V.E.S.
Since 1962

Vivekanand Education Society's Institute of Technology

NAME: NARENDER KESWANI

ROLL NO: 24

DIVISION: B

DEPARTMENT: MASTER OF COMPUTER APPLICATION (M.C.A)

SUBJECT: MCA14 - SOFTWARE PROJECT MANAGEMENT (SPM)

EXAM: CONTINOUS ASSESTMENT-2 (CA-2)

PROFESSOR: MONA DESHMUKH (MD)

DATE: 20/01/2022

PROBLEM STATEMENT:

WHICH OF THE FOLLOWING PROCESS MODELS WOULD YOU FOLLOWS FOR THE FOLLOWING PROJECTS? GIVE JUSTIFICATIONS.

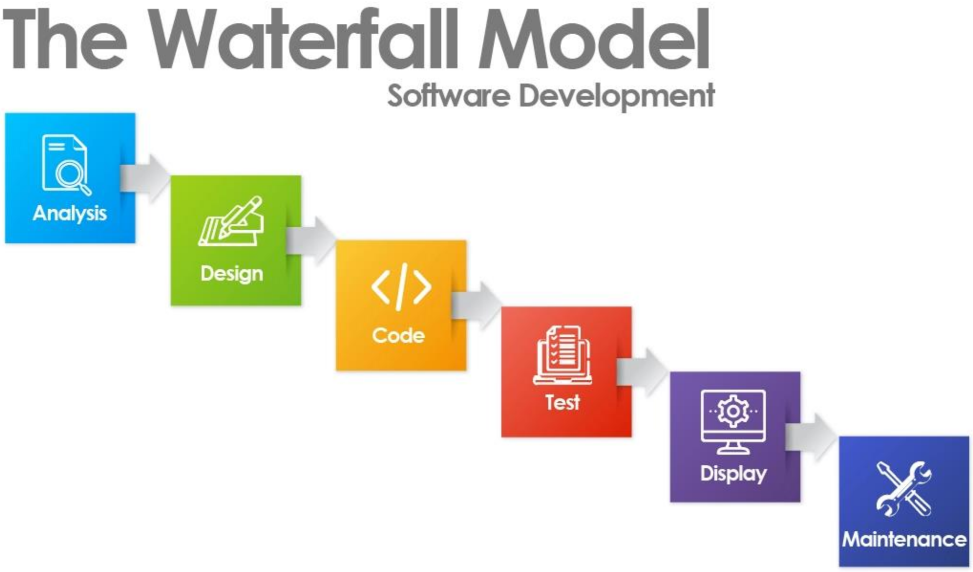
A) A SIMPLE DATA PROCESSING PROJECT
B) A NEW SYSTEM COMPARING FINGER PRINTS
C) AN ONLINE INVENTORY MANAGEMENT SYSTEM FOR AN AUTOMOBILE INDUSTRY
D) A NEW MISSILE TRACKING SYSTEM

SOLUTIONS:

A) A SIMPLE DATA PROCESSING PROJECT

SOLUTION:

For above project I would like to go for WATERFALL MODEL.



WATERFALL MODEL

LIFE CYCLE OF WATERFALL MODEL:

1) REQUIREMENTS ANALYSIS

The linear nature of the Waterfall methodology gives additional significance to this first stage in its life cycle. All requirements of the final software product's utility and features have to be gathered here.

Once the requirements analysis stage is completed the software development team should have all the information needed to complete the project without any, or minimal, further involvement from whoever has contracted or otherwise. initiated the project.

2) DESIGN:

The design phase is often divided into two subphases – logical (or preliminary) design and physical (or detailed) design. The logical or preliminary design phase involves putting all possible solutions on the table and analysing their strengths and weaknesses within the context.

Once theoretical ideas have been assessed and decisions taken on which to go with, the physical or detailed design phase is when they are documented and detailed as concrete specifications.

3) IMPLEMENTATION/CODING:

The rump of the project – when software developers write and assemble the actual code that turns the specifications detailed in the design phase into a functioning software system.

4) TESTING:

When the implantation phase has been fully completed, manual software testers (who might be supported by automated testing tools in contemporary software development projects) have to make sure every component of the software system works as intended, both autonomously and across any dependencies.
Testers will use documentation created at the design phase, user personas and user journey scenarios to run as many test cases as possible in the attempt to uncover any bugs that need to be fixed before deployment.

5) DEPLOYMENT:

When the software system has been tested and approved for release, a copy must be transferred from the software development environment and released in the live staging environment from which users will be able to access and use it. This stage is called deployment.
Team members responsible for deployment should be aware of any differences between the software development and live server environments and make any adjustments needed for the software to run the same way in both.

6) MAINTENANCE:

In the maintenance phase, the software is in use and the primary job is now to keep it available and running smoothly as well as fixing any bugs reported by users that may have been missed during the testing phase.

