

# Assignment Brief 01 (RQF)

## Higher National Certificate/Diploma in Business

<b>Student Name/ID Number:</b>	<b>Tran Dang Khoa / GCS190407</b>
<b>Unit Number and Title:</b>	<b>Unit 09: Software Development Life Cycle</b>
<b>Academic Year:</b>	
<b>Unit Assessor:</b>	
<b>Assignment Title:</b>	<b>Plan a software development life cycle</b>
<b>Issue Date:</b>	<b>12/12/2020</b>
<b>Submission Date:</b>	<b>12/06/2021</b>
<b>Internal Verifier Name:</b>	<b>Vo Ngoc Mai</b>
<b>Date:</b>	

<b>Submission Format:</b>
<p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• The submission is in the form of 1 document.</li> <li>• You must use the Times font with 12pt size, turn on page numbering; set line spacing to 1.3 and margins to be as follows: left = 1.25cm, right = 1cm, top = 1cm, bottom = 1cm. Citation and references must follow the Harvard referencing style.</li> </ul> <p><i>Submission:</i></p> <ul style="list-style-type: none"> <li>• Students are compulsory to submit the assignment in due date and in a way requested by the Tutor.</li> <li>• The form of submission will be a soft copy posted on <a href="http://cms.greenwich.edu.vn/">http://cms.greenwich.edu.vn/</a>.</li> <li>• Remember to convert the word file into <b>PDF</b> file before the submission on CMS.</li> </ul> <p><i>Note:</i></p> <ul style="list-style-type: none"> <li>• The individual Assignment must be your own work, and not copied by or from another student.</li> <li>• If you use ideas, quotes or data (such as diagrams) from books, journals or other sources, you must reference your sources, using the Harvard style.</li> <li>• Make sure that you understand and follow the guidelines to avoid plagiarism. Failure to comply this requirement will result in a failed assignment.</li> </ul>

## Unit Learning Outcomes:

**LO1** Describe different software development lifecycles.

**LO2** Explain the importance of a feasibility study.

## Assignment Brief and Guidance:

### Assignment scenario

Tune Source is a company headquartered in southern California. Tune Source is the brainchild of three entrepreneurs with ties to the music industry: John Margolis, Megan Taylor, and Phil Cooper. Originally, John and Phil partnered to open a number of brick-and-mortar stores in southern California specialising in hard-to-find and classic jazz, rock, country, and folk recordings. Megan soon was invited to join the partnership because of her contacts and knowledge of classical music. Tune Source quickly became known as the place to go to find rare audio recordings. Annual sales last year were \$40 million with annual growth at about 3%–5% per year. Tune Source currently has a website that enables customers to search for and purchase CDs. This site was initially developed by an Internet consulting firm and is hosted by a prominent local Internet Service Provider (ISP) in Los Angeles. The IT department at Tune Source has become experienced with Internet technology as it has worked with the ISP to maintain the site.

### System Request

**Project Sponsor:** Carly Edwards, Assistant Vice President, Marketing.

**Business Need:** 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 our website.

**Business Requirements:** Using the Web or in-store kiosks, customers will be able to search for and purchase digital music downloads. The specific functionality that the system should have includes the following:

- Search for music in our digital music archive.
- Listen to music samples.
- Purchase individual downloads at a fixed fee per download.
- Establish a customer subscription account permitting unlimited downloads for a monthly fee.
- Purchase music download gift cards.

**Business Value:** We expect that Tune Source will increase sales by enabling existing customers to purchase specific digital music tracks and by reaching new customers who are interested in our unique archive of rare and hard-to-find music. We expect to gain a new revenue stream from customer subscriptions to our download services. We expect some increase in cross-selling, as customers who have downloaded a track or two of a CD decide to purchase the entire CD in a store or through our website. We also expect a new revenue stream from the sale of music download gift cards.

Conservative estimates of tangible value to the company include the following:

- \$757,500 in sales from individual music downloads.
- \$950,000 in sales from customer subscriptions.
- \$205,000 in additional in-store or website CD sales.
- \$153,000 in sales from music download gift cards.

