UNIVERSITY of GREENWICH
Alliance with FPT Education

ĐẠI HỌC GREENWICH (VIỆT NAM)
Liên kết với Tổ chức Giáo dục FPT

ASSIGNMENT SOFTWARE DEVELOPMENT LIFE CYCLE REPORT

Student performance: TRAN QUANG HUY

ID: GCD18457

Class: GCD0822

Teacher: NGUYEN DUY NGHIEM

ASSIGNMENT 1 FRONT SHEET

Qualification	**BTEC Levels 4 and 5 Higher Nationals in Computing**		
Unit number and title	**Unit 9: Software Development Lifecycles**		
Assignment due		**Assignment submitted**	1
Learner's name	Tran Quang Huy	**Assessor name**	Do Quoc Binh

Learner declaration: I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.			
Learner signature	Huy	**Date**	20/02/2019

Grading grid

P	M	D

Assignment title	**Assignment 1: Planning a Software Development Lifecycle**

<table>
<tr><th colspan="3">Learning Outcomes and Assessment Criteria</th></tr>
<tr><th>Pass</th><th>Merit</th><th>Distinction</th></tr>
<tr><td colspan="2"><b>LO1</b> Describe different software development lifecycles</td><td rowspan="2"><b>D1</b> Assess the merits of applying the Waterfall lifecycle model to a large software development project.</td></tr>
<tr><td><b>P1</b> Describe two iterative and two sequential software lifecycle models.
<b>P2</b> Explain how risk is managed in the Spiral lifecycle model.</td><td><b>M1</b> Describe, with an example, why a particular lifecycle model is selected for a development environment.</td></tr>
<tr><td colspan="2"><b>LO2</b> Explain the importance of a feasibility study</td><td rowspan="2"><b>D2</b> Assess the impact of different feasibility criteria on a software investigation.</td></tr>
<tr><td><b>P3</b> Explain the purpose of a feasibility report.
<b>P4</b> Describe how technical solutions can be compared.</td><td><b>M2</b> Discuss the components of a feasibility report.</td></tr>
</table>

Summative feedback:			
Assessor’s signature		**Date**	

Contents

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

SDLC stands for Software development life cycle. It is a process that describes how to develop, design and maintain the software project ensuring that all the functional & user requirement, goals and objective are met. This methodology improves the quality of the software project and over all process of software development and determining how an information system (IS) can support business needs, designing the system, building it, and delivering it to users. If you have taken a programming class or have programmed on your own, this probably sounds pretty simple. In the real world, however, it is not so easy.

The key person in the SDLC is the systems analyst, who analyzes the business situation, identifies opportunities for improvements, and designs an information system to implement the improvements. Many systems analysts view their profession as one of the most interesting, exciting, and challenging jobs around. As a systems analyst, you will work as a team with a variety of people, including business and technical experts. You will feel the satisfaction of seeing systems that you designed and developed make a significant positive business impact, while knowing that your unique skills helped make that happen.

The document is written for the benefit of education about Software development Life Cycle to Tune Source – a music company looking to expand their system features for users. This project has been initiated to increase sales by creating the capability of selling digital music downloads to customers through kiosks in our stores, and over the Internet using their website.

PART I: The Different software development lifecycles

1. Two iterative and two sequential software lifecycle models

1.1. Iterative model

1.1.1. Overview

Iterative model is a model in software development life cycle (SDLC), it does not start with a fully developed software in mind. Instead, the developments begin by specifying and implementing parts of software. After a fixed period of time, the software is reviewed and further developed that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. In the iterative model, the process will start by simply executing a small set of software requirements, then it is enhanced to repeat in subsequent versions until a complete system is built. Fully built and ready to release. Each step / milestone of the iterative model is developed over a specific time period, called iteration. (Base on: (Powell-Morse, 2016))

- Spiral model:
    - Spiral model is a risk-oriented development process for software projects. The combination of the strengths of other models and the difficulty of the previous models still exist. Based on the specific risk models of each project, the spiral model shows how to apply the elements of one or more processing models, such as acceleration models, waterfall models or models, evolutionary prototyping.
    - The process is divided into several iterative steps, each step begins with planning, risk analysis, prototyping, system improvement and development, retesting and sequencing so on. (Wikipedia, n.d.)

Advantages	Disadvantages
The model converges good features and overcomes weaknesses of many other development models encountered.	Risk analysis is quite expensive, mainly applied to large projects with financial potential
Monitoring projects is easy and effective	Requiring a time-varying change leads to infinite, complex iterations, a team of specialists in risk analysis is required
Very suitable for high risk projects and risk reduction, dealing with changes in project implementation	Not widely used as waterfall model, prototype.
Predict the actual deadlines and costs	

Table 1- Advantage and Disadvantages Spiral model (Wikipedia, n.d.)

