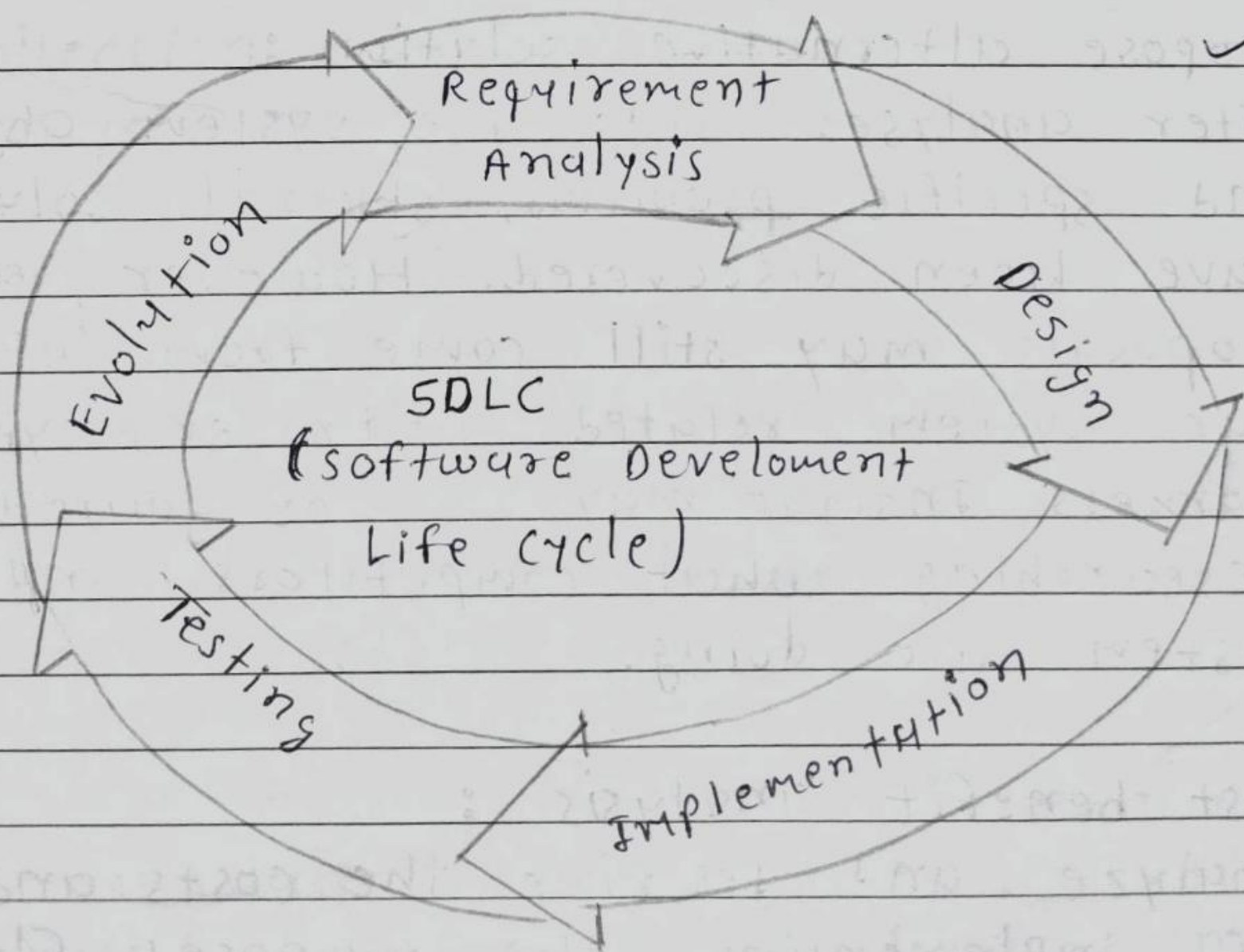
Page No.:
Date:

Software Testing Assignment

Module-1 (Fundamental)

write SDLC phases with basic Introduction
~~what is SDLC~~

The software development life cycle (SDLC) is a formal way of ensuring that adequate security controls and requirements are implemented in a new system or application. Integrating technologies and practices into the development of new system and application deployments provides an opportunity to design security into the solution on the front end of the process, rather than retrofitting it after the solution is deployed.



Scanned by TapScanner

Page No.:
Date:

* various phases of Life cycle :

→ preliminary analysis :

To identify the real problem of the system with all necessary user requirement is called preliminary investing of the system. The preliminary investigation of the project or system is help in :

1. Conduct the preliminary analysis :
Discover the system or projet objectives and the nature and scope of the problem under study. Even if a problem refers only to a small segment of the system itself, find out what the objective of the system itself are. Then see how the problem being studied fits in with them.

2. propose alternative solution :
After analyses into the system objectives and specific problems, several solution may have been discovered. However, alternate proposals may still come from interviewing our system related entity employees, customer, workers Insight may also be gained by researching what competitors and existing system are doing.

3. Cost benefit analysis :
Analyze and describe the costs and benefits of implementing the proposed Changes. In the end, the ultimate desision on whether

Scanned by TapScanner

to leave the system as is, improve it, or develop a new system will be guide by this and the rest of the preliminary analysis data.

→ systems analysis, requirements definition :

Define project goals into defined functions and operations of the intended application. This involves the process of gathering and interpreting facts, diagnosing problems, and recommending improvements to the system. project goals will be further aided by analysis of end-user information needs and the removal of any inconsistencies and incompleteness in these requirements.

A series of steps followed by the developer include.

1. collection of facts
2. Analysis of the proposed system

→ systems design :

At this step desired features and operations are described in detail, including screen layouts, shop management rules, process schedule and other documentation.

Develop detailed design specifications that translate functional specifications into a logical and physical design. Detailed design specification are developed during

Scanned by TapScanner

phase of the SDLC and describe how the system or application is designed to satisfy the requirements documented in the functional specifications.

Detailed design specifications should include the following :

- database requirements
- file requirements
- system flow diagram
- program requirements
- system operations requirements
- screen designs
- Report designs
- security design

→ Development

The development phase is where the system or application's security features are developed, configured and enabled. Use the program specifications to describe the program logic and processing requirements. program specifications are developed as part of the development phase prior to the commencement of programming. These specifications provide the thought process required to determine the steps to code the programs.

Scanned by TapScanner

→ System testing :

The system testing is used experimentally to ensure that the project works is perfectly or completely. Data are input in the system and the result of the system performance is checked for data processing. Testing is performed by analyst, user and persons who are not at all connected with the system that help to the developer to developed well system for real environment.