Special Issues or Constraints:

- The marketing department views this as a strategic system. The ability to offer digital music downloads is critical in order to remain competitive in our market niche. Our music archive of rare and hard-to-find music is an asset that is currently underutilized.
- Many of our current loyal customers have been requesting this capability, and we need to provide this service or face the loss of these customers' business.
- Because customers have a number of music download options available to them elsewhere, we need to bring this system to the market as soon as possible.

### Tasks

Complete the following tasks:

#### **Task 1 – SDLC model**

You are a project manager of a company named ABC. Your company has been hired by Tune Source to carry out a project that helps them develop a software for the requirements specified in the system request. As the first step, you need to:

1. Describe the following SDLC models: waterfall, v-model, prototyping, scrum and spiral. Choose one that you think suitable for the project and explain why.

- 350 - 500 words for each model.
- Explanation: 400 – 600 words.

Discuss the suitability of each of the SDLC models for the project. For each model, specify whether it is most, moderately or least suitable.

- Word limit: 800 - 1000 words.

Discuss the merits of applying the waterfall model to a large software development project.

- Word limit: 800 – 1200 words.

2. Identify some risks and discuss an approach to manage them.

You will have the present what is **Risk Management process** with clear illustrations and explanations.

Then you will create a **Risk Management Matrix** to assess and manage risks of Tune Source project.

- Word limit: 600 – 1000 words.

## **Task 2 – Feasibility study**

1. Discuss the purpose of conducting a feasibility study for the project.

- Word limit: 400 – 600 words.

2. Discuss how the three feasibility criteria (technical, economic, organizational) are applied to the project. Discuss whether the project is feasible.

Discuss alternative technical solutions using the alternative matrix.

- Word limit: 1200 – 1500 words.

3. Explain the components of a feasibility report.

Discussion economic feasibility study on Tune Source project (NPV, Cashflow, Break-Even Point)

- Word limit 350 – 500 words.

Discussion organizational feasibility study on Tune Source project

- Word limit 350 – 500 words.

4. Assess the impact of each feasibility criterion on a software investigation.

Discussion and represent as feasibility alternatives matrix for Tune Source project

- Word limit: 500 – 700 words.

Learning Outcomes and Assessment Criteria (Assignment 01):			
Learning Outcome	Pass	Merit	Distinction
<b>LO1</b> Describe different software development lifecycles	<p><b>P1</b> Describe two iterative and two sequential software lifecycle models.</p> <p><b>P2</b> Explain how risk is managed in the Spiral lifecycle model.</p>	<p><b>M1</b> Describe, with an example, why a particular lifecycle model is selected for a development environment.</p>	<p><b>D1</b> Assess the merits of applying the Waterfall lifecycle model to a large software development project.</p>
<b>LO2</b> Explain the importance of a feasibility study	<p><b>P3</b> Explain the purpose of a feasibility report.</p> <p><b>P4</b> Describe how technical solutions can be compared.</p>	<p><b>M2</b> Discuss the components of a feasibility report.</p>	<p><b>D2</b> Assess the impact of different feasibility criteria on a software investigation.</p>

## Contents

<b>Assignment Brief 01 (RQF)</b>	<b>1</b>
<b>Higher National Certificate/Diploma in Business</b>	<b>1</b>
<b>P1 Describe two iterative and two sequential software lifecycle models.</b>	<b>8</b>
<b>1. SDLC - Waterfall.</b>	<b>8</b>
1.1. Waterfall Model - Design.	8
1.2. Waterfall Model – Application.	9
1.3. Waterfall Model – Advantages.	10
1.4. Waterfall Model – Disadvantages.	10
<b>2. SDLC - Prototyping Model.</b>	<b>10</b>
2.1. What is prototyping model.	10
2.2. Steps of the prototyping model.	11
2.3. Types of prototype models.	12
2.4. Advantages of the prototyping model.	12
2.5. Disadvantages of the prototyping model.	13
<b>3. SDLC - Agile Model.</b>	<b>13</b>
3.1. What is Agile?	13
3.2. The advantages of the Agile Model are as follows.	14
3.3. The disadvantages of the Agile Model are as follows.	15
<b>4. SDLC - Spiral Model.</b>	<b>15</b>
4.1. Spiral Model – Design.	15
4.2. The advantages of the Spiral SDLC Model are as follows.	17
4.3. The disadvantages of the Spiral SDLC Model are as follows.	17
<b>5. Model for suitable for the project.</b>	<b>18</b>
<b>P2. EXPLAIN HOW RISK IS MANAGED IN THE SPIRAL LIFECYCLE MODEL.</b>	<b>18</b>
1. Introduction	18
2. What is risk?	18
3. What is risk management?	18
Step 1: Identify the Risk.	19
Step 2: Analyze the Risk.	19
Step 3: Evaluate the Risk or Risk Assessment.	19
Step 4: Treat the Risk.	19
Step 5: Monitor and Review the Risk	20
<b>Common risks in development project.</b>	<b>20</b>
1. Technology risk.	20
2. Communication risk.	20

