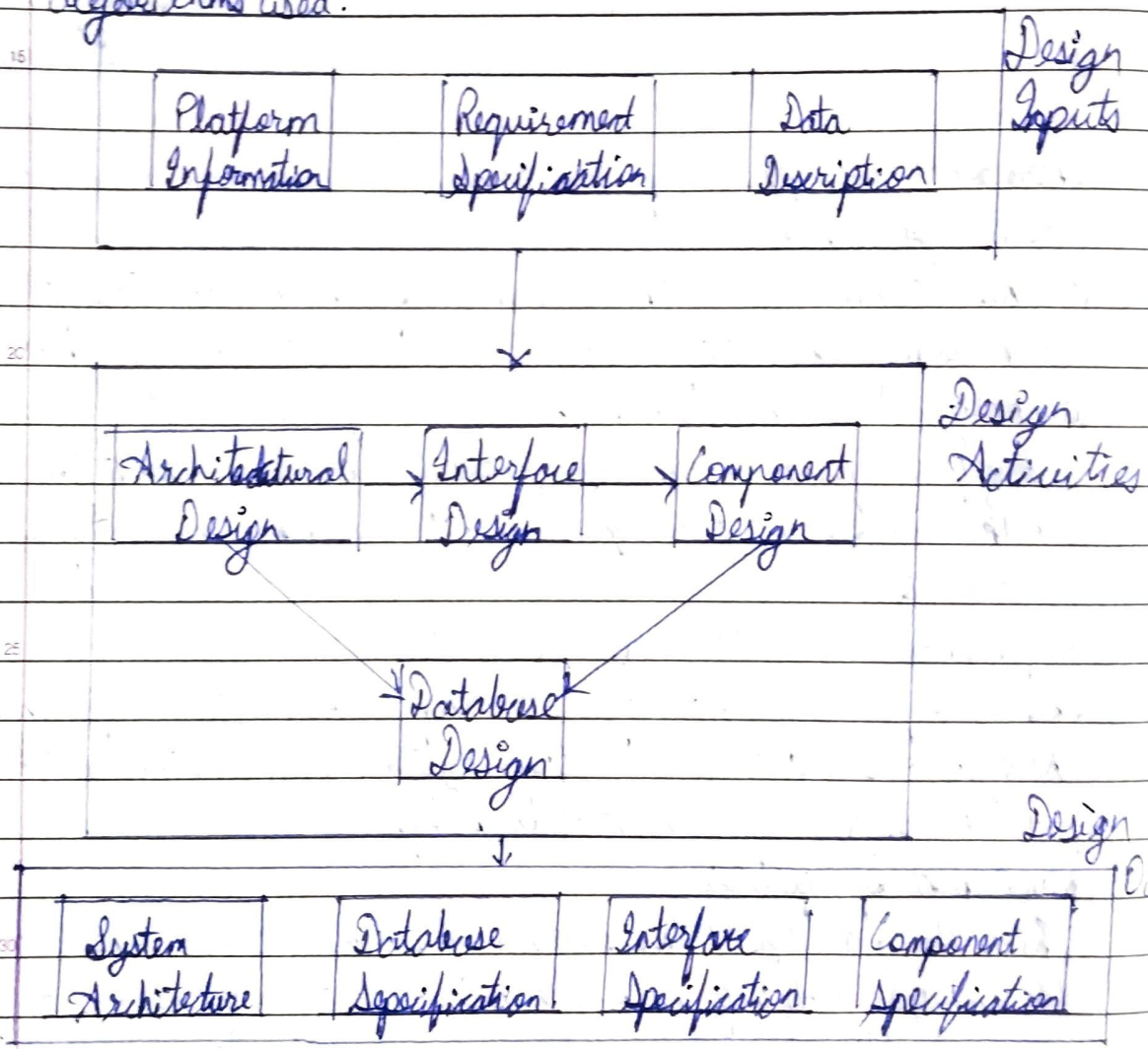
3. Requirement Specification

This is the activity of translation of information gathered during the analysis activity into a document that defines a set of requirements. Two ~~types~~ types of requirement may be

included in these documents: User & System requirements.

4. Requirement Validation
It is determined whether the requirements defined are complete. This activity also checks the requirements for consistency.

2. Software Design and Implementation
The implementation phase of software development is the process of converting a system specification into an executable system through the design of system. A software design is a description of the architecture of the software to be implemented, the data which is part of the system, the interfaces between system components and sometimes, the algorithms used.

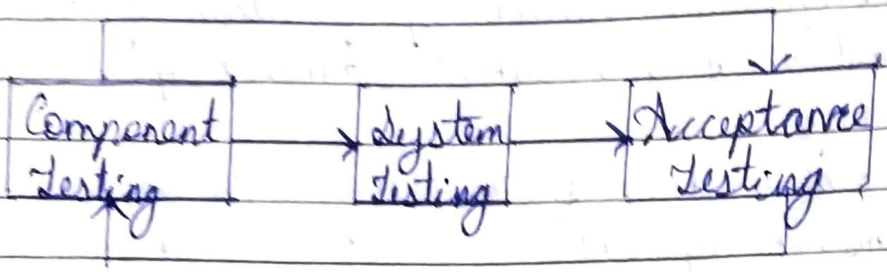


The design process activities are the following:

1. Architectural Design
The sub-system of a system and their relationships are identified based on their main functional requirements of software.
2. Abstract Specification
For each subsystem, an abstract specification of its services and the constraints under which it must operate is defined.
3. Interface Design
Interface allow the subsystem services to be used by other sub-systems. The representation of interface should be hidden. In this activity, the interface is designed and documented for each subsystem. The specification of interface must be unambiguous.
4. Component Design
Services are allocated to components and the interfaces of these components are designed.
5. Data Structure Design
The database structures used in the system implementation are designed in detail and specified.
6. Algorithm Design
The algorithms used to provide services are designed in detail and specified.

