PRACTICAL-1

AIM:

List & draw at least 7 various Software Development Life Cycle (SDLC) models and preparing the detailed case study on "VLC Media Player" that which SDLC model is suitable to develop the "VLC Media Player" desktop application?

SOLUTION:

WHAT IS SDLC?

- SDLC is a process that defines the various stages involved in the development of software for delivering a high-quality product

- Basically, it is the series of phases (6 or 7), that takes a business case to completion.

WHY DO WE USE SDLC?

- The SDLC provides a framework that outlines the tasks that need to be performed in each phase of software development.

- Organizations can use the SDLC process to provide structure when designing and building of applications

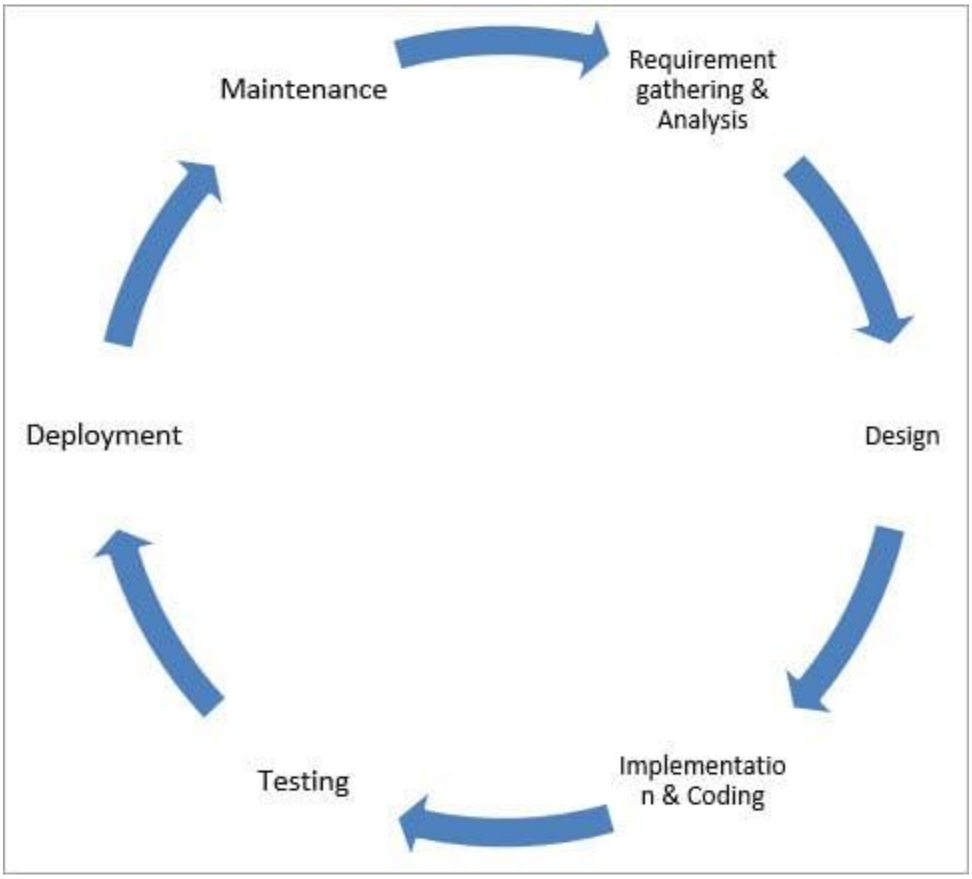
IMPORTANCE OF SDLC:

- It provides an effective framework and method to develop software applications.
- It helps in effectively planning before starting the actual development. SDLC allows developers to analyse the requirements.
- It helps in reducing unnecessary costs during development. During the initial phases, developers can estimate the costs and predict costly mistakes.

SDLC CYCLE:

- SDLC Cycle represents the process of developing software.