3. Scope creep risk.....	20
4. Cost risk. ....	20
5. Operational risk.....	20
<b>P3. EXPLAIN THE PURPOSE OF A FEASIBILITY REPORT. ....</b>	<b>22</b>
1. Definition Of Feasibility Report/Study. ....	22
2. Key components of a feasibility report.....	22
3. Purpose Of A Feasibility Report.....	22
<b>P4 DESCRIBE HOW TECHNICAL SOLUTIONS CAN BE COMPARED. ....</b>	<b>23</b>
1. Introduction.....	23
2. Feasibility study of Tune Source project.....	23
2.1. Technical feasibility.....	23
2.2. Economic feasibility. ....	23
2.3. Organisational feasibility .....	24
3. Provide an alternative matrix for Tune Source project.....	24
3.1. Introduction. ....	24
3.2. ASP.NET technology.....	24
3.2.1. Definition of ASP.NET technology. ....	24
3.2.2. Advantages of ASP.NET .....	25
3.2.3. Disadvantages of ASP.NET.....	25
3.3. JavaScript technology .....	25
3.3.1. Introduction .....	25
3.3.2. Advantages of JavaScript technology .....	25
3.3.3. Disadvantages of Java Script technology. ....	26
3.4. PHP technology. ....	26
3.4.1. Introduction.....	26
3.4.2. Advantages of PHP technology. ....	26
3.4.3. Disadvantages of PHP technology. ....	26
4. Weighted Alternative Matrix.....	26
5. Solution for TS and Explain. ....	27
5.1 Introduction.....	27
5.2 Tune Source Project.....	27
Reference.....	28

## **P1 Describe two iterative and two sequential software lifecycle models.**

### **1. SDLC - Waterfall.**

#### **1.1. Waterfall Model - Design.**

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.





The sequential phases in Waterfall model are –

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

### 1.2. Waterfall Model – Application.

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are –

- 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.

### **1.3. Waterfall Model – Advantages.**

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the major advantages of the Waterfall Model are as follows –

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

### **1.4. Waterfall Model – Disadvantages.**

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The major disadvantages of the Waterfall Model are as follows –