→ Implementation :

The implementation view of software requirement presents the real world manifestation of processing functions and information structures. This computerized system is specified in a manner that dictates accommodation of certain implementation details.

The implementation environment of the developed system facilitates multiple users to use this system simultaneously. The user interfaces are designed keeping in mind that the users of this system are familiar to using GU-based systems. Thus, we restricted ourselves to developing a GUI - based system so that it becomes easier for the end user to get acquainted to the developed system.

Scanned by TapScanner

Implementation includes :

- side preparations
- Installations of new equipment
- User's training, seminars, meeting to increase user support.
- use of new inputs and procedures.

→ Evaluation

Evaluation of the system is performed to identify its strength and weakness. The actual evaluation can occur along any of the following dimensions

A. operational Evaluation
B. organizational Impact
C. User Manager Assessment
D. Development performance

→ Maintenance :

- Maintenance is necessary to reduce the errors in the system during its operations and to correct the system any variations in its working environment.

- In this phase changes are made in the system by adding or deleting features to satisfy user needs.

- The importance is to continue to bring the new system to standard system.

Scanned by TapScanner

what is agile Methodology ? / Explaing working Methodology of agile model and also write pros and cons

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

Agile methods break the product into small incremental builds.

These builds are provided in iterations.

Each iteration typically lasts from about one to three weeks.

~~Every iteration typically lasts from about one to three weeks.~~ Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.

At the end of the iteration a working product is displayed to the customer and important stakeholders. what is Agile ?

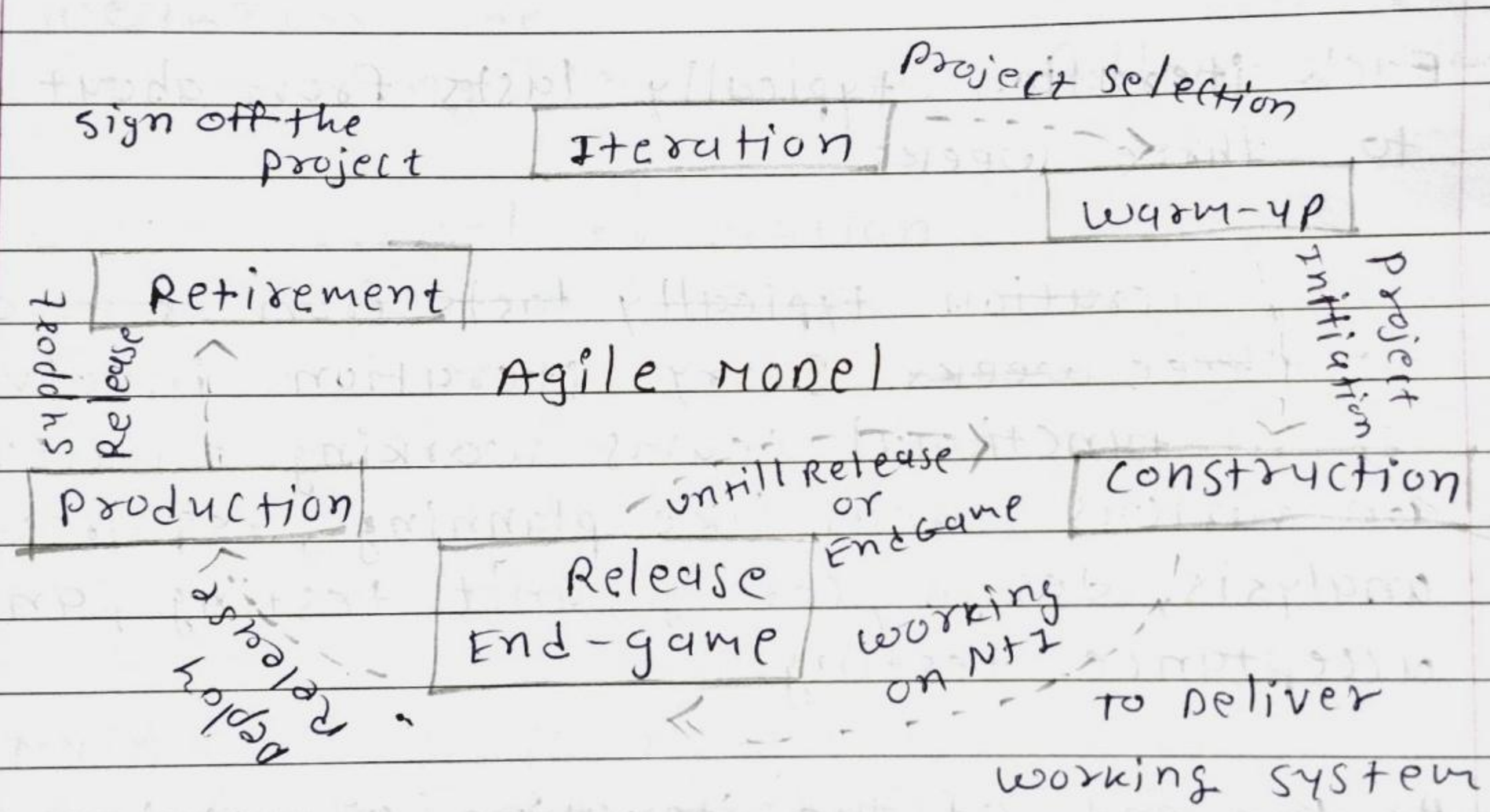
Agile model believes that every project needs to be handled differently and the existing methods needs to be tailored to best suit the project requirements. In agile the tasks are divided to time.

Scanned by TapScanner

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features, the final holds all the features required by the customer.

Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

Agile Model work flow



Agile manifesto principles

Individuals and interactions :- in agile development; self-organization and motivation are important, as are interactions like co-location and pair programming.