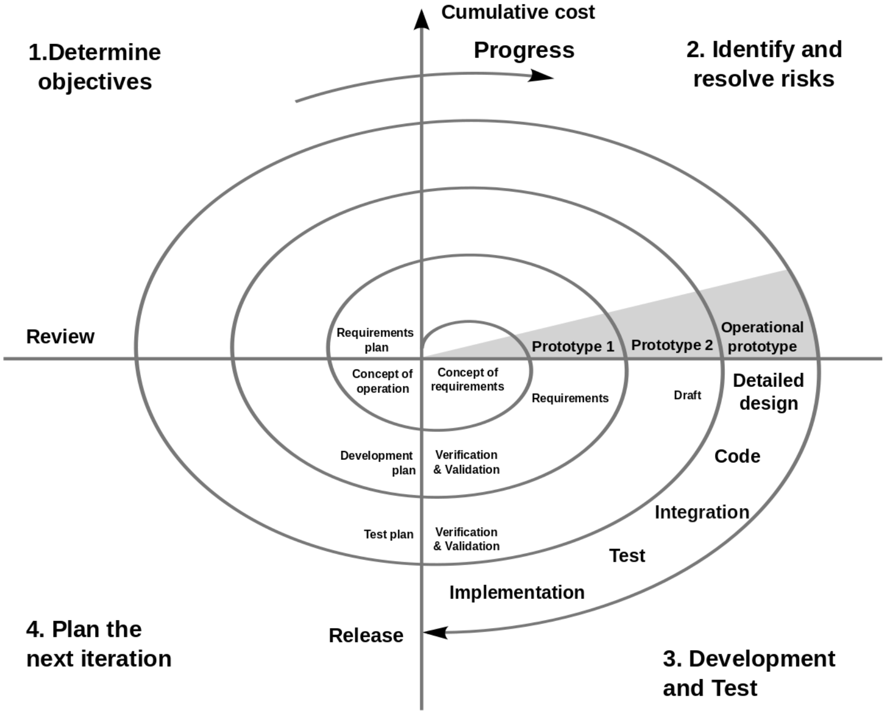

Figure 1 - Spiral model (Wikipedia, n.d.)

- DSDM Model:
  - Dynamic systems development method (DSDM) is an agile project delivery framework, initially used as a software development method First released in 1994, DSDM originally sought to provide some discipline to the rapid application development (RAD) method. (Wikipedia, n.d.)
  - The DSDM project consists of 7 phased steps which are organized and embedded in a rich set of roles and responsibilities and are supported by several core techniques.
    - Pre-Project
    - Feasibility Study
    - Business Study
    - Functional Model Iteration
    - Design & Build Iteration
    - Implementation
    - Post-Project

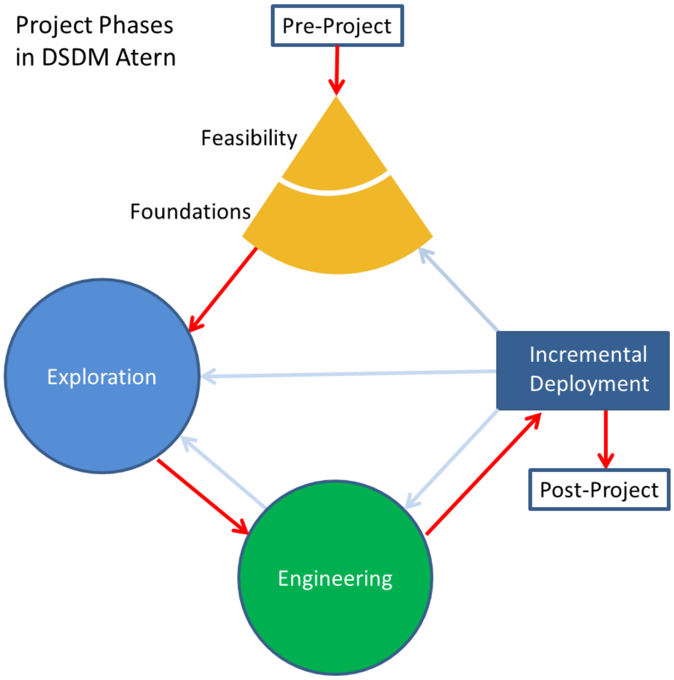

Figure 2 - DSDM model (Wikipedia, n.d.)

Advantages	Disadvantages
Decrease the time required to avail some system features.	Scalability.
Face to face communication and continuous inputs from customer representative leaves no space for guesswork.	The ability and collaboration of the customer to express user needs.
The end result is the high-quality software in the least possible time duration and satisfied customer.	Documentation is done at later stages.
	Reduce the usability of components.
	Needs special skills for the team

Table 2 - Advantages and Disadvantage DSDM model (Wikipedia, n.d.)

1.1.2. When to use Iterative model?

The Iterative model is used in the following scenarios:

- The requirement of the system is defined and understood clearly.
- New technology is being used and is being learned by the development team.
- There are some high-risk features and goals, which might change in the future.
- The resources with needed skill sets are unavailable and are planned to be used on a contract basic for specific iterations.
  (Base on: Professionalqa, 2018)

1.1.3. Process of Iterative Model:

In this process, once the initial planning is complete, a handful of phases are repeated again and again, with the completion of each cycle incrementally improving and iterating on the software.

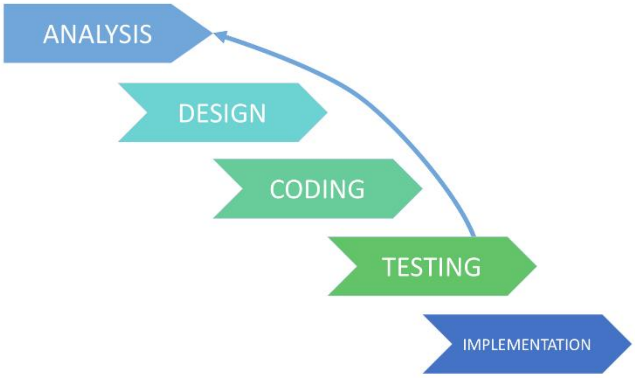

Figure 3 - Iterative SDLC Model (Existek, 2017)

Other phases of the iterative model are described below:

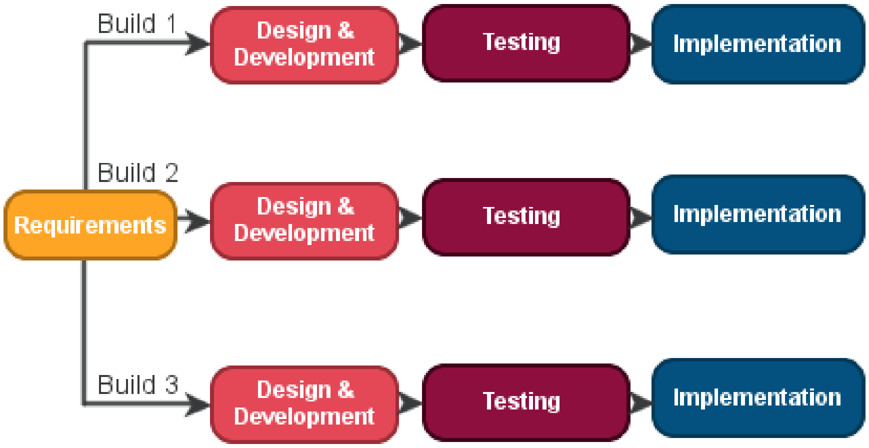

Figure 4 - Other phases of iterative model (Professionalqa, 2018)