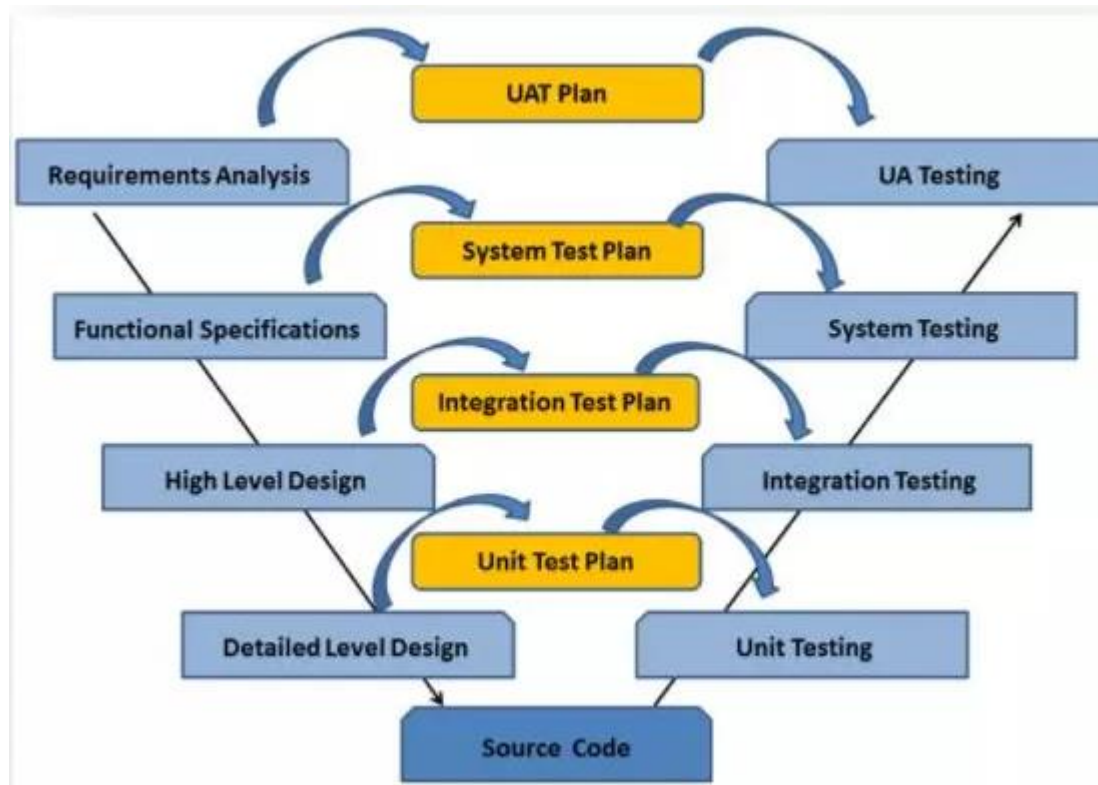
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang" at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

## **2. SDLC - Prototyping Model.**

### **2.1. What is prototyping model.**

The prototyping model is a systems development method in which a prototype is built, tested and then reworked as necessary until an acceptable outcome is achieved from which the complete system or product can be developed. This model works best in scenarios where not all of the project requirements are known in detail ahead of time. It is

an iterative, trial-and-error process that takes place between the developers and the users.



## 2.2. Steps of the prototyping model.

**In most cases, the steps of the prototyping model are as follows.**

- The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the departments or aspects of the existing system.
- A preliminary, simple design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- The users thoroughly evaluate the first prototype and note its strengths and weaknesses, what needs to be added and what should to be removed. The developer collects and analyzes the remarks from the users.
- The first prototype is modified, based on the comments supplied by the users, and a second prototype of the new system is constructed.
- The second prototype is evaluated in the same manner as was the first prototype.

- The preceding steps are iterated as many times as necessary, until the users are satisfied that the prototype represents the final product desired.
- The final system is constructed, based on the final prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.

### **2.3. Types of prototype models.**

**There are a few types of prototype models that can be implemented by development teams based on their needs.**

- Rapid throwaway- This method involves exploring ideas by quickly developing a prototype based on preliminary requirements that is then revised through customer feedback. The name rapid throwaway refers to the fact that each prototype is completely discarded and may not be a part of the final product.
- Evolutionary- This approach uses a continuous, working prototype that is refined after each iteration of customer feedback. Because each prototype is not started from scratch, this method saves time and effort.
- Incremental- This technique breaks the concept for the final product into smaller pieces, and prototypes are created for each one. In the end, these prototypes are merged into the final product.
- Extreme- This prototype model is used specifically for web development. All web prototypes are built in an HTML format with a services layer and are then integrated into the final product.

### **2.4. Advantages of the prototyping model.**

**Using a prototype model can bring multiple advantages, including.**

1. This model is flexible in design.
2. It is easy to detect errors.
3. We can find missing functionality easily.
4. There is scope of refinement, it means new requirements can be easily accommodated.
5. It can be reused by the developer for more complicated projects in the future.
6. It ensures a greater level of customer satisfaction and comfort.
7. It is ideal for online system.
8. It helps developers and users both understand the system better.