JUSTIFICATION:

In the given project, the requirement will be fixed and there is no near chance of changing it. Also for processing data, those basic operations will be fixed and we have delivered all the operations at a time. In waterfall model, we can very well implement all specified requirements and deliver the whole product at a time.

A simple data processing project has following features such as:

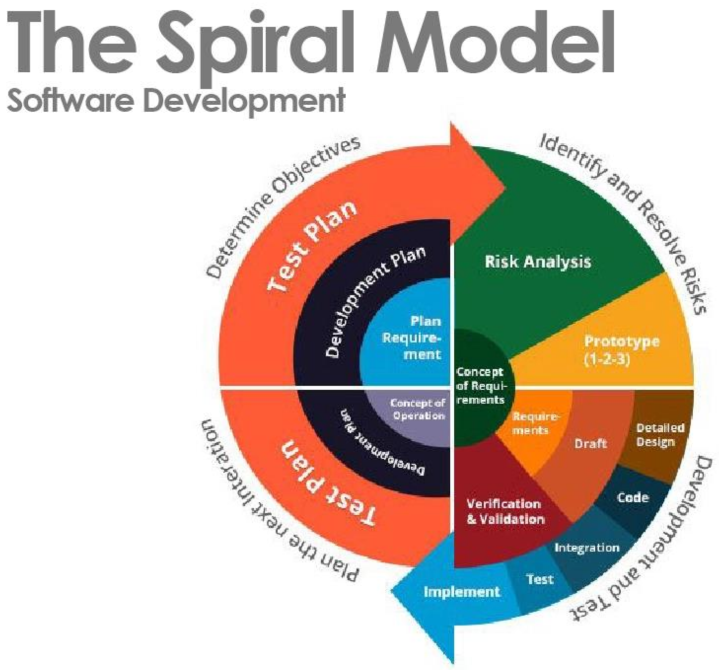
1) INSERTING RECORDS
2) VIEW RECORDS
3) UPDATE RECORDS
4) DELETE RECORDS

These features are fixed so we will go with WATERFALL MODEL.

B) A NEW SYSTEM COMPARING FINGER PRINTS.

SOLUTION:

For above project I would like to go for SPIRAL MODEL.



Spiral Model - Design

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

1) Identification:

This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.

2) Design:

The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

3) Construct or Build:

The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

4) Evaluation and Risk Analysis:

Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

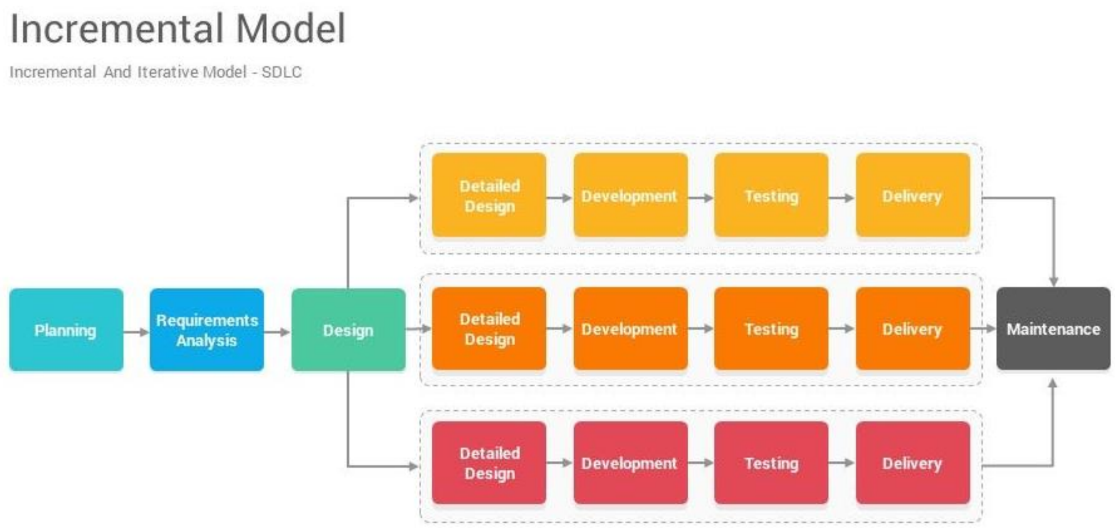
JUSTIFICATION:

In the given project, requirement i.e. finger print data will be added continuously so we need to consider it. Even customer feedback i.e. usefulness of the project has to be checked every time. In spiral model, for each set of requirements we can follow up six activities and in turn that spiral way will be continuing till the project is delivered.

C) AN ONLINE INVENTORY MANAGEMENT SYSTEM FOR AN AUTOMOBILE INDUSTRY.

SOLUTION:

For above project I would like to go for INCREMENTAL MODEL.



1) Requirement analysis:

In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.

2) Design & Development:

In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

3) Testing:

In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behaviour of each task.