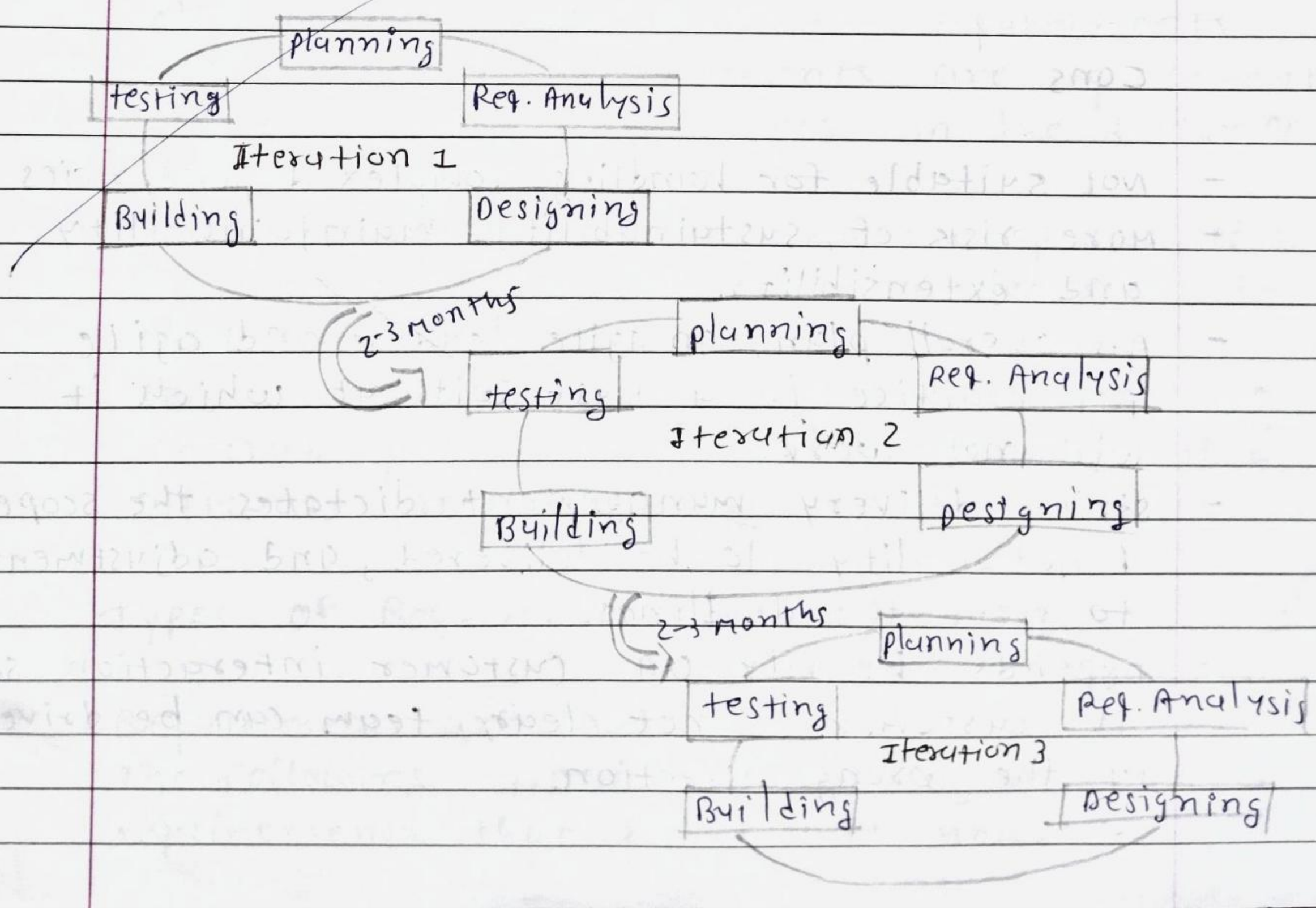
Scanned by TapScanner

Working software :- Demo working software is considered the best means of communication with the customer to understand their requirement, instead of joust depending on documentation.

Customer collaborution :- As the requirements cannot be gathered completely in the beginning of the project due to various factors, continyous customer interaction is very importunt to get proper product requirements

Responding to change :- agile development is focused on quick responses to change and continyous development.

Agile Model work flow



Scanned by TapScanner

pros

- Is a very realistic approach to software development.
- promotes teamwork and cross training.
- functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- suitable for fixed or changing requirements
- delivers early partial working solutions.
- Good model for environments that change steadily.
- minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required
- Easy to manage
- Gives flexibility to developers

cons

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.

Scanned by TapScanner

- There is very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of.

What is SRS

Introdyction

- A software requirements specification (SRS) is a complete description of the behavior of the system to be developed.
- It includes a set of use cases that describe all of the interactions that the users will have with the software.
- Use cases are also known as functional requirements. In addition to use cases, the SRS also contains nonfunctional requirements.
- Non-functional requirements are requirements which impose constraints on the design or implementation.
- Recommended approaches for the specification of software requirements are described by IEEE 830-1998.
- This standard describes possible structures, desirable ~~con~~ contents, and qualities of a software requirements specification.

Types of Requirements

Requirements are categorized in several ways. The following are common categorizations of requirements that relate to technical management

Page No.:
Date:

Customer Requirements

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. operational requirements will define the basic need and, at a minimum, answer the questions posed in the following listing:

- operational distribution or deployment: where will the system be used?
- Mission profile or scenario: how will the system accomplish its mission objective?
- performance and related parameters: what are the critical system parameters to accomplish the mission?
- Utilization environments: How effective or efficient must the system be in performing its mission?
- operational life cycle: How long will the system be in use by the user?
- Environment: what environments will the system be expected to operate in an effective manner?

Functional Requirements

Functional Requirements are very important system requirements in the system design process. These requirements are the technical specifications, system design parameters and guidelines.

Scanned by TapScanner

data manipulation, data processing, and calculation modules etc, of the proposed system.

For Example :

- The following are the requirements of Google Email service
- The system shall support the ability to receive emails
- The system shall support the ability to create new folders.

Non-functional Requirements

- Non-functional Requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors. Non function requirements are qualities.

- or standards that the system under development must have or comply with, but which are not tasks that will be automated by the system.

Example :- system must be built for a total installed cost of $ 1,050,000,00

- system must run on ~~wid~~ windows server2003

following categories :-

- Usability
- Redilbility
- performance
- security

Scanned by TapScanner

what is oops

- Identifying objects and assigning responsibilities to these objects.
- objects communicate to other objects by sending message.
- Message are received by the methods of an object
- An object is like a black box.
- The internal details are hidden.
- object is derived from abstract data type.
- object - oriented programming has a web of interacting, each house-keeping its own state.
- objects of a program interact by sending messages to each other.

write Basic concepts of oops

- object
- class
- Encapsulation
- Inheritance
- polymorphism
  - overriding
  - overloading
- Abstraction

what is object

- Tangible Things as a car, printe,....
- Roles as employee, boss,....
- Incidents as flight, overflow,...

Scanned by TapScanner

- Interactions as contract, sale,...
- specifications as color, shape,...

What is class

- when you difine a class, you define a blueprint for an object

- This doesn't actually define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.

- A class represents an abstraction of the object and abstracts the properties and behavior of the object.

- class can be considered as the blueprint or definition or a template for an object and describes the properties and behavior of that object, but without any actual existence.

- An object is a particular instance of a class which has actual existence and there can be many objects for a class.