9. Integration requirements are very well understood and deployment channels are decided at a very early stage.

10. It can actively involve users in the development phase.

## **2.5. Disadvantages of the prototyping model.**

- The main disadvantage of this methodology is that it is more costly in terms of time and money when compared to alternative development methods, such as the spiral or waterfall model. Since in most cases the prototype is discarded, some companies may not see the value in taking this approach.
- Additionally, inviting customer feedback so early on in the development lifecycle may cause problems. One problem is that there may be an excessive amount of change requests that may be hard to accommodate. Another issue could arise if after seeing the prototype, the customer demands a quicker final release or becomes uninterested in the product.

## **3. SDLC - Agile Model.**

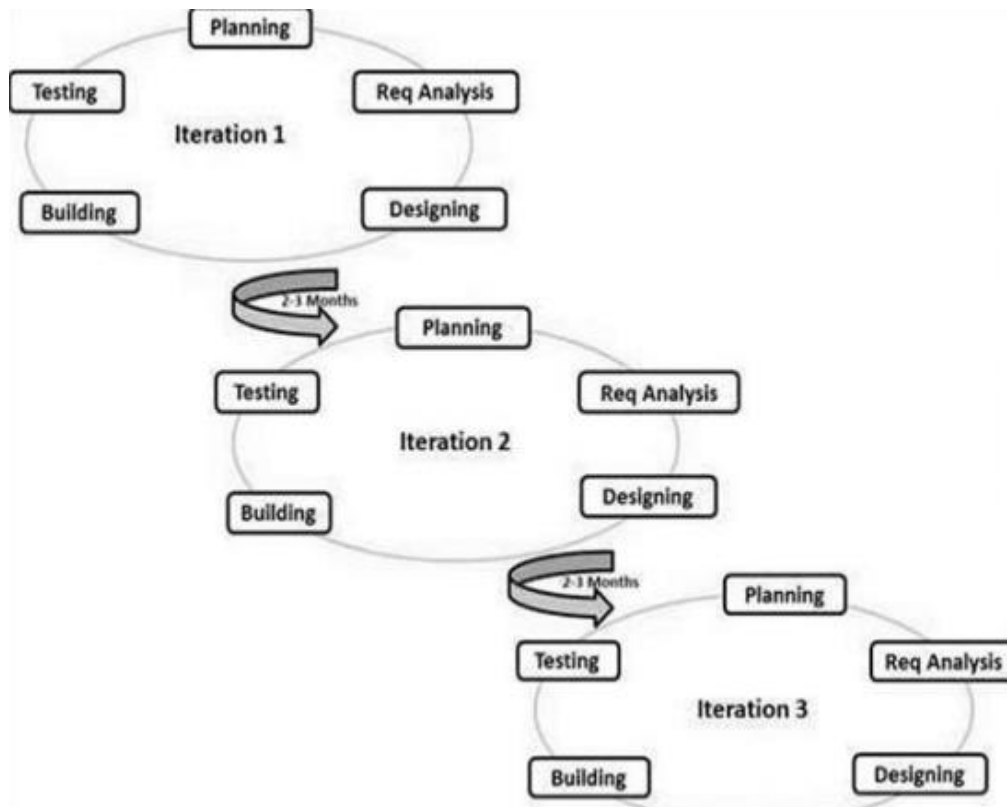
Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration 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.

### **3.1. What is Agile?**

- Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.
- Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.
- Here is a graphical illustration of the Agile Model.



- The Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.
- The most popular Agile methods include Rational Unified Process (1994), Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995). These are now collectively referred to as Agile Methodologies, after the Agile Manifesto was published in 2001.

### **Following are the Agile Manifesto principles.**

- Individuals and interactions – In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- Working software – Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
- Customer collaboration – As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
- Responding to change – Agile Development is focused on quick responses to change and continuous development.

### **3.2. The advantages of the Agile Model are as follows.**

- 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.

### **3.3. The disadvantages of the Agile Model are as follows.**

- 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.
- There is a 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 documentation.