1.1.4. Advantages and Disadvantages of Iterative Model:

Advantages	Disadvantages
Some functions can be quickly developed at the beginning of the development lifecycle	Iterative model requires more resources than the waterfall model
The paralleled development can be applied	Constant management is required
The progress is easy measurable	It is not suitable for smaller projects.
Testing and debugging during smaller iteration is easy.	The process is difficult to manage
In iterative model less time is spent on documenting and more time is given for designing.	Project progress is highly dependent upon the risk analysis phase.
One can get reliable user feedback, when presenting sketches and blueprints of the product to users for their feedback.	Defining increments may require definition of the complete system.

Table 3 - Advantages and Disadvantages of iterative model (Professionalqa, 2018) (Existek, 2017)

1.2. Sequential model

1.2.1. Overview

On the other hand, Sequential models such as Waterfall or V-Model is the result of intensive periods of collecting and refining product prerequisites before development can take place. The products that are developed by this model are expected to be finished/completed upon release. Central to the approach is an assumption that by adhering to the requirements captured at the outset, the product will be fulfilled the requirements of customers. (Roth et al., 2013)

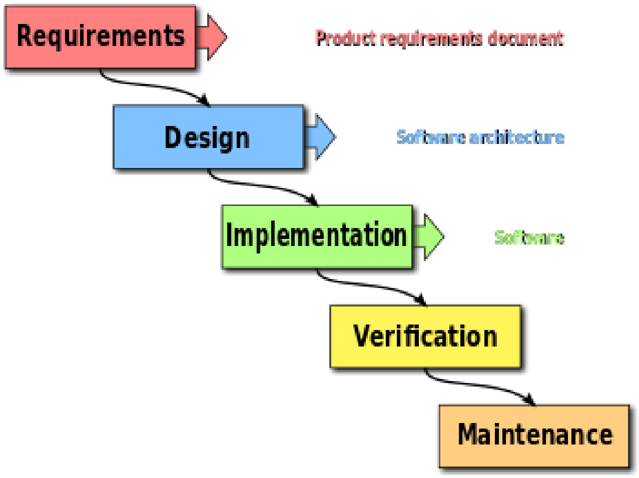

Figure 4 - Waterfall model (Wiki, n.d.)

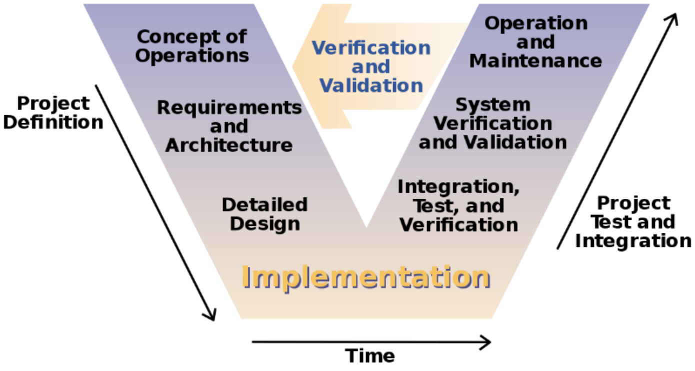
Concept of Operations
Verification and Validation
Operation and Maintenance
Project Definition
Requirements and Architecture
System Verification and Validation
Detailed Design
Integration, Test, and Verification
Project Test and Integration
Implementation
Time

Figure 5 - V model (Wikipedia-V-Model, n.d.)

1.2.2. When to use?

- V-Model:
  - The V-shaped model should be used for small to medium sized projects where requirements are clearly defined and fixed.
  - The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise.
  - For the projects where an accurate product testing is required.
- Waterfall Model:
  - This model is used only when the requirements are very well known, clear and fixed.
  - Product definition is stable.
  - Technology is understood.
  - There are no ambiguous requirements
  - Ample resources with required expertise are available freely
  - The project is short.

1.2.3. Advantages and Disadvantages of sequential model

- V-Model:

Advantages	Disadvantages
Simple and easy to use.	Very rigid and least flexible.
Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.	Software is developed during the implementation phase, so no early prototypes of the software are produced.
Proactive defect tracking – that is defects are found at early stage.	If any changes happen in midway, then the test documents along with requirement documents has to be updated.
Avoids the downward flow of the defects.	
Works well for small projects where requirements are easily understood.	

Table 4 - Advantages and Disadvantages of V-model (Tryqa, 2013)

- Waterfall Model

Advantages	Disadvantages
This model is simple and easy to understand and use.	Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.	No working software is produced until late during the life cycle.
Waterfall model works well for smaller projects where requirements are clearly defined and very well understood.	High amounts of risk and uncertainty.
In this model phases are processed and completed one at a time. Phases do not overlap.	Not a good model for complex and object-oriented projects.
	Poor model for long and ongoing projects.
	Not suitable for the projects where requirements are at a moderate to high risk of changing.

Table 5 - - Advantages and Disadvantages of Waterfall-model (Tryqa, 2013)

1.3. Advantages and Disadvantages for each following model in Tune Source:

1.3.1. Iterative model in Tune Source:

Advantages	Disadvantages
Tune Source can be developed as fast as possible: less documentation, simpler testing and debugging, less features to be developed per iteration.	More human resource is required.
Parallel development can be planned.	More management attention is required.
Easier to check and correct throughout a smaller iteration.	Highly skilled resource are required for analysis
Cost-effective.	

Table 6 - Advantages and Disadvantages of Iterative model in Tune Source

1.3.2. Sequential model in Tune Source:

Advantages	Disadvantages
Simple and straightforward to plan as well as applying.	No software works until late development, may not be able to meet timely requirement of Tune Source.
Easy to manage and arrange tasks.	High risk and uncertainty.
Phases and processes are completed one at a time.	Cannot adjust scope.
Phases are well documented.	Cannot accommodate changing requirements.