- In the case of a car or laptop, there will be a blueprint or design created first and then the actual car or laptop will be built based on that. we do not actually buy these blueprints but the actual objects.

Scanned by TapScanner

What is encapsulation

- Encapsulation is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.

- Encapsulation is placing the data and the functions that work on that data in the same place. while working with procedural language, it is not always clear which functions work on which variables but object-oriented programming provids you framework to place the data and the relevant functions together in the same object.

- Encapsulation in Java is the process of wrapping up of data (properties) and behavior (methods) of an object into a single unit, and the unit here is a class (or interface).

- Encapsulate in plain English means to enclos or be enclosed in or as if in a capsule. In Java, a class is the capsule (or unit).

- In Java, everything is enclosed within a class or interface, unlike languages such as C and C++, where we can have global variable outside classes.

Scanned by TapScanner

- Encapsulation enables data hiding, hiding irrelevant information from the users of a class and exposing only the relevant details required by the user.

- We can expose our operations hiding the details of what is needed to perform that operation.

- We can protect the internal state of an object its attributes from the outside world, and then exposing them through setter and getter methods.

Now modifications to the object internals are only controlled through these methods.

What is inheritance

- Inheritance means that one class inherits the characteristics of another class. This is also called a "is a" relationship.

- One of the most useful aspects of object oriented programming is code reusability. As the name suggests Inheritance is the process of forming a new class from an existing class that is from the existing class called as base class, new class is formed called as derived class.

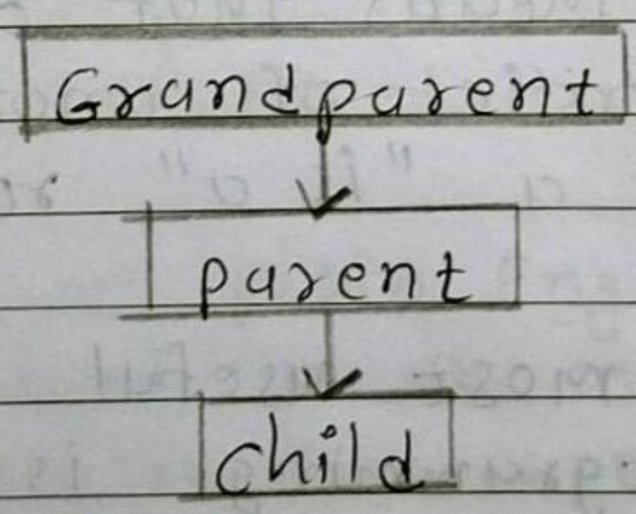
Scanned by TapScanner

- This is a very important concept of object-oriented programming since this feature helps to reduce the code size.

- Inheritance describes the relationship between two classes. A class can get some of its characteristics from a parent class and then add unique features of its own.

- In general, Java supports single-parent, multiple-children inheritance and multilevel inheritance (Grandparent → parent → child) for classes and interfaces. Java supports multiple inheritances (multiple parents, single child) only through interfaces.

- In a class context, inheritance is referred to as implementation inheritance, and in an interface context, it is also referred to as interface inheritance.



For example consider a vehicle parent class and its child class car.

- vehicle class will have all common properties and functionalities for all vehicles in common and car will inherit those

common properties from the vehicle class and then add those properties which are specific to a car.

- Here, vehicle is known as base class, parent class, or super class.
- car is know as derived class, child class or subclass.

what is polymorphism

- polymorphism means "having many forms".
- It allows different objects to respond to the same message in different ways, the response specific to the type of the object.
- The most important aspect of an object is its behaviour (the things it can do). A behaviour is initiated by sending a message to the object (usually by calling a method).
- The ability to use an operator or function in different ways in other words giving different meaning or functions to the operators or functions is called polymorphism.
- poly refers too many. That is a simple function or an operator functioning in many ways diffent upon the usage is called polymorphism.
- E.g. the message display details () of the person class should give different results when send to a student objet
- The ability to change form is known as polymorphism.

Scanned by TapScanner

There is two types of polymorphism in Java
- Compile time polymorphism (overloading)
- Runtime polymorphism (overriding)

What is RDBMS

- RDBMS stands for Relational Database Management system. RDBMS is the basis for SQL, and for all modern database systems like MS SQL server, IBM DB2, oracle, MySQL, and Microsoft Access.

- A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E.F. Codd.

- Most of today's database are relational.
  - database contains 1 or more tables
  - table contains 1 or more records
  - record contains 1 or more fields
  - fields contain the data

- So why is it called "relational"? tables are related (joined) based on common fields.

Example

Here's a simple database schema for tracking sales
- 3 tables, related by primary keys (CID, OID, PID)
- primary keys (in boldface) are unique record identifiers.

Scanned by TapScanner

• customer may place order for one product at a time...

customers	orders	products
*	*	*
CID	OID	PID
Firstname	CID	Name
Lastname	PID	Description
creditLimit	Quantity	Stock
Balance	Dateofpurchase	price
Notes		

what is SQL

- SQL is structured Query Language, whice is a computer language for storing, munipulating and retrieving data stored in relational database.

- SQL is the standard language for Relation database system. All relational database management systems like MySQL, MS Access, and oracle, sybase, Informix, postgres and SQL Server use SQL as standard database language.

- Also they are using different dialects, sychas:

• MS SQL server using T-SQL, ANSL SQL
• oracle using PL/SQL,
• MS Access version of SQL is called JET SQL (native format) etc.

Scanned by TapScanner

Write SQL Commands

DDL - Data Definition Language
DML - Data Manipulation Language
DCL - Data Control Language
DQL - Data Query Language

DQL - Data Query Language

Command	Description
SELECT	Retrieves certain records from one or more tables.

DML - Data Manipulation Language

Command	Description
INSERT	Creates a record
UPDATE	Modifies records
DELETE	Deletes records

DCL - Data Control Language

Command	Description
GRANT	Gives a privilege to user
REVOKE	Takes back privileges granted from user

~~Write SDLC phases~~
<u>What is SDLC</u>

- SDLC is a structure imposed on the ~~det~~ development of a software product that defines the process for planning, implementation, testing, documentation,