Below is the diagrammatic representation of the SDLC cycle:



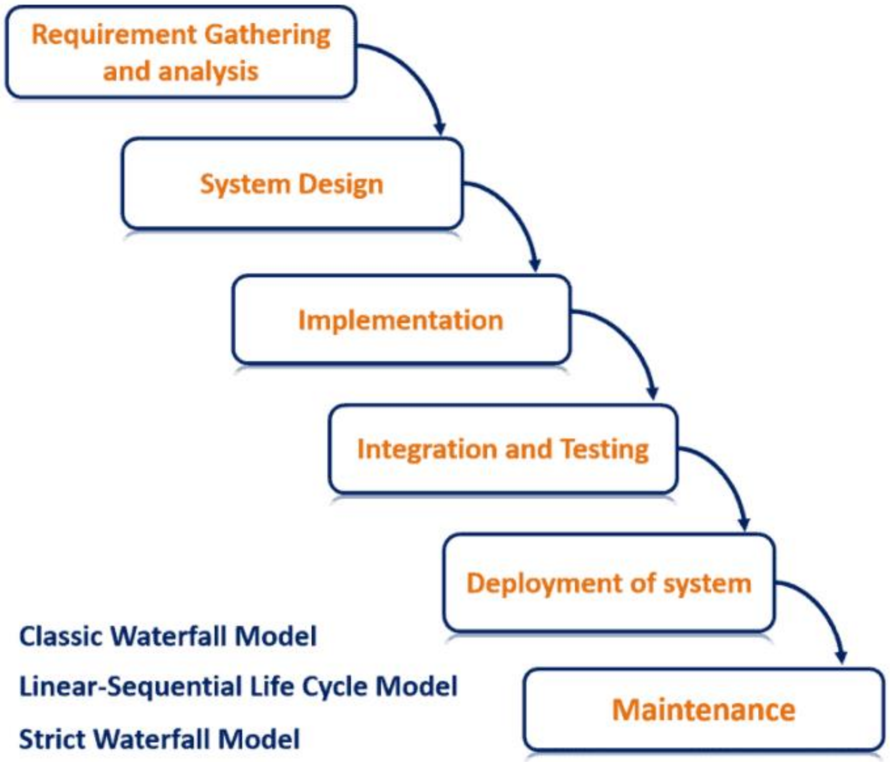
SDLC PHASES:

Given below are the various phases:

1. Requirement gathering and analysis
2. Design
3. Implementation or coding
4. Testing
5. Deployment
6. Maintenance

SDLC MODELS:

1. WATERFALL MODEL:



- Waterfall model is the very first model that is used in SDLC.
- The waterfall model is a reference model i.e. it is used as a reference for deriving other models.
- It is not practically implemented.
- It is also known as the linear sequential model.
- In this model, the outcome of one phase is the input for the next phase.

- Development of the next phase starts only when the previous phase is complete.