Table 7 - Advantages and Disadvantages of sequential model in Tune Source

2. Managing risk in the Spiral lifecycle model

2.1. What is risks?

The spiral model is inherently a risk-driven software development process model, in which the entire development process is guided by the associated risks. It aims to identify and evaluate software project risks and help reduce these risks and control project costs in a way that benefits a better controlled software project. Indeed, managing spiral-shaped risks distinguishes it from other process models using several types of risk management tasks; There is no apparent level of representation as in spiral. In spirals, this feature ensures that most risks are identified early and much earlier than other process models. Spiral development supports risk management in software projects in many ways summarized as follows. (al, 2012)

2.2. Risks management

- The initial risk analysis that acts aim at:
    - Identifying most risks threaten the project.
    - Classifying risks into user interface risks and development risks.
    - Evaluate these risks to determine the risks that need to be addressed through each cycle. Moreover, this classification helps developers implement risk-solving techniques such as prototyping and benchmarking.
- The evolution prototyping spirals that aim at resolving performance and user interface related risks. These spirals help in reducing major risks before proceeding into the development process.
- The risk analysis stage at each cycle that each phase of the waterfall phases in purpose of:
    - Resolving program development and interface control risks inherent from the start of the project.
    - Evaluating and resolving the new risks that might arise after changing any of the objectives, alternatives, or constraints at the beginning of the cycle.
- The iterative feature of the spiral which allows the development process to go back to the first quadrant at any point in progress which allows:
    - Objectives, alternatives and constraints to change as more attractive alternatives exist.
    - New technology to be incorporated easily during the development process.
    - The maximum optimization of project resources usage.

    - To deal with poorly done activities in the earlier phases.

- The review conducted at the end of each cycle with main stakeholders as a decision point to avoid the lack of commitment risks during the next cycle.

- Time and cost overrun risks are best managed using spiral development due to the risk analysis stage conducted at each cycle. In this stage, the cost and time required for each cycle are analyzed in advance to give a clear picture about the critical state of the project. This helps the project manager and the developers get more control over these risks.

- Risks related to the increased complexity of the project are also managed using spiral. This is achieved by the partitioning activity conducted at the planning phase.

- Decomposing the project into portions to be developed in parallel spirals obviously reduces time contention related risks, since more work could be achieved during the same interval.

2.3. How to register risks?

When someone want to manage risk in the spiral lifecycle model have follow each step:

- Name: Describe risk content and name it.
- Impacts: How risk effect.
- Priority: Priority of risk like: Urgent – High – Medium – Low.
- PIC: Assign risk to someone watch and follow
- Prevent plan: An action to prevent risks

Register risk	Describe	Priority	PIC	Status
Server down	Database is not working when too much people download music at the same time	Hight	Developer	Processing
…				

Table 8 - Register risk

3. A particular lifecycle model is selected for development environment in an example

- Base on pros, cons and nature of every potential SDLC models that may fit the requirements to develop the Tune Source, Spiral Model is the most suitable one. That model is used because:
    - The project is large: The project has 150 man-month and a lot of requirements, function or Technical that they need in the future.
    - Releases are update frequent: Web and Application are required to be frequent.
    - Creation of a prototype is applicable.
- Spiral model phase in Tune Source:
    - Planning: The Spiral model can estimate the cost, schedule and resources for the iteration that makes Tune Source can determine what they need to do next by a good way. It also involves understanding the system requirements for continuous communication between the system analyst and the customer.
    - Risk analysis: Identification of potential risk is done while risk mitigation strategy is planned and finalized.
    - Requirements are base lined at high level: Since the desired requirements of Tune Source has been already mentioned, which means that clients are already understood what they are looking for, the development process would be way faster.
    - Engineering: Tune Source can test, coding and deploying software at the customer site.
    - Evaluation: In the evaluation phase, Tune Source will review and evaluated project before the project moves on to next spiral. Any critical issues will be identified here and the necessary steps will be taken to deal with them.
    - At the end of each cycle of phases, the developers return to go through the same cycle once more. The Different phases can be review. (Base on: Kienitz, 2017)

4. Applying the Waterfall lifecycle model to a large software development project

- The waterfall model is connected to high costs and efforts. That is it requires approval of many documents, changes are costly to implement, iterations take a lot of effort and rework, problems are usually pushed to later phases. Few studies of Tune Source are explicitly focused on the waterfall model and some reasons for successful of the waterfall approach have been identified.
- The Waterfall model used at the Tune Source company runs through the phases: requirements engineering, design and implementation, testing release, maintenance. That explain the different phases and provide a selection of checklist-items to show what type of quality checks are made in order to decide whether the software artifact developed in a specific development phase can be passed on to the adjacent phase.