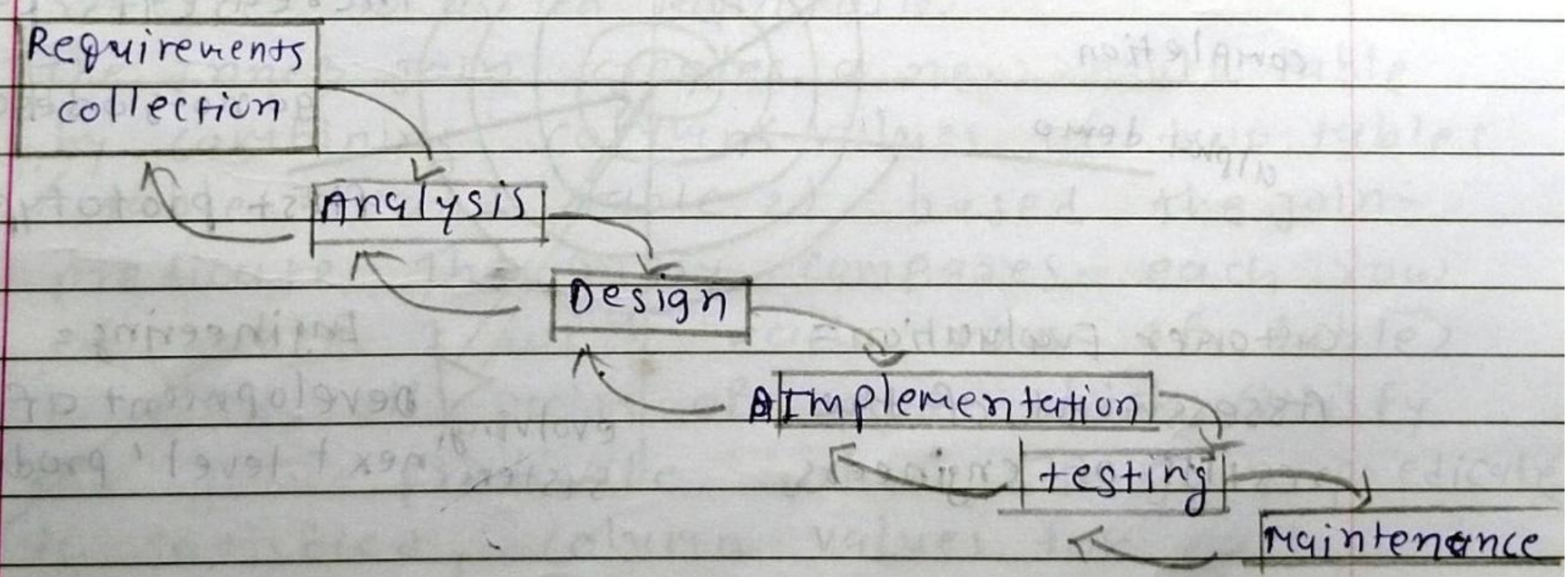
Scanned by TapScanner

deployment, and ongoing maintenance and support, There are a number of different devlopment, models.

- A software development Life cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.

- The methodology within the SDLC process can vary across industries and organizations, but standards such as ISO/IEC 12207 represent processes that establish a lifecycle for software, and provide a mode for the development, acquisition, and configuration of software systems.

Explain phases of the waterfall model



The classical software lifecycle models the software development as a step-by step "waterfall" between the various development phases.

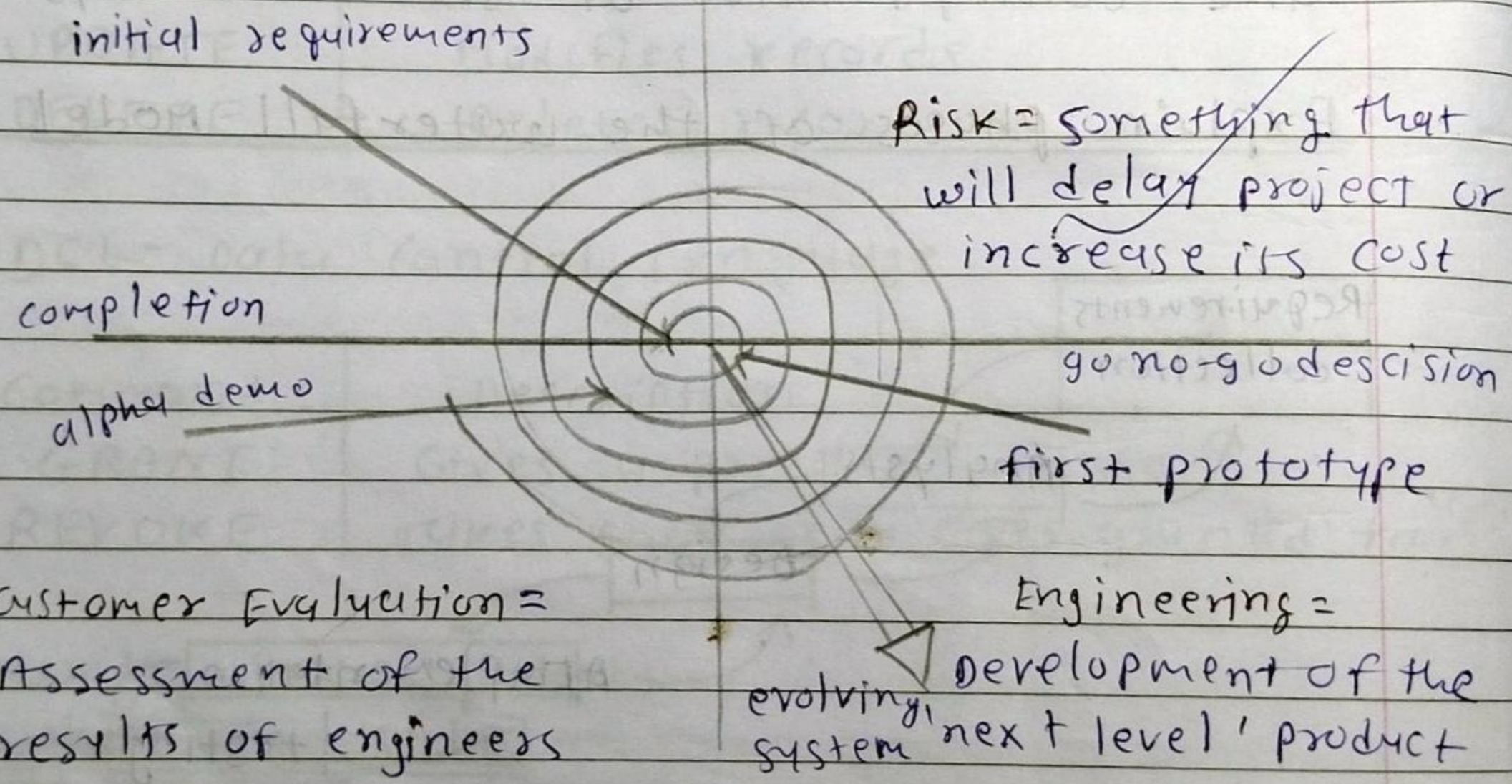
Applications (when to use?)