3. Software Validation
Software Validation or more generally, verification and validation is intended to show that a system conforms its specification and that the system meets the expectation of the customer buying the system. It involves checking the process at every stage of the software process. The majority of validation costs are incurred after implementation when the operation of system is tested. The software is

in the usual 3-stages testing process. The system components, the integrated system and finally the entire system are tested. Component defects are generally discovered early in the process and the interface problems during the system integration.



The stages in testing process are:-

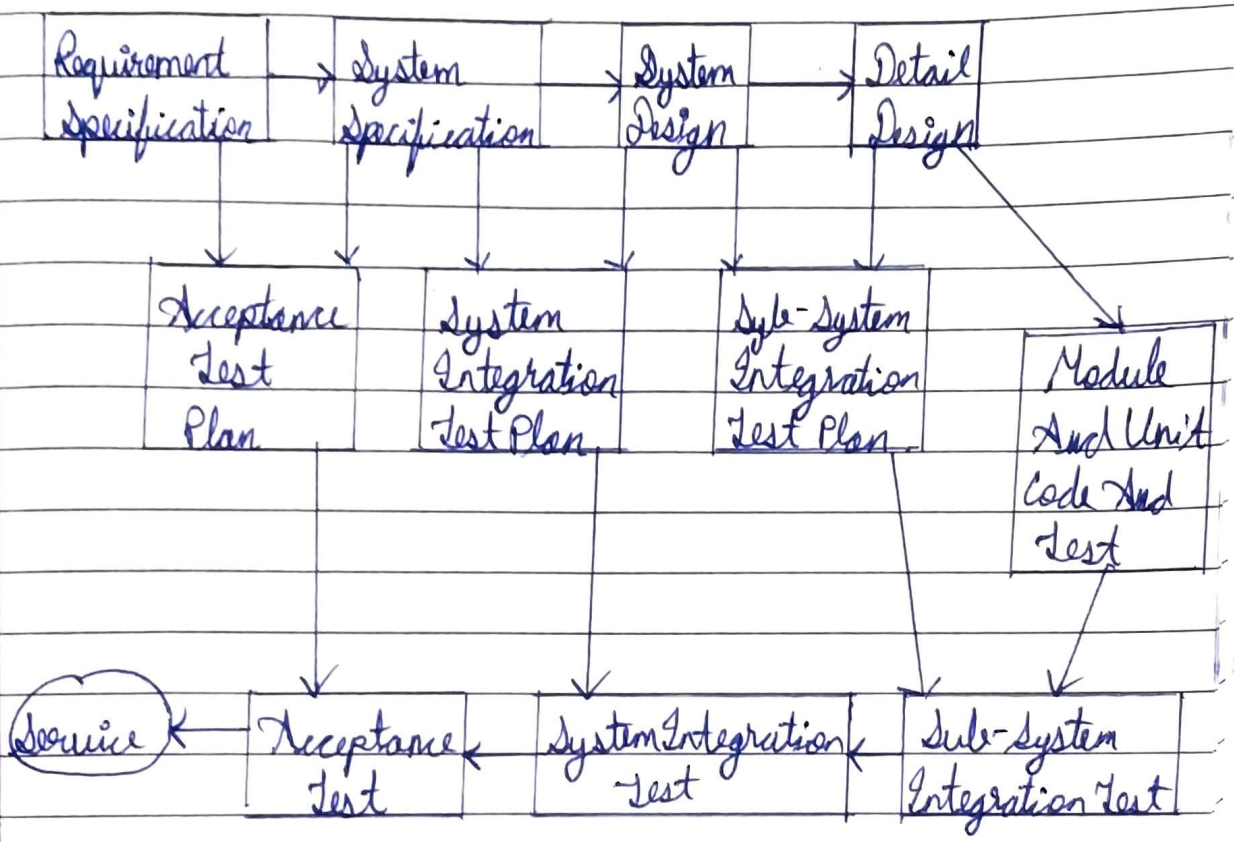
1. Component Testing
Individual component are tested to ensure that they operate correctly. Each component is ~~used~~ tested independently, without other system component.

2. System Testing
The components are integrated to make up the system. This testing process is concerned with finding errors that result from iteration between components and component interface problems. It is also concerned with validation that the system meets its functional and non-functional requirements.

3. Acceptance Testing
It is considered a functional testing of system. The system is tested with data supplied by the system customer.

Usually component development & testing are interleaved. Programmers make their own test data and test the code as it is developed. However in many process model, such as in V-model, Test-Driven Development etc. The design of

test cases starts before the implementation phase of development. If an incremental approach to development is used, each increment should be tested as it is developed, with these test based on the requirements for that increment.



4. Software Evolution

Software evolution, specifically software maintainance, is the term used in software engineering to refer to the process of developing software initially, then repeatedly updating it for various reason. The aim of software evolution would be to implement the possible major changes to the system. The existing larger system is never complete and continuous to evolve. As it evolves, the complexity will grow. The main objective of software evolution and ensuring the reliability and flexibility of the system. The cost of maintainance are often several times the initial development cost of software.