## **4. SDLC - Spiral Model.**

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

### **4.1. Spiral Model – Design.**

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



### **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.

### **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.

### **Construct or Build.**

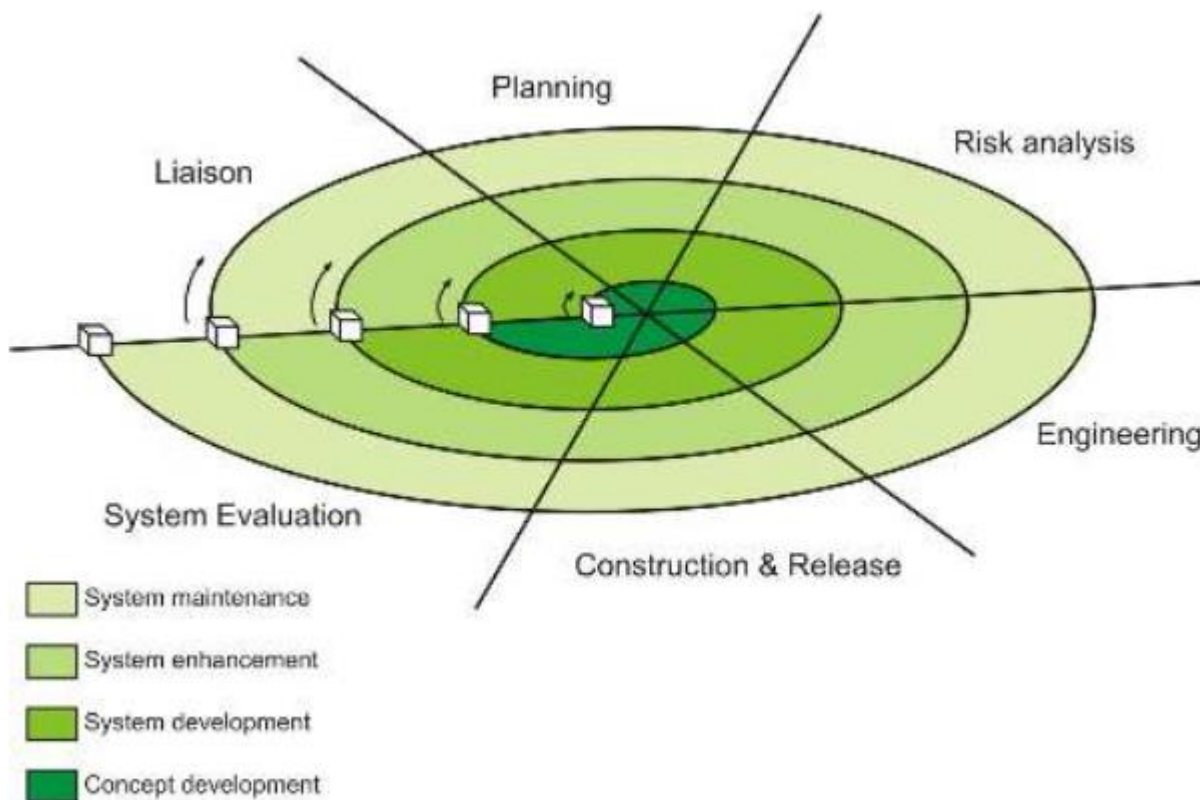
- The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.
- Then in the subsequent spirals with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

### **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.

**The following illustration is a representation of the Spiral Model, listing the activities in each phase.**





Based on the customer evaluation, the software development process enters the next iteration and subsequently follows the linear approach to implement the feedback suggested by the customer. The process of iterations along the spiral continues throughout the life of the software.

#### 4.2. The advantages of the Spiral SDLC Model are as follows.

- Changing requirements can be accommodated.
- Allows extensive use of prototypes.
- Requirements can be captured more accurately.
- Users see the system early.
- Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

#### 4.3. The disadvantages of the Spiral SDLC Model are as follows.

- Management is more complex.
- End of the project may not be known early.
- Not suitable for small or low risk projects and could be expensive for small projects.

- Process is complex
- Spiral may go on indefinitely.
- Large number of intermediate stages requires excessive documentation.

## **5. Model for suitable for the project.**

- I will choose the waterfall model for the project "Tune Source".
- Below are the benefits of choosing a waterfall model for this project.
- Waterfall is great if we can come up with a coherent plan, process, and projections that arise in the first place.
- This method is simple, easy to apply and easy to understand.
- Clear implementation process, easy and quick project distribution, easy cost allocation.
- Suitable for small projects and does not generate many new requirements in the implementation process.
- Easily manage and track where the project is going as each phase has a specific process, a clear list of tasks and predictable results.