- Requirements are very well documented, clear and fixed.
- product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

write phases of spiral model

planning: determination of objectives, alternatives and constraints

Risk Analysis = Analysis of alternatives and identification/ resolution of risks.



~~write agile manifesto principls~~

What is join ?

A join is an SQL operation performed to establish a connection between two or more database tables based on matching columns, thereby creating a relationship between the tables. Most complex queries in an SQL database management system involve join commands. There are different types of joins.

Write type of joins.

Inner join : returns rows when there is a match in both tables.

- The most frequently used and important of the joins is the Inner Join. They are also referred to as an EQUIJOIN.
- The Inner Join creates a new result table by combining column values of two tables (table 1 and table 2) based the join-predicate. The quesy compares each row of table 1 with each row of table 2 to find all pairs of rows which satisfy the join-predicate. When the join-predicate is satisfied, column values for each matched pair of rows of A and B are combined into a result row.

Scanned by TapScanner

Syntax :-
The basic syntax of Inner join is as follows:
select table. column 1, table2, column 2...
from table 1 Inner Join table2 on table2.
common_filed = table2. common_field;

Left Join : returns all rows from the left table, even if there are no matches in the right table.

- The sql Left Join returns all rows from the left table, even if there are no matches in the right table. This means that if the on clause matches 0 (zero) records in right table, the Join will still return a row in the result, but with NULL in each column from right table.

- This means that a left join returns all the values from the left table, plus matched values from the right table or null in case of no matching Join predicate.

Syntax :-
The basic syntax of Right Join is as follows.

select table 1, Column 1, table2. column 2...
from table1 ~~Right~~ left Join table2 on table2.
common_filed = tabl2. common_field;

Right Join :- returns all rows from the right table, even if there are no matches in the left table.

Scanned by TapScanner

- The SQL Right Join returns all rows from the right table, even if there are no matches in the left table. This means that if the on clause matches 0 (zero) records in left table, the join will still return a row in the result, but with NULL in each column from left table.

- This means that a right join returns all the values from the right table, plus matched values from the left table or NULL in case of no matching join predicate.

syntax :-
select table 1. column, table 2. column 2....
From table 1 Right join table2 on table 1. common_filed = table 2. common_field;

Full Join : returns rows when there is a ~~much~~ in one of the tables. DDL - Data Definition Language.

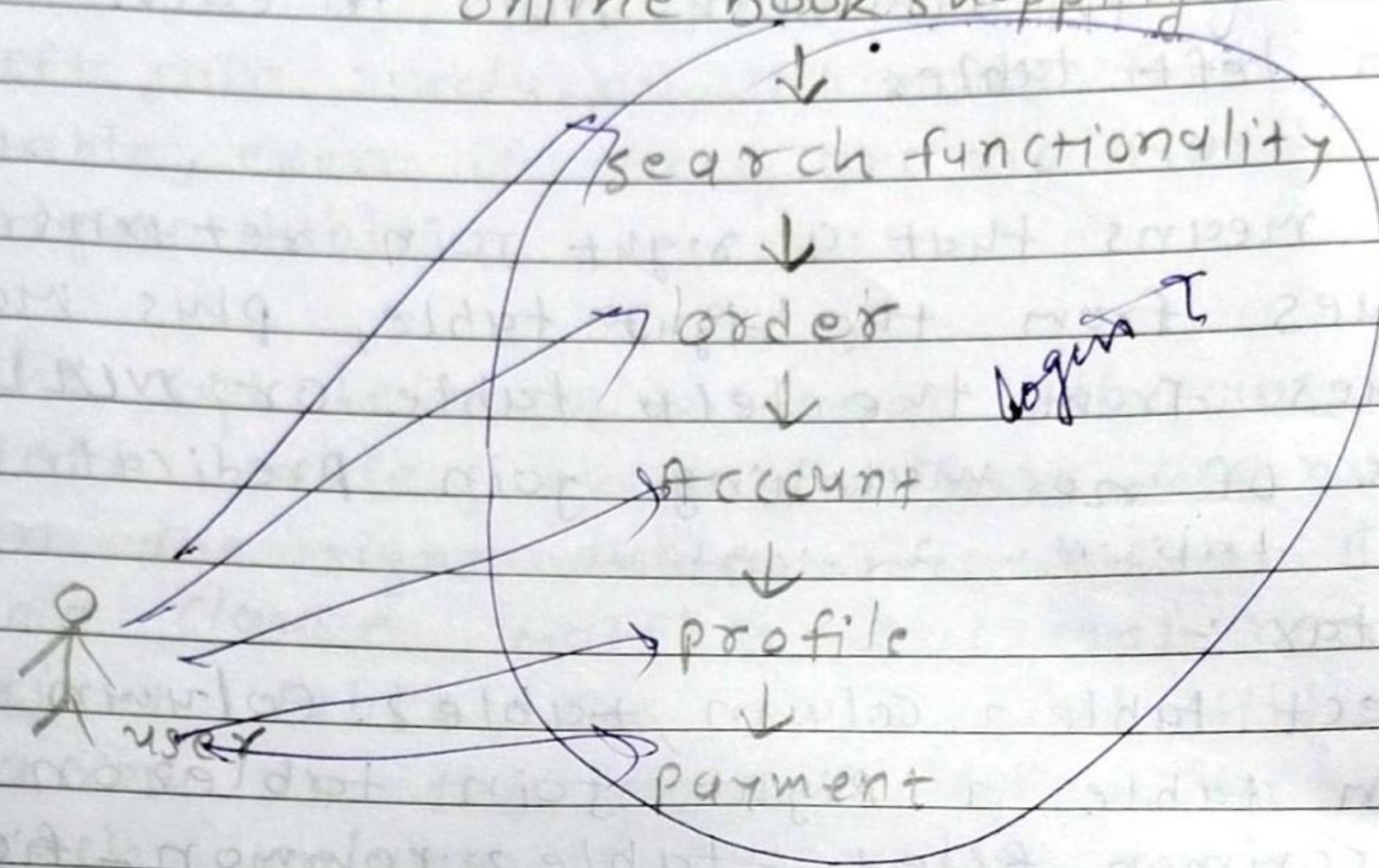
- The SQL full join combines the results of both left and right outer joins. The joined table will contain all records from both tables, and fill in NULL for missing matches on either side.

syntax :-
select table 1. column 1, table 2. column 2....
From table 1 Full join table on table 1. common_filed = table 2. common_field;