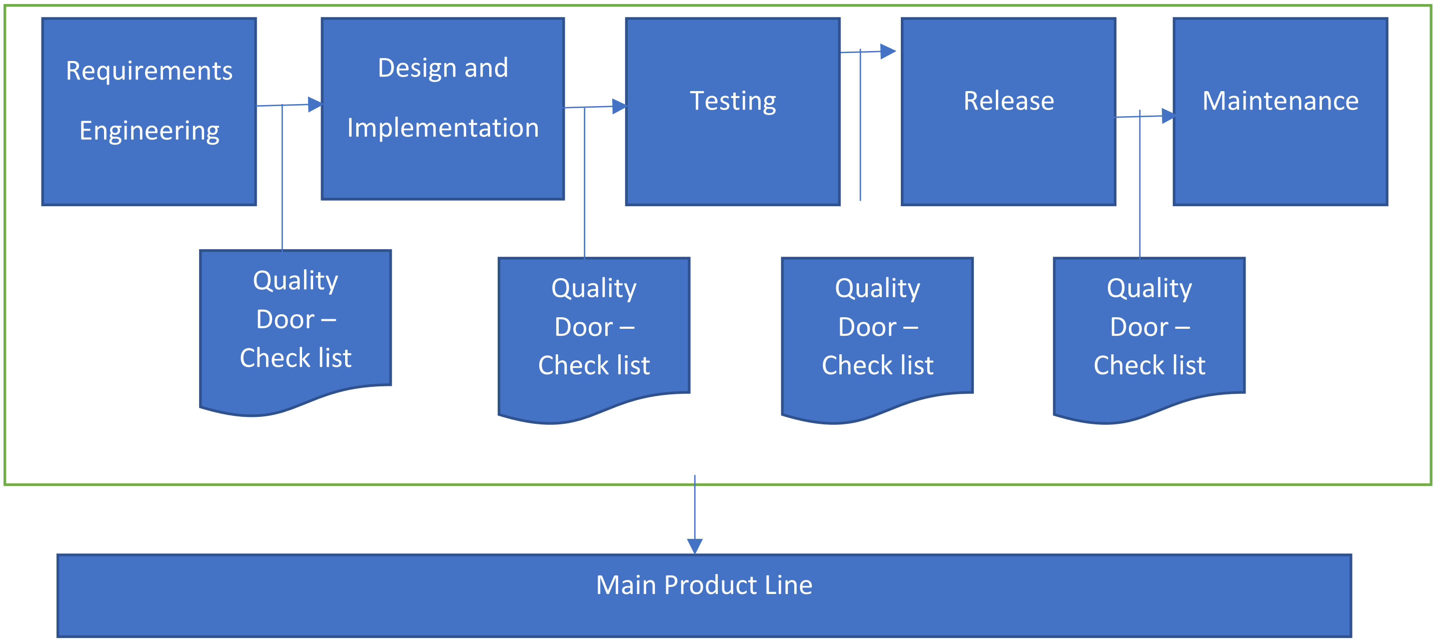
Requirements Engineering
Design and Implementation
Testing
Release
Maintenance
Quality Door – Check list
Quality Door – Check list
Quality Door – Check list
Quality Door – Check list
Main Product Line

Table 9 - Waterfall Development at Tune Source (Petersen, 2009)

- **Requirements Engineering:**

In this phase, customer needs are defined and recorded at a high level of abstraction. The requirements are then refined so that they can be used as input for the design and implementation phase. Requests are stored in a required repository. The number of requests selected depends on the resources available to the project. Since new products are not built from scratch, parts from the old product are also used as inputs for the requirements phase. Moreover, it is checked whether the stakeholders are identified and whether the solution supports business strategies.

- **Design and Implementation:**

In the design phase, the architecture of the system is created and documented. After that, the actual development of the system took place. Developers also conduct basic unit testing before handing over the developed code to the testing phase. The quality port checklist determines whether the architecture has been evaluated, whether there is a deviation from the requirements compared to the previous quality port decision, and there is a deviation from the timeline, plan or product scope as planned.

- **Testing:**

During this period, system integration is examined in terms of quality and functionality. To make a decision as to whether the system can be deployed, implementation measures are collected in the test version. Because the company provides complete solutions including graphics hardware and software and tests must be conducted on a variety of hardware and software such as software between customers. The results of the phase are reviewed according to the checklist to see if the system has been verified to see if there is a deviation from previous quality port decisions in terms of quality and time, even plan to deliver products to customers according to the company's instructions and whether the results of the products meet customer requirements.

- **Release:**

During the release phase, the product is put into a convertible state. That is, the release documents are categorized like the system's installation instructions for customers and instructions for use. Moreover, the build instructions for the system must be programmed. Building instructions can be used to turn on and turn off the main product line features to adjust the system to specific customizations. It is checked whether the results meet customer requirements, whether the customer accepts the results, and whether the final outcome is made timely and meets the requirements. its quality. A post-mortem analysis has to be performed as well.

- **Maintenance:**

After the product has been released to the customer, it will be maintained. That is, if customers discover problems in the product, they will take them to the company and receive support to solve them. If the problem is caused by an error in the product, the system update package will be sent to the customer.

- The most critical problems in waterfall development in Tune Source when large-scale industrial development:
    - The criticality of problems is not addressed in any way so far, making it hard to decide in which way it is most beneficial to improve the model, or whether the introduction of a new way of working will help in improving the key challenges experienced in the waterfall model.
    - Test coverage in waterfall development was reduced due to multiple reasons. Testing is done late in the project and thus if there have been delay in development, testing has to be compromised as it is one of the last steps in development and system has too much test at once after the over all system has been implemented.
    - From a design perspective, it is not always clear which version of the request should be made and by whom. The cause of this problem is that work often starts on unfinished or unapproved requirements which have not been properly baselined.
- Conclusion when applying the Waterfall lifecycle to large software development project:

This case study investigates issues related to waterfall models applied in the context of large-scale software development and compares business problems. The result is the most important issues in waterfall development regarding requirements and verification. Therefore, the waterfall model is not suitable for use in large-scale development. Therefore, Tune Source company needs to switch to another model. The comparison of case studies with business results shows that all problems found in the literature are found in the case study. Nevertheless, case studies provide corrective explanations of issues and identify four new issues such as:

- Co-fusion of who implements which version of the requirements
- High effort for maintenance
- Specialized competence focus and lack of confidence of people
- Localization problems due to communication barriers. (Base on: Petersen, 2009)

PART II: The importance of a feasibility study.