## **P2. EXPLAIN HOW RISK IS MANAGED IN THE SPIRAL LIFECYCLE MODEL.**

### **1. Introduction**

- In every project, the risk is inevitable. This Tune Source project has some risks that we need to solve. For this reason, we need to understand how to manage the risk and what are the risk in this project.

### **2. What is risk?**

- Software development process is a risky process; SDLC is vulnerable. to risks from the start of the project till the final acceptance of the software. product. Each phase of the SDLC is susceptible to different sets of threats. that might hinder the development process from being completed.

### **3. What is risk management?**

- Risk management encompasses the identification, analysis, and response to risk factors that form part of the life of a business. Effective risk management means attempting to control, as much as possible, future outcomes by acting proactively rather than reactively. Therefore, effective risk management offers the potential to reduce both the possibility of a risk occurring and its potential impact.



### **Step 1: Identify the Risk**

- It is important to identify as many of these risk factors as possible. In a manual environment, these risks are noted down manually. If the organization has a risk management solution employed all this information is inserted directly into the system.

### **Step 2: Analyze the Risk.**

- Once a risk has been identified it needs to be analyzed. The scope of the risk must be determined. It is also important to understand the link between the risk and different factors within the organization. To determine the severity and seriousness of the risk it is necessary to see how many business functions the risk affects. There are risks that can bring the whole business to a standstill if actualized, while there are risks that will only be minor inconveniences in the analysis.

### **Step 3: Evaluate the Risk or Risk Assessment.**

- Risks need to be ranked and prioritized. Most risk management solutions have different categories of risks, depending on the severity of the risk. A risk that may cause some inconvenience is rated lowly, risks that can result in catastrophic loss are rated the highest. It is important to rank risks because it allows the organization to gain a holistic view of the risk exposure of the whole organization. The business may be vulnerable to several low-level risks, but it may not require upper management intervention. On the other hand, just one of the highest-rated risks is enough to require immediate intervention.

### **Step 4: Treat the Risk.**

- Every risk needs to be eliminated or contained as much as possible. This is done by connecting with the experts of the field to which the risk belongs. In a manual environment, this entails contacting each stakeholder and then setting up meetings so everyone can talk and discuss the issues. The problem is that the discussion is broken into many different email threads, across different documents and spreadsheets, and many different phone calls. In a risk management solution, all the relevant stakeholders can be sent notifications from within the system. The discussion regarding the risk and its possible solution can take place from within the system. Upper management can also keep a close eye on the solutions being suggested and the progress being made within the system. Instead of everyone contacting each other to get updates, everyone can get updates directly from within the risk management solution.

## **Step 5: Monitor and Review the Risk .**

- Not all risks can be eliminated – some risks are always present. Market risks and environmental risks are just two examples of risks that always need to be monitored. Under manual systems monitoring happens through diligent employees. These professionals must make sure that they keep a close watch on all risk factors. Under a digital environment, the risk management system monitors the entire risk framework of the organization. If any factor or risk changes, it is immediately visible to everyone. Computers are also much better at continuously monitoring risks than people. Monitoring risks also allows your business to ensure continuity. We can tell you How you can create a risk management plan to monitor and review the risk.

### **Common risks in development project.**

#### **1. Technology risk.**

- The technological aspect of running a project is a complex deliverable because there is a high turnover of new and advanced technologies. The tech aspect of a project poses a critical threat to data security, organization services, compliance, and information security. Risks associated with technology are more challenging because implementing new IT programs often requires fresh personnel training and software acquisition. There are also other technological-related risks like service outages that might lead to delays and project failure.

#### **2. Communication risk.**

- Effective and timely communication is a significant work ethic that you must strictly observe when you oversee a project. Setting up meetings with stakeholders, such as project donors, helps you track any changes, reassign tasks, and foster a cohesive team environment. With all the communication channels and gadgets at our disposal, sometimes team members neglect the critical components of effective communication, leading to loss of data or misinformation and eventual project disruption.