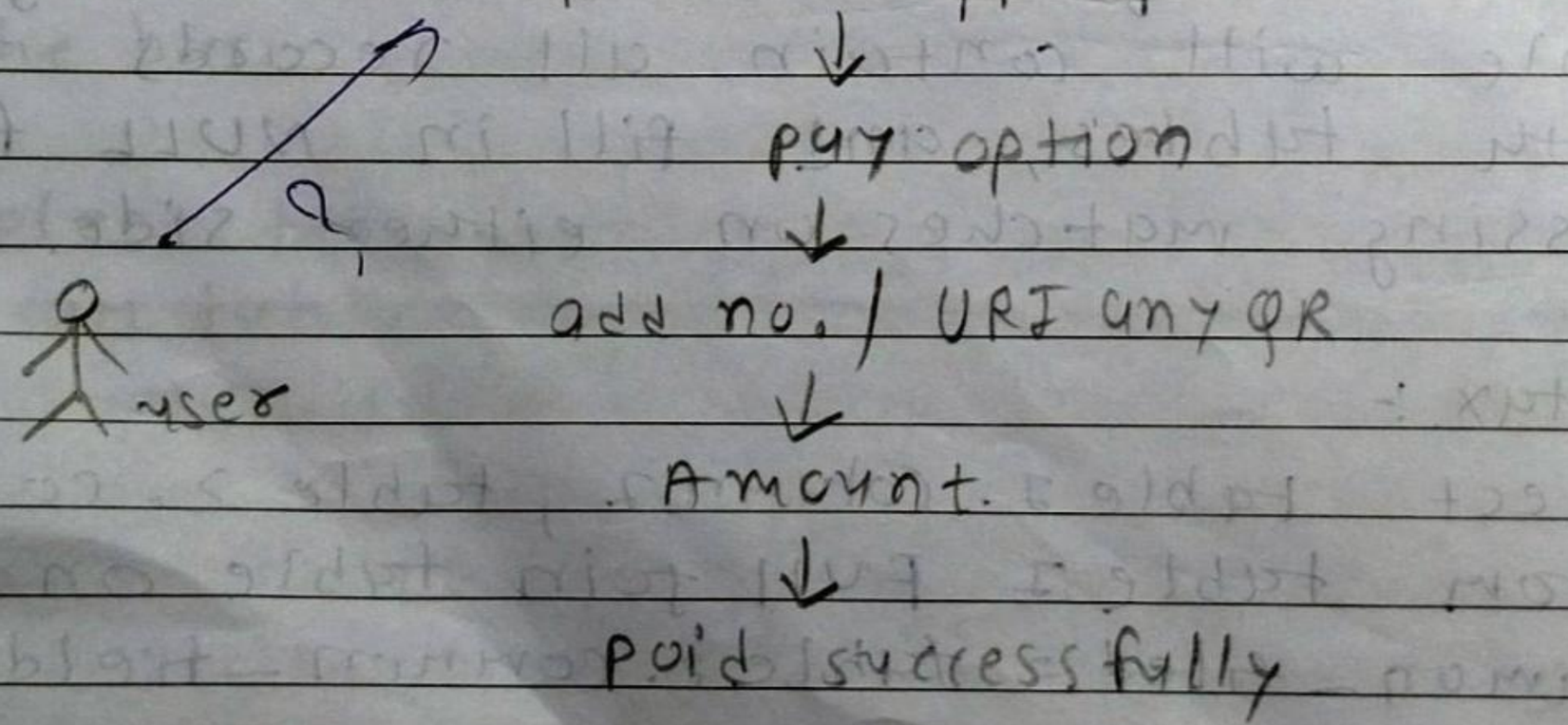
Scanned by TapScanner

Explain working methodology of agile mo
and also write pros and cons.

Draw Usecase on online book shopping

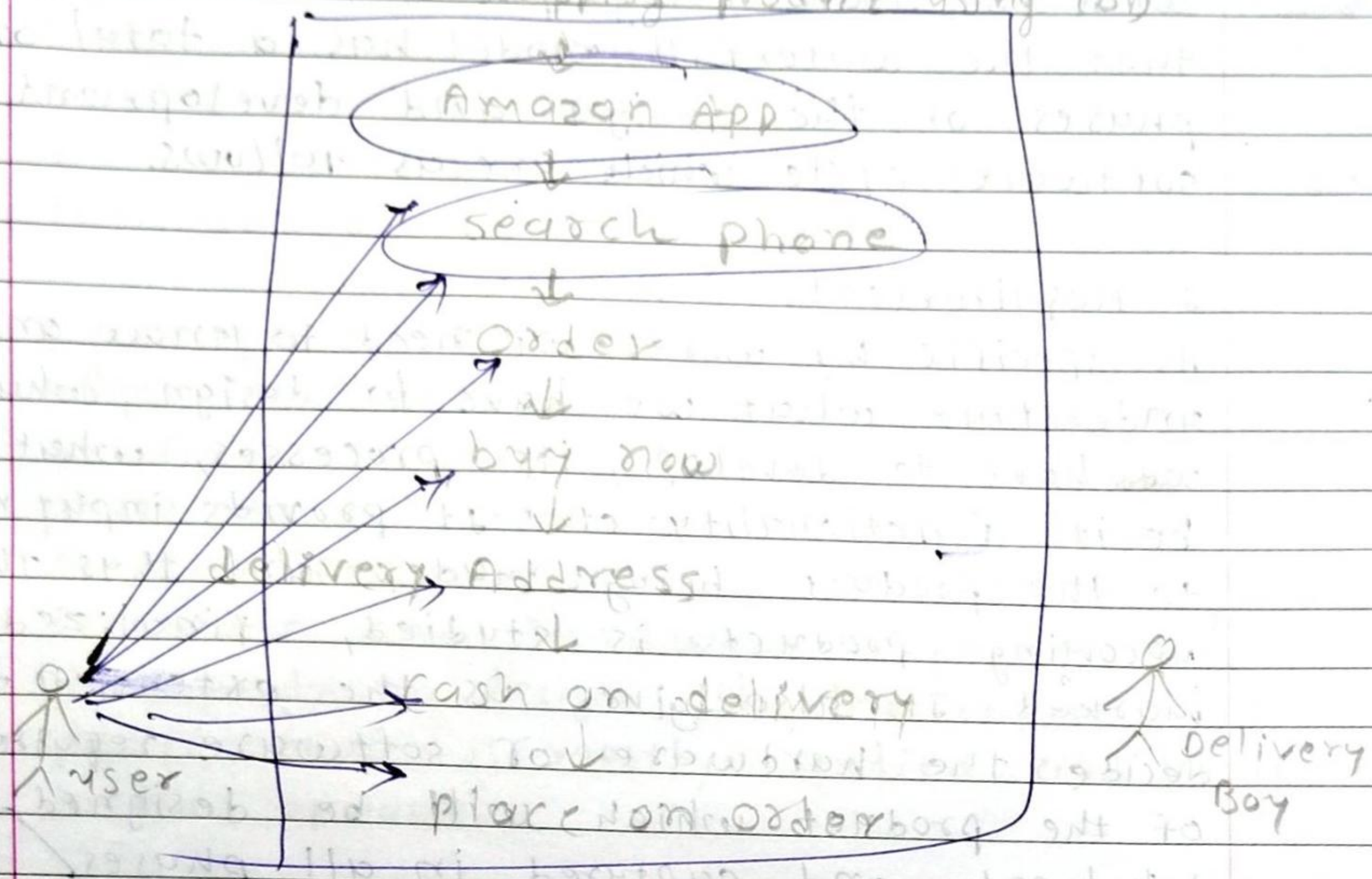


Draw Usecase on online bill payment system (paytm)