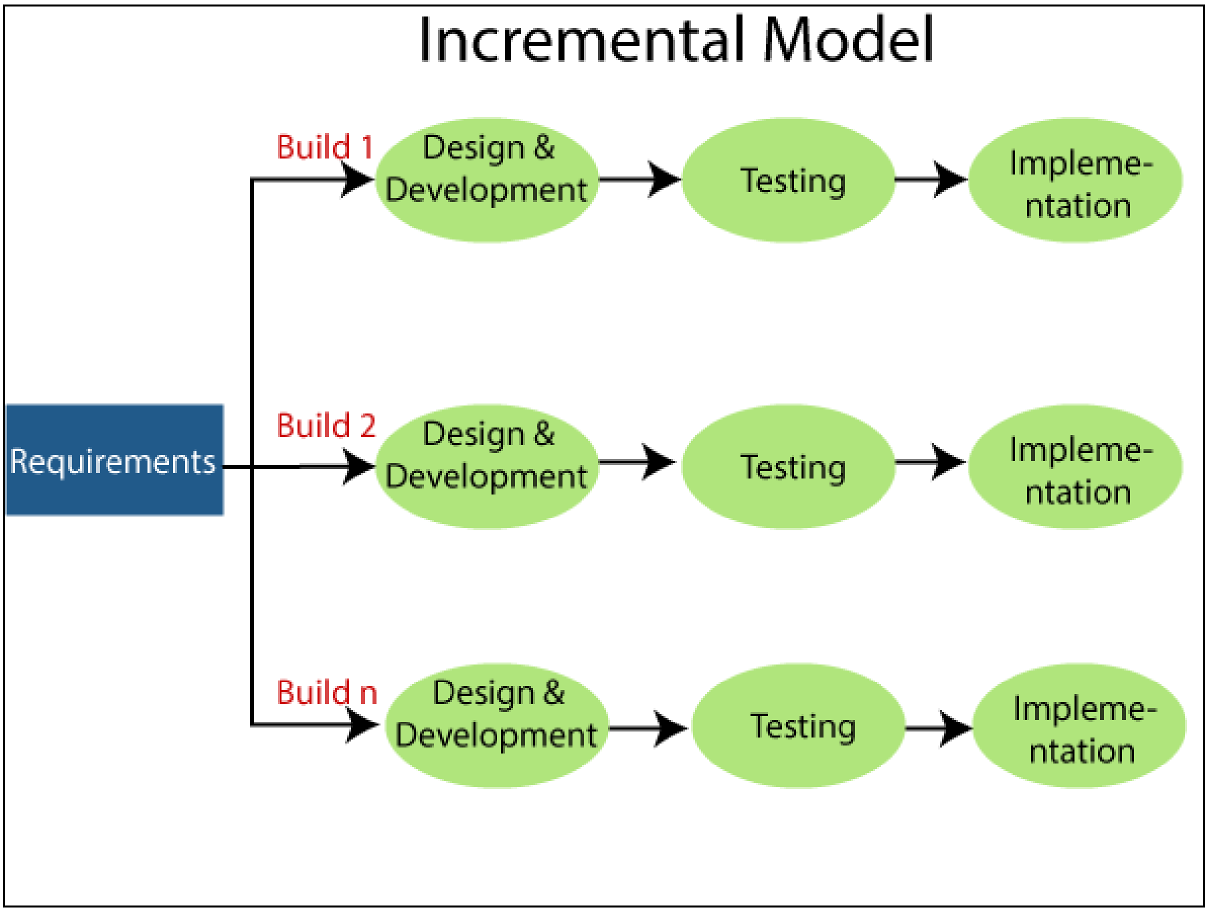
Advantages:

- Forces structured, disciplined organization.
- It is simple to understand, follow and arrange tasks.
- Deliverables of each phase are well defined, and this leads to no complexity and makes the project easily manageable.

Disadvantages:

- Not a good model for complex and object-oriented projects.
- No working software is produced until late during the life cycle.
- As the output of previous phase is input to next phase, so simultaneous work is not possible, which makes it time consuming model.

2. INCREMENTAL MODEL



- Incremental Model is a software development process where requirements are divided into several stand-alone software development modules.

- subsequent release of the module adds a feature to the previous release.

- The process will continue until the whole software is achieved.

Requirement analysis

- In the first step of the incremental model, the product analysis expertise identifies the functional requirements and non-functional requirements.

Design & Development

- In this phase of the SDLC's incremental model, the system functionality and design of the development methodology has ended with success.

Testing

- In the incremental model, the testing phase examines the performance of each existing function as well as additional functionality.

Implementation

- In the implementation phase, coding is done for developing software.
- The design of the software, which is made in the designing phase, is now implemented practically, and final coding is done.
- Upon completion of this process, the quality of the product working will be enhanced and upgrade to the final system product.