1. The purpose of a feasibility report.

1.1. What is feasibility study?

- A feasibility study is a preliminary study undertaken before the real work of a project starts to ascertain the likelihood of the project's success.
- The feasibility is defined as an evaluation or analysis of the potential impact of a proposed project or program.
- The feasibility is conducted to assist decision-makers in determining whether or not to implement a particular project or program. (Uobabylon, n.d.)
- The feasibility study will normally include:
    - A description of the tasks the system is required to do.
    - A preliminary design and initial cost for the system
    - Alternative designs so that most appropriate can be chosen
    - A cost/benefit analysis
- Conclusion as to whether or not the design is worth implementing and which of the alternatives should be chosen. (Vijay, 2013)
- So depending upon the aspect on which feasibility is being done it can be categorized into 4 classes: (Slideplayer, 2016)
    - Technical Feasibility:
        - A measure of the practicality of a specific technical solution.
        - A measure of the availability of technical resources and expertise.
    - Operational Feasibility:
        - A measure of how well the solution of problems or a specific solution will work in the organization.
        - A measure of how people feel about the system/project
    - Economic Feasibility:
        - A measure of the cost-effectiveness of a project or solution. This is often called a cost-benefit analysis.
    - Schedule Feasibility:
        - A measure of how reasonable the project timetable is.

.

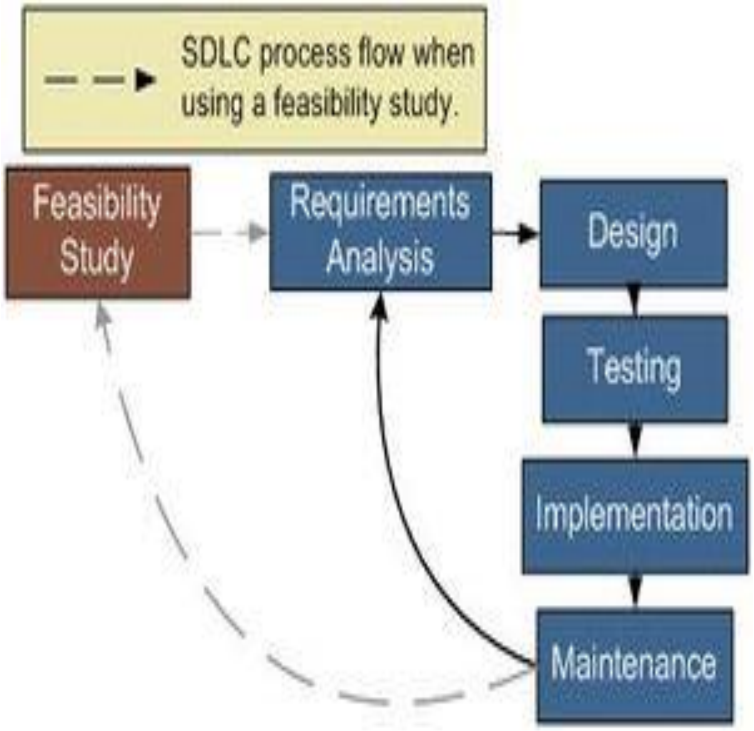

Figure 6 - SDLC process flow when using a feasibility study. (Kodiyath, 2013)

1.2. The purpose of feasibility report when use in Tune Source

- Technical Feasibility (Can We Build It?):
    - Familiarity with application: When develop application in Tune Source that familiar to people and easily to use.
    - Familiarity with technology: In the future, Tune Source will not sell the CD in store so use technology will familiarity generates less risk.
    - Project Size: Large project have more risk.
    - Compatibility: The harder it is to integrate the system with the company's existing technology, the higher risk
- Economic Feasibility (Should We Build it?)
    - Return on Investment: Over 3 years.
    - Break-even Point: 2-3 years
    - Benefit: Over 3 years.

Development Costs	Operational Costs
Hardware and software	Software upgrade
Development training	Hardware repairs
Data conversion costs	User training
Office space and equipment	Communications charges

Table 10 - Example Economic Feasibility

- Organizational Feasibility (If WE Build It, Will They Come?):
  - Project champions:
  - Senior management:
  - Users:
  - Other stakeholders:

	Role	To Enhance Organizational Feasibility
Champion	- Promotes the project - Provides resource	- Make a presentation about the object of the project. - Create a prototype of the system to demonstrate its potential value
Organizational Management	- Know about the project - Budget enough money for the project	- Make a presentation to management about the objective of the project - Encourage the champion.
Users	- Make decisions that influence the project - Perform hands-on activities for the project	- Assign users official roles on the project team. - Assign for feedback from user regularly.

Table 11 - Example of Organizational Feasibility (Wixom)

2. How technical solutions can be compared.

2.1. Three programming languages and Alternative Matrix.

- C Sharp (C#):

C Sharp (C#) is a general-purpose, multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed around 2000 by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2018). C# is one of the programming languages designed for the common language infrastructure. (Wikipedia, n.d.)

- JavaScript:

JavaScript often abbreviated as JS, is a high-level, interpreted programming language that conforms to the ECMAScript specification. It is a language that is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it. This language supports event-driven, functional, and imperative programming style (wikipedia, n.d.).

- Python:

Python is an interpreted high level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. That available for many operating system. (Wikipedia, n.d.)

- Alternative Matrix

An alternative matrix can be used to organize the pros and cons of the design alternatives so that the best solution will be chosen in the end. This matrix is created by the same steps as the feasibility analysis. The only difference is that the alternative matrix combines several feasibility analyses into one matrix so that the alternatives can be easily compared. The alternative matrix is a grid that contains the technical, economical, and organizational feasibilities for each system candidate, pros and cons associated with adopting each solution, and other information that is helpful when making comparisons. Sometimes, weights are provided for different parts of the matrix to show when some criteria are more important to the final decision. (Sons, 2012)

2.2. Alternative technical solutions matrix

The alternative technical solutions assessment is carried in the following these steps:

- A system of criteria is built and relative importance score imposed.
- A set of candidate solutions is chosen and given scores.
- For each solution and each criteria compute weighted score by formula:

  Weighted score = relative importance * score
- For each solution compute total score by SUM all weighted score.
- Candidates are compared by scores. Greater is better.