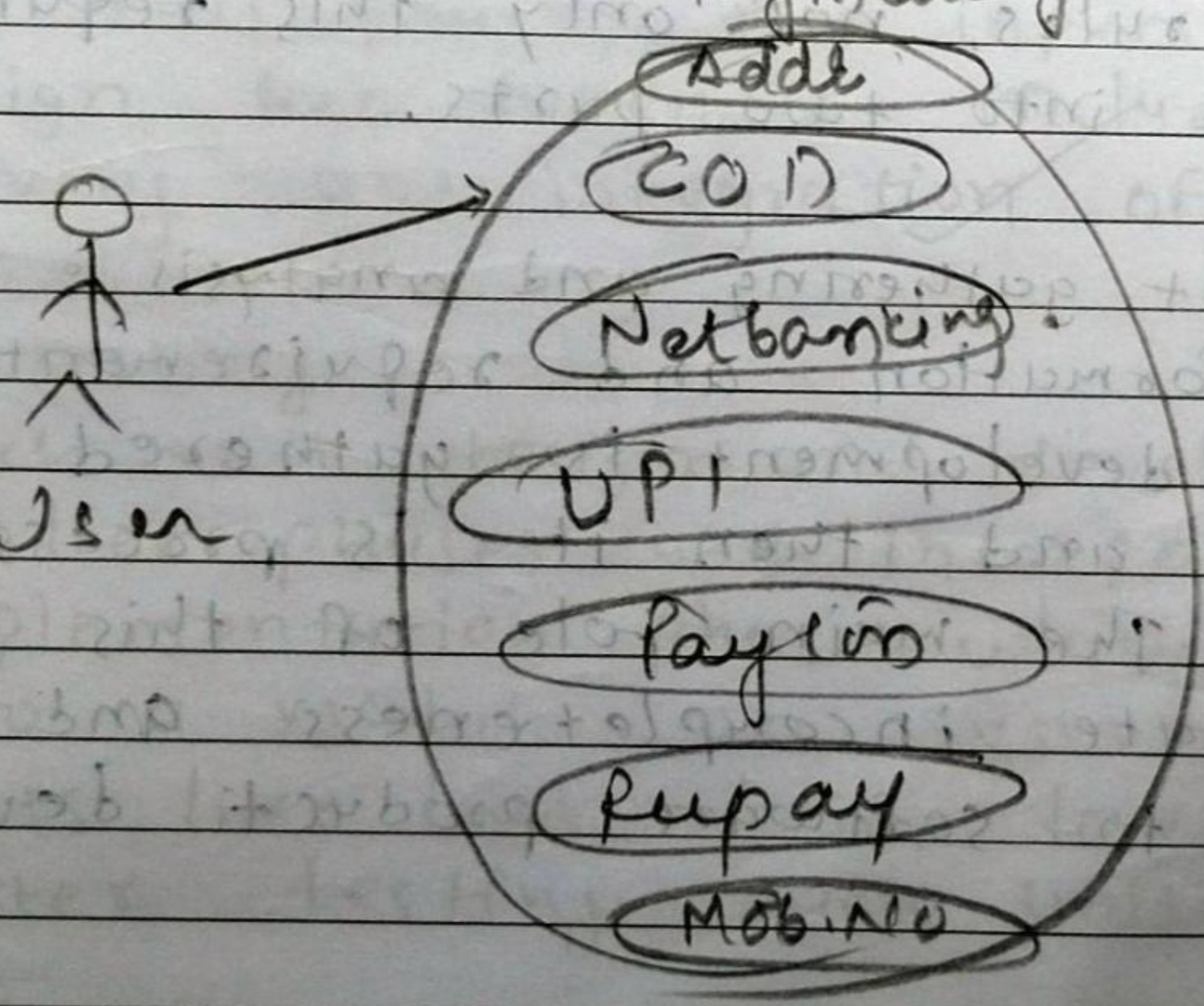


Scanned by TapScanner

Draw usecase on online shopping product using COD.



Draw usecase on online shopping using payment gateway.



(Continue) phuses of the waterfall Model.

with the above infographics, we can understand that the waterfall model has a total of 7 phases of the design and development software cycle which are as follows.

1. Requirements
As specific by words, we need to know and understand what we have to design, what we have to develop, its processes, what will be its functionality etc. It provides input material to the product being made, and thus the upcoming product is studied, finalized and marked. It also gives us the extension to decide the hardware or software requirements of the product which will be designed, developed, and captured in all phuses.

2. Analysis
It results in designing models, schema, and business rules. not only this requirement is divided into two parts.

Requirement gathering and analysis : first all the information and requirement for the product development is gathered from the customer and then it is processed for analysis. The main role of this part is to eradicate incompleteness and inconsistencies related to software product development

Requirement specification :- Then the above analyzed requirements are documented in an SRS document. It serves as a path between the customer and the SRS development team. Any disputes in the future are managed and settled through this SRS documentation only.

3. Design :-
After the first phase is completed and verified, it is the next most important phase to be studied as it is used for system design. It helps in specifying software and hardware requirements for the product design. So the requirement specification is mostly studied and verified in this phase.

4. Implementation
with system design fully verified, the implementation phase comes in a row. In this phase, the inputs from system design are taken. It includes the development proving and integration of the software.

5. Testing
Each unit design and development in the earlier phases are incorporated from the implementation phase which is integrated into a module or system for various tests like load test, load test.. etc.. after testing each unit.

Scanned by TapScanner

6. Maintenance

It is the last but the most important phase in the waterfall workflow model. This step comes just after Installtion, and it includes making the appropriate modification to the product or system, or to enhance, change or modify attributes related to performance issue related to ~~to~~ the system.

Scanned by TapScanner