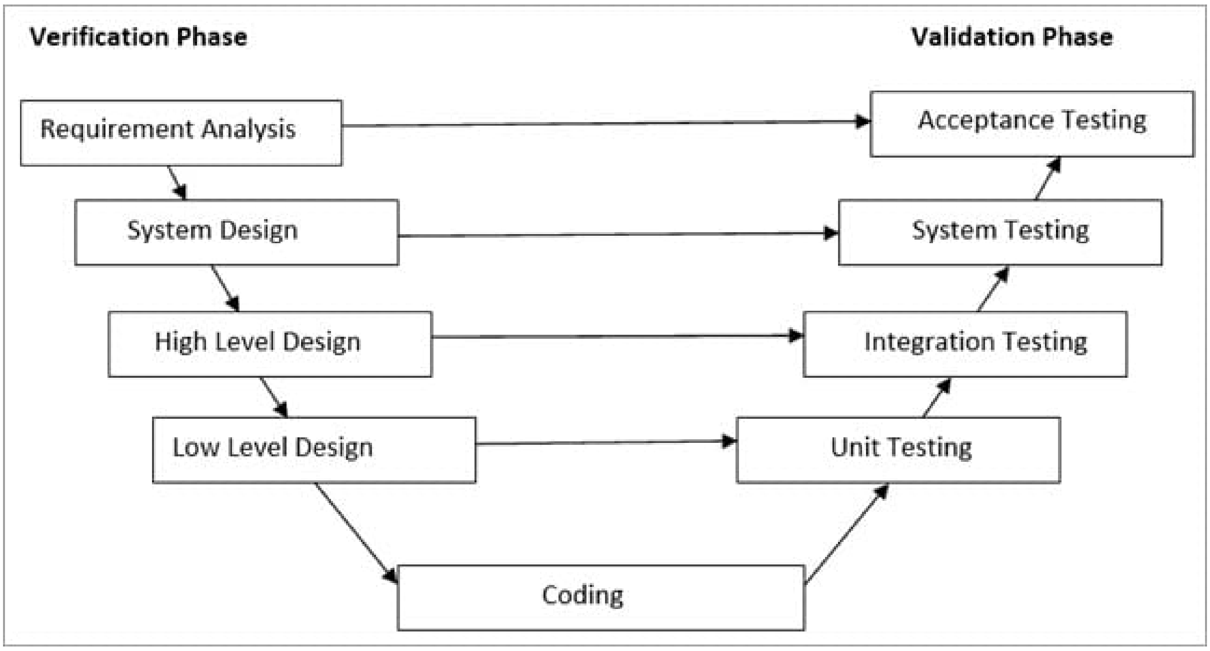
Advantages:

- Errors are easy to detect.
- Easy to test and debug.
- Flexible.
- Easy to manage risk because it has been managed through iteration.
- The client is provided with significant functionality at an early stage.

Disadvantages:

- It needs good planning.
- The total cost is high.
- It needs well-defined module interfaces.

3. V-SHAPED MODEL:



- V- Model is also known as Verification and Validation Model.
- In this model Verification & Validation goes hand in hand i.e. development and testing goes parallel.
- V model and waterfall model are the same except the only difference is that the test planning and testing start at an early stage in V-Model.

Verification Phase:

(i) **Requirement Analysis:**

- In this phase, all the required information is gathered & analysed.
- Verification activities include reviewing the requirements.

(ii) **System Design:**

- Once the requirement is clear, a system is designed i.e. architecture, components of the product are created and documented in a design document.

(iii) **High-Level Design:**

- High-level design defines the architecture/design of modules.
- It defines the functionality between the two modules.

(iv) **Low-Level Design:**

- Low-level Design defines the architecture/design of individual components.

(v) **Coding:**

- Code development is done in this phase.

Validation Phase:

(i) **Unit Testing:**

- Unit testing is performed using the unit test cases that are designed and is done in the Low-level design phase.
- Unit testing is performed by the developer itself. It is performed on individual components which lead to early defect detection.

(ii) **Integration Testing:**

- Integration testing is performed using integration test cases in High-level Design phase.
- Integration testing is the testing that is done on integrated modules.
- It is performed by testers.

(iii) **System Testing:**

- System testing is performed in the System Design phase.
- In this phase, the complete system is tested i.e. the entire system functionality is tested.