Evaluation Criteria	Priority point	Alternative 1: Plugin using C#	Score (1-5)*	Weighted Score	Alternative 2: plugin using JavaScript	Score (1-5)*	Weighted Score	Alternative 3: Open Source using Python	Score (1-5)*	Weighted Score
Technical Issues:										
C1: Support front-end and back-end (Full stack)	**10**	Current system uses Windows server	**2**	**20**	JS supports full stack	**5**	**50**	Not fully support Front-end	**4**	**40**
C2: Able to run on Windows Server	**10**	Current system uses Windows server	**3**	**30**	Current system fully supports JS	**5**	**50**	Fully supported	**5**	**50**
C3: Support Multimedia libraries	**10**	Microsoft supports what we want	**3**	**30**	Easy to find open source library	**4**	**40**	Multiple open source libraries	**5**	**50**
Economic Issues										
C4: License costs	**10**	Visual studio + NetFramework cost $4500	**2**	**20**	Free for commercial use	**5**	**50**	Free for commercial use	**5**	**50**
C5: Server costs	**10**	Reuse current server	**4**	**40**	Reuse current server	**4**	**40**	Reuse current server	**4**	**40**
Organizational Issues:										
C6: Able to recruit a senior developer	**10**	Very high salary	**3**	**30**	Not easy to find	**1**	**10**	Easy to recruit	**4**	**40**
C7: Able to recruit a experienced project manager	**10**	Easy to find	**5**	**50**	Easy to find	**5**	**50**	Easy to find	**5**	**50**
C8: Able to hire a tester	**25**	Easy to find	**5**	**125**	Easy to find	**4**	**100**	Easy to find	**5**	**125**
Risks:										
Maintenance / Upgrade	**-20**	Need experience	**4**	**-80**	Need experience	**3**	**-60**	Need experience	**4**	**-80**
Synchronization	**-30**	Hard to sync with system	**4**	**-120**	Yes, but not easy	**3**	**-90**	Yes, but not easy	**3**	**-90**
TOTAL				**145**			**240**			**275**

Table 12 - Alternative solutions Matrix

2.3. Final solution alternative:

After completing the Alternative matrix, **Python** presents as the most possible and suitable solution for the development of Tune Source integration:

- Various useful framework Python provides a huge source of framework that will increase the designing and developing process's efficiency and reduce time consumption since Tune Source's required functions could have been included in those frameworks. Frameworks such as: Django, Flask, Pyramid, etc.
- Usability: Python is considered to be one of the most commonly used and also one of the top programming languages in demand today and future. As per GitHub, it is one of the most used programming languages thanks to its user-friendly data structure and simple, concise language syntax.
- When need to upgrade the system, It easily to find people with professional competence and synchronize with old applications.
- Friendly Database programming Python supports various databases like MySQL, Oracle, Sybase, PostgreSQL, etc. Python also supports Data Definition Language (DDL), Data Manipulation Language (DML) and Data Query Statements. For database programming, the Python DB API is a widely used module that provides a database application programming interface.

3. The components of a feasibility report.

A project feasibility report, the documentation of the feasibility study is a complete report to aim to objectively and rationally uncover the strength, weaknesses, constraints of an existing project, business or proposed venture. In relation to Tune Source, base on alternative matrix of Tune Source, there are some suggest components of a feasibility:

- Technical feasibility: This review focuses on the technical resources available for project. It helps both the developer and the client side determine whether technical resources meet the capacity and the development team is not able to convert customer ideas into working systems. Technical feasibility also involves assessing hardware, software and other technological requirements of the proposed system.
  - Support front-end and back-end: This criteria allow Tune Source know what is importance when develop using between C#, JavaScript and Python, the most suitable use for design front-end and back-end
  - Able to run on Windows Server: What programming languages Tune Source will use to get fully supported on Windows Server that might help system avoid system-error.
  - Support Multimedia libraries: When Tune Source want to develop Website or Application, this will let company know what supports what company want to multiple open source libraries.

- Economic Issues: This assessment is often related to the cost / benefit analysis of The project, which helps both parties determine the viability, costs and benefits associated with the project before the financial resources are allocated. This assessment also serves as an independent project evaluation and enhancement of the project's credibility, helping decision makers identify positive economic benefits for the organization that the proposed project will provide.
    - License costs:
    - Server costs
- Organizational Issues: A definition of business and corporate structure legislation. This may include information about the founder, their professional background and the skills they have needed to take the company off the ground and keep it active.
    - Able to recruit a senior developer: The company will easy when find a senior developer to build web, application.
    - Able to recruit a experienced project manager: Project manager is very important when the company wants to implement any plan related to software or project development
    - Able to hire a tester: Easily check whether the performance of the web or application of Tune Source meets all of the features or errors and especially the security of the company.
- Risks:
    - Maintenance / Upgrade: When the system wants to upgrade as well as warranty, which language is easy for IT staff and easy for new employees.
    - Synchronization: When Tune Source wants to synchronize web or application to make it easy for users to use and manage

4. The Impact of different feasibility criteria on a software investigation.

As a music business, Tune Source has grown steadily over the years with 3-5% a year, but to integrate with industry 4.0 and compete with other competitors, Tune Source needs short-term and long-term plans, as well as more efficient use of capital. Maximize the market in the world. Besides, there should be additional and supplementary products to diversify products to increase revenue. Additional knowledge for employees and managers.

Depend on Tune Source company, There are some feasibility criteria for Tune Source company:

- Technical Feasibility:
  - Using Internet Service Provider (ISP):
    - As in the current scenario, Tune Source is using local ISP for the hosting solution. In the future, this hosting solution can not works well for international market. However, as if the site got developed with more features, as well as the company are willingly to expand the market area into international market, this hosting solution may no longer suitable.
    - The solution is looking for and co-operate with international IPS to make larger amount of server across the world that Tune Source can extend and be able to exploit to big market.
    - When Tune Source has enough ISP and server for system, this also means develop related technologies such as Web, Application and another technologies.
    - This solution will cost a plenty amount of money
  - Using Web:
    - Tune Source is having a website to sell music that will help the company keep abreast of technologies as the world is moving to industry 4.0.
    - Website allow customers to search and purchase CDs. Since the company has decide to develop a new system that help customers can search for digital music, listen to music online and download music, specially is gift cards and customer subscription account.
    - There are 2 feasibility criteria:
      - Create new website using Web app – PWA: that allow user browsing web in smartphone like application (without download any application).
      - Upgrade current website: Upgrade bandwidth, server, firewall, security, data storage.
    - The most possible option at present is upgrade current website because the basic infrastructure of an e-commerce website has been developed and functioned, so the development of the system

    will take less time to process compared to build an entire new website that will save money for company. But in the future, when Tune Source want to grow up more, they must create new Web App.