#### **3. Scope creep risk.**

- Uncontrolled and unauthorized change to the initial intended project scope may lead to the extra cost of additional features, products or functions. Almost all projects face this risk, and sometimes it poses an irreversible challenge because some of the added functions are significant to the project and desirable to the project's success.

#### **4. Cost risk.**

- A shortage or mismanagement of project funds resulting from an inflated budget or other constraints is a threat to the project's completion. When the project cost is higher than the budgeted funds, the risk might shift to other operations and workforce segments. The reduction of the funds may also contribute to an occurrence of a scope risk.

#### **5. Operational risk.**

- A project may stall or terminate if there is a poor implementation of critical operations and core processes such as production or procurement. The risks could result in a direct or indirect loss owing to inadequacy or failed qualitative, quantitative or strategies. Depending on the project type, operational risks are:

- IT system risk.
- Human and process direct implementation risk.

- Human and process indirect implementation risk.
- Financial capacity risk.

### **Apply to TS project**

- In our team project, we still must face some risks when we choose to use prototype model for this project.
- These are some risks in our Tune Source project.

### **Risks Assessment**

**Risk #1:** Dishonesty in the way of working.

**Likelihood of risk:** The possibility of that is very high.

**Potential impact on the project:** This risk greatly affects people's time and work up to 80%.

**Way to address this risk:** As a project manager, you must get a clear explanation from the client of what they need and listen carefully to all project stakeholders as they provide input. A clear, shared vision can prevent problems and provide inspiration for the whole team.

**Risk #2:** Trouble with project discussion.

**Likelihood of risk:** There is a high probability about this.

**Potential impact on the project:** If this is not resolved soon, it will lead to project delays and in turn will lead to distance between the people working with each other.

**Way to address this risk:** To reduce the risk of discussion and discussion in your project to a minimum that will allow your team to remain productive. You may also need to train some team members to develop skills. better communicate.

**Risk #3:** Inefficient division of work.

**Likelihood of risk:** Low risk.

**Potential impact on the project:** This risk will not reduce the progress of the project

**Way to address this risk:** Detailed planning is essential to creating a project schedule. When dividing the project properly and one more thing, it must be fair to everyone.

**Risk #4:** Lazy at work.

**Likelihood of risk:** The probability of the risk is very high.

**Potential impact on the project:** This risk could be a big problem if not handle properly also might impact completion time of the project.

**Way to address this risk:** Some people when lazy the manager will not know. To minimize the risk associated with some people being lazy, which will lead to project delays. To solve this problem, you must find yourself someone who is enthusiastic and dedicated to the project.

### **P3. EXPLAIN THE PURPOSE OF A FEASIBILITY REPORT.**

#### **1. Definition Of Feasibility Report/Study.**

A Feasibility Study Report (FSR) is a formally documented output of feasibility study that summarizes results of the analysis and evaluations conducted to review the proposed solution and investigate project alternatives for the purpose of identifying if the project is really feasible, cost-effective and profitable. It describes and supports the most feasible solution applicable to the project.

#### **2. Key components of a feasibility report.**

- The project scope.
- Current market trends and detailed risk analysis.
- The Business requirements.
- The Business approach.
- Technical and organizational requirements for project viability .
- A comprehensive cost analysis for the project economic feasibility .
- Evaluation.

#### **3. Purpose Of A Feasibility Report.**

##### **• Purpose of each element in a Basic Feasibility Report:**

The process to write the report is called feasibility study reporting. Often it is a responsibility of the project manager to control such a process. The importance of writing the report consists in providing legal and technical evidence of the project's vitality, sustainability and cost-effectiveness. The reporting process allows the senior management to get the necessary information required for making key decisions on budgeting and investment planning. A well-written feasibility study report template lets develop solutions for.

- Project Analysis because an example of FSR helps link project efficiency to budgeted costs.
- Risk Mitigation because it helps with contingency planning and risk treatment strategy development.
- Staff Training because the report can be used by senior management to identify staffing needs as well as acquire and train necessary specialists.

The process of reporting is the trigger to run the project investing process through underpinning the business case document, stating the reasons for undertaking the project, and analyzing project costs and benefits.

##