(iv) **Acceptance Testing:**

- Acceptance testing is associated with the Requirement Analysis phase and is done in the customer's environment.

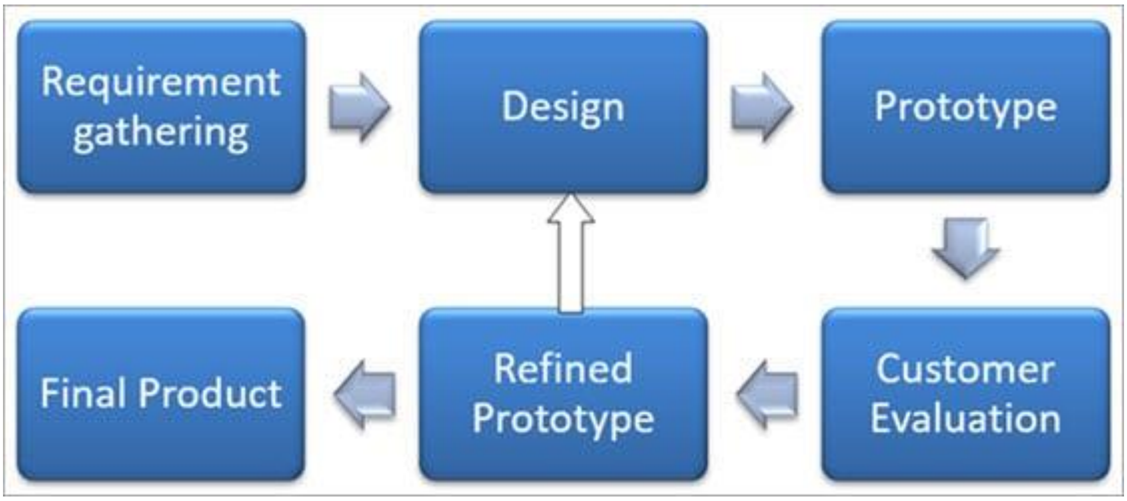
Advantages:

- It is a simple and easily understandable model.
- V –model approach is good for smaller projects wherein the requirement is defined and it freezes in the early stage.
- It is a systematic and disciplined model which results in a high-quality product.

Disadvantages:

- V-shaped model is not good for ongoing projects.
- Requirement change at the later stage would cost too high.

4. PROTOTYPE MODEL:



- The prototype model is a model in which the prototype is developed prior to the actual software.
- Prototype models have limited functional capabilities and inefficient performance when compared to the actual software.
- Dummy functions are used to create prototypes.
- This is a valuable mechanism for understanding the customers' needs.
- Feedbacks are implemented and the prototype is again reviewed by the customer for any change.
- This process goes on until the model is accepted by the customer.
- Customer feedback and the refined requirement is used to modify the prototype and is again presented to the customer for evaluation.

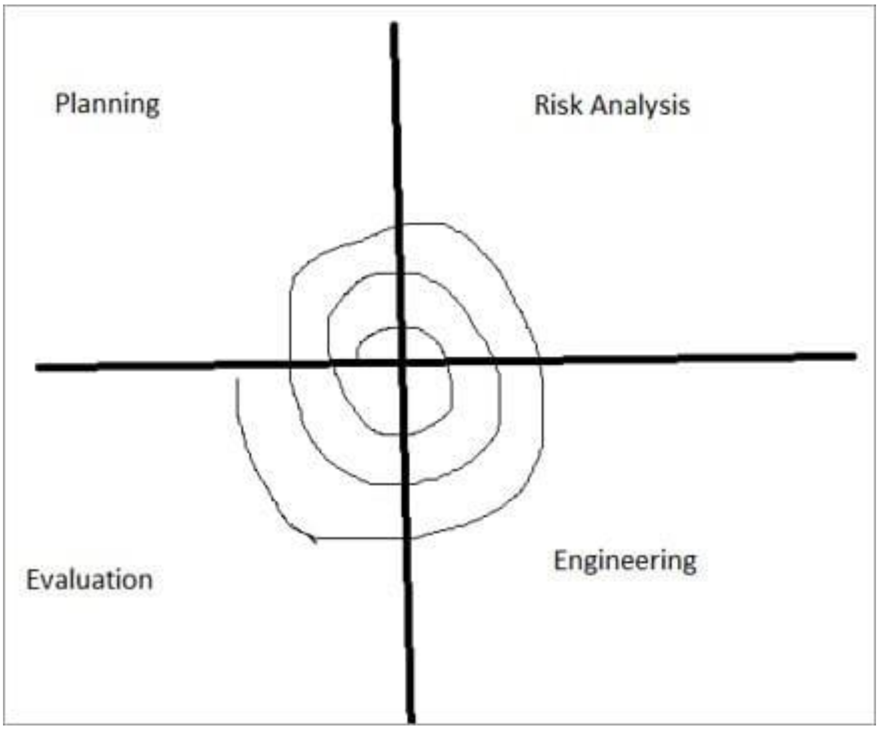
Advantages:

- Cost and Time efficient
- Defects can be found out early and easily
- Customer is actively involved.

Disadvantages:

- Customer involvement may affect the speed as their requirements may frequently change.
- It sometimes leads to waste of time and money.

5. SPIRAL MODEL:



- The Spiral Model includes iterative and prototype approach.
- Spiral model phases are followed in the iterations.
- Spiral Model has four phases:
  - Planning
  - Risk Analysis
  - Engineering
  - Evaluation

(i) **Planning:**

- The planning phase includes requirement gathering wherein all the required information is gathered from the customer and is documented.
- Software requirement specification document is created for the next phase.

(ii) **Risk Analysis:**

- In this phase, the best solution is selected for the risks involved and analysis is done by building the prototype.

(iii) **Engineering:**

- Once the risk analysis is done, coding and testing are done.

(iv) **Evaluation:**

- Customer evaluates the developed system and plans for the next iteration.

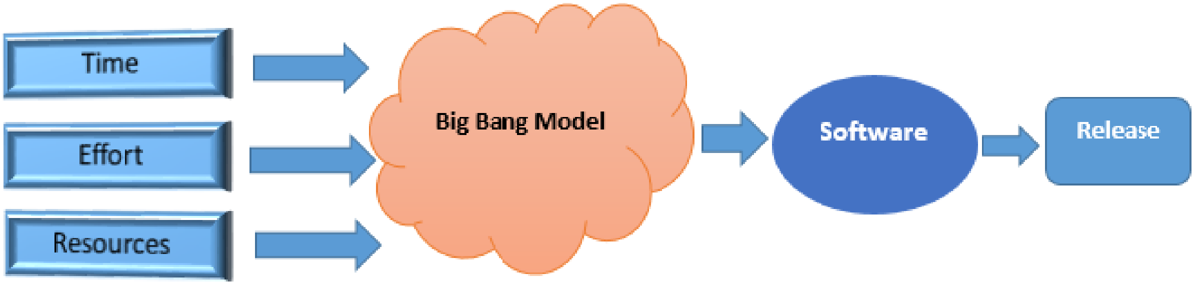
Advantages:

- Risk Analysis is done extensively using the prototype models.
- Any enhancement or change in the functionality can be done in the next iteration.

Disadvantages:

- The spiral model is best suited for large projects only.
- The cost can be high as it might take a large number of iterations which can lead to high time to reach the final product.

6. BIG BANG MODEL:



Big Bang SDLC MODEL

- Big Bang Model does not have any defined process.
- Money and efforts are put together as the input and output come as a developed product which might be or might not be the same as what the customer needs.
- Big Bang Model does not require much planning and scheduling. The developer does the requirement analysis & coding and develops the product as per his understanding.