  - Converting traditional CDs music to digital music:
    - Nowadays, Sell CD music is no longer popular as 10-15 years ago. Because of the development of digital music and dominate with efficiency, versatility. The main reason for company to looking a new system development is digital music market.
    - To catch-up with trends and consumer tastes, Tune Source need to convert great amount of rare audio recordings which save in CDs into digital versions.
    - This is very feasible, but the problem is license so Tune Source need to pay more attention to this issue to ensure maintaining revenue and maintaining the market.
- Economic Feasibility:
  - Keep store sell CDs:
    - CDs nowadays may no longer keeping its own position as it used to be. The development shops to sell CSs will cause the company to reduce profits sooner or worse situation can go bankrupt.
    - However, as long as other company keeps giving up con CDs to pursue the Digital music market, the rarity of Tune Source's CDs will keep accelerating and getting higher price.
    - The most possible feasibility plan is keep up with the store but not too much focus on them. Just only keep a few large stores to develop the brand and reputation of the company. If this plan is still maintained then in the future company may monopolize the market about selling traditional CDs.
  - Cross-selling:
    - Customer who have downloaded a digital track or two of a CD will decide to purchase the entire CD in the store is the way for Tune Source get more benefit and develop both digital music and CDs.
    - Customer may listen to digital music samples from original CDs then decide if they want to buy it or not. When they like the song, they can be able to download only that song, instead of buying CDs with other songs that they might not interested. In another situation, if they are interested they can find out that entire CD suit their interested so they will like to purchase them.
    - This plan is feasibility for company to increase the profit.
  - Selling digital music and related products
    - The most profit is come from sales from customer subscription and music downloads. The sales of these products by Tune Source is

very effective and it will bring more and more profits in the future if Tune Source properly developed with more suitable technology products.

- Tune Source with customer subscriptions that not only earn a lot of profits but also have a lot of loyal customer who buy products regularly at present or in the future.

- Organizational Feasibilities:
    - Promptly respond to requests from customers:
        - Tune Source understands what the requirements of customers and markets need, thereby making appropriate complaints.
        - Customer recommendations will be noted by Tune Source, then improved systems, sales forms, and more focused on customer benefits.
    - Operate project activities and development as required from the business plan:
        - When Tune Source launched new business ideas as well as new products, business plans were launched and implemented
        - After each stage, Tune Source continues to dissuade the orientation issues for the future of the company
    - Founding and operating from experienced people:
        - Tune Source is brainchild of three entrepreneurs with ties the music industry with a number of brick and mortar stores in southern California, specially in hard-to-find, classic jazz, rock, country and folk recordings.
        - The IT department at Tune Source has become experienced with Internet technology as it has worked with the ISP to maintain the site

CONCLUSION

Tune Source not in an exceedingly very secure monetary position. enhancements in each space of the corporate square measure required if the corporate is, within the initial instance, to survive and so grow. The key square measures of reform are the liquidity of the corporate and also the amount and quality of capital, profitableness, and monetary stability. Management should address these areas at the same time if the corporation is to beat its gift poor record. It should be remembered that this analysis is limited: a larger depth of understanding and analysis will solely occur with usage of alternative resources like comparisons with budget forecasts and also the statement of changes in monetary position. solely once this method will a full appreciation of the company's current state of affairs and attainable future occur. At now the corporate doesn't have sturdy future prospects within the areas of profitableness, liquidity or stability if it continues on its current path. Investors ought to be troubled with current rates of come and management and creditors ought to be troubled with the liquidity of the corporate as indicated within the magnitude relation analysis.

References

al, H. e. (2012).

Existek. (2017, 8 04). Retrieved from Existek: https://existek.com/blog/sdlc-models/

Kienitz, P. (2017, 06 03). *dcslsoftware*. Retrieved from dcslsoftware: https://www.dcslsoftware.com/pros-cons-spiral-software-development/

Kodiyath. (2013). *solidswiki*. Retrieved from solidswiki: http://www.solidswiki.com/index.php?title=Feasibility_Studies&mobileaction=toggle_view_mobile

Petersen, K. (2009, 06). *researchgate*. Retrieved from researchgate: https://www.researchgate.net/publication/30498645_The_Waterfall_Model_in_Large-Scale_Development

Powell-Morse, A. (2016, 12 15). *Airbrake*. Retrieved from https://airbrake.io/blog/sdlc/iterative-model

Professionalqa. (2018, 12 31). Retrieved from Professionalqa: http://www.professionalqa.com/iterative-model

Slideplayer. (2016). Retrieved from Slideplayer: https://slideplayer.com/slide/7604028/

Sons, J. W. (2012). System analysis and design. In J. W. Sons, *System analysis and design.*

Tryqa. (2013). Retrieved from Tryqa: http://tryqa.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/

Uobabylon. (n.d.). Retrieved from Uobabylon: http://www.uobabylon.edu.iq/uobcoleges/ad_downloads/4_13233_344.pdf

Vijay, N. (2013, 11 25). *slideshare*. Retrieved from slideshare: https://www.slideshare.net/ilovenaveen/system-development-life-cyclenaveen-vijay

Wiki. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Waterfall_model

wikipedia. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/JavaScript

Wikipedia. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/C_Sharp_(programming_language)

Wikipedia. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Python_(programming_language)

Wikipedia-V-Model. (n.d.). Retrieved from https://en.wikipedia.org/wiki/V-Model

Wixom, R. (n.d.). System Analysis and Design 5th Edition Dennis.