4) Implementation:

Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product

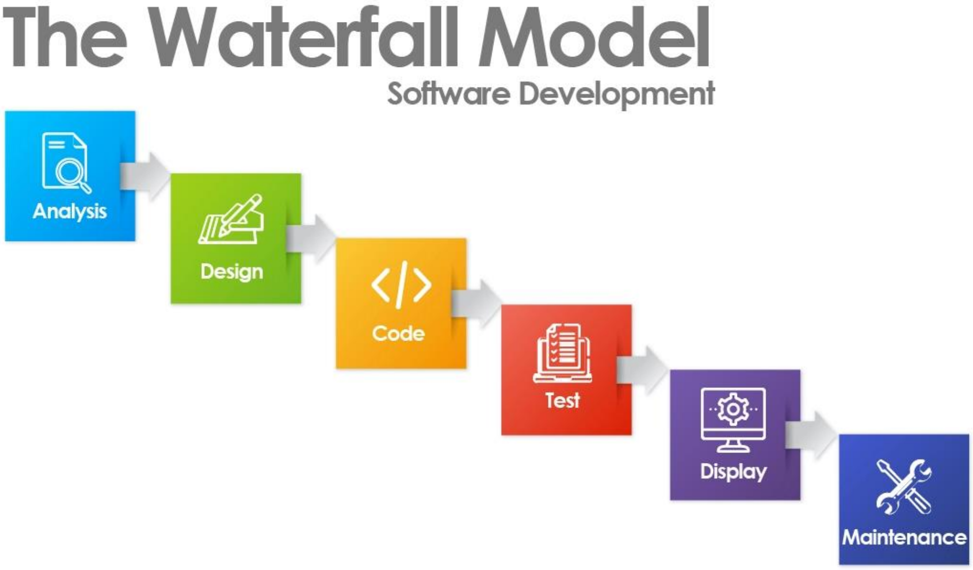
JUSTIFICATION:

In this project, we will be delivering new services in every increment. Hence, in online MIS system at a time, it is not possible to deliver the entire module and we cannot, even delay the delivery. So, for this reason, we can consider some features and deliver first increment. Later on, in each increment we can add new features and deliver the entire module increment by increment.

D) A NEW MISSILE TRACKING SYSTEM.

SOLUTION:

For this project I would like to go for WATERFALL MODEL.



WATERFALL MODEL

LIFE CYCLE OF WATERFALL MODEL:

1) REQUIREMENTS ANALYSIS

The linear nature of the Waterfall methodology gives additional significance to this first stage in its life cycle. All requirements of the final software product's utility and features have to be gathered here.
Once the requirements analysis stage is completed the software development team should have all the information needed to complete the project without any, or minimal, further involvement from whoever has contracted or otherwise. initiated the project.

2) DESIGN:

The design phase is often divided into two subphases – logical (or preliminary) design and physical (or detailed) design. The logical or preliminary design phase involves putting all possible solutions on the table and analysing their strengths and weaknesses within the context.
Once theoretical ideas have been assessed and decisions taken on which to go with, the physical or detailed design phase is when they are documented and detailed as concrete specifications.

3) IMPLEMENTATION/CODING:

The rump of the project – when software developers write and assemble the actual code that turns the specifications detailed in the design phase into a functioning software system.

4) TESTING:

When the implantation phase has been fully completed, manual software testers (who might be supported by automated testing tools in contemporary software development projects) have to make sure every component of the software system works as intended, both autonomously and across any dependencies.

Testers will use documentation created at the design phase, user personas and user journey scenarios to run as many test cases as possible in the attempt to uncover any bugs that need to be fixed before deployment.

5) DEPLOYMENT:

When the software system has been tested and approved for release, a copy must be transferred from the software development environment and released in the live staging environment from which users will be able to access and use it. This stage is called deployment.

Team members responsible for deployment should be aware of any differences between the software development and live server environments and make any adjustments needed for the software to run the same way in both.

6) MAINTENANCE:

In the maintenance phase, the software is in use and the primary job is now to keep it available and running smoothly as well as fixing any bugs reported by users that may have been missed during the testing phase.

JUSTIFICATION:

Here for missile tracking system the frequency range for tracking missiles will be fixed. All data will be provided at start only. Using this data, the entire system has to deliver at a time only so waterfall model is useful here.

CONCLUSION:

Based on research and study I conclude my solutions:

A) A SIMPLE DATA PROCESSING PROJECT
WATERFALL MODEL

B) A NEW SYSTEM COMPARING FINGER PRINTS
SPIRAL MODEL

C) AN ONLINE INVENTORY MANAGEMENT SYSTEM FOR AN AUTOMOBILE INDUSTRY
INCREMENTAL MODEL

D) A NEW MISSILE TRACKING SYSTEM
WATERFALL MODEL

This is what SDLC methodologies are about in a nutshell. The development of a software method ideally suited for your project may rely on factors such as timeline, cost, quality, etc but keep in mind that one solution may not fit in every scenario or business.
Every method has its advantages and disadvantages; some projects are better carried out with a waterfall strategy, whereas others may benefit from flexibility from agile or iterative models.