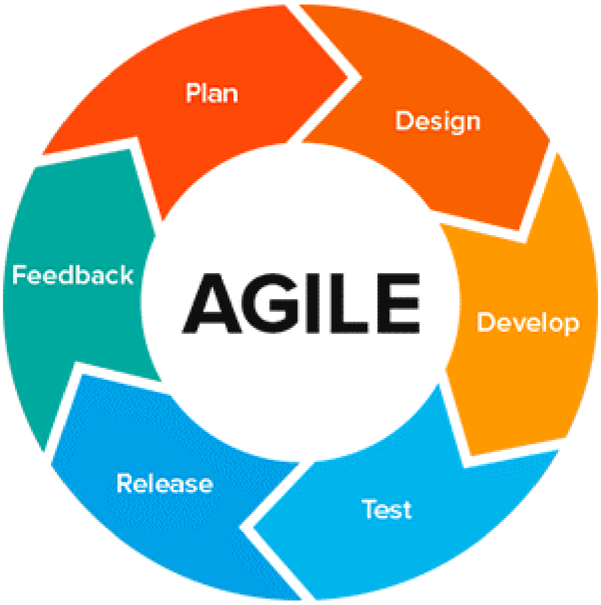
Advantages:

- It’s a very simple Model.
- Less Planning and scheduling is required.
- The developer has the flexibility to build the software of their own.

Disadvantages:

- It can be used only for small projects
- High risk and uncertainty.

7. AGILE MODEL



- Agile Model is a combination of the Iterative and incremental model.
- This model focuses more on flexibility while developing a product rather than on the requirement.
- In Agile, a product is broken into small incremental builds.
- It is not developed as a complete product in one go.
- Each build increments in terms of features. The next build is built on previous functionality.
- In agile iterations are termed as sprints. Each sprint lasts for2-4 weeks.
- At the end of each sprint, the product owner verifies the product and after his approval, it is delivered to the customer.
- Customer feedback is taken for improvement and his suggestions and enhancement are worked on in the next sprint.
- Testing is done in each sprint to minimize the risk of any failures.

Advantages:

- It allows more flexibility to adapt to the changes.
- The new feature can be added easily.
- Customer satisfaction as the feedback and suggestions are taken at every stage.

Disadvantages:

- Lack of documentation.
- Agile needs experienced and highly skilled resources.
- If a customer is not clear about how exactly they want the product to be, then the project would fail.

CASE STUDY ON VLC MEDIA PLAYER

BUSINESS REQUIREMENT:

Need to create an application which is a multipurpose media player specifically designed for desktops.

BASIC PURPOSE:

The application should support and run the major digital formats of audio and video.

The application should give user full control on the audio/video played by providing the basic functionalities like pause, play, rewind, etc.

The application should be updated on regular basis.

PROPOSED SDLC MODEL: SPIRAL MODEL

WHY SPIRAL MODEL?

We know that spiral model has 4 phases.

- Identification
- Design
- Build
- Evaluation

Firstly, talking about the step 1, we have already identified the purpose and also, requirement gathering is also completed. Then, the design phase starts with conceptual design or basic design that is sufficient to fulfill the basic functionalities that the client requires to be satisfied. The design will get mature in the subsequent spirals. In the third phase, where the actual development takes place, initially in the first spiral more focus will be laid to satisfy the minimum requirements of the client, which in this case is providing user the functionalities to play, pause, rewind, replay, etc. More functionalities, secondary functions and bug fixes take place in the subsequent spiral.

In the final phase, Risk Analysis takes place. It 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.

This is the end of the first spiral. After this, subsequent spirals will start with the aim of making the application mature and to give regular updates to the application, featuring new controls, settings, functionalities and most important, bug fixes.

To conclude, the vlc media player can be developed using spiral model because of the following reasons:

- The project is for long term.
- There will be frequent releases.
- Significant changes are expected in the product during the development.
- New product line will have to be released in phases to get enough customer feedback.
- The project falls in medium to high risk category.

CONCLUSION:

- By performing the above practical, I learned about the basics of SOFTWARE DEVELOPMENT LIFE CYCLE and also, made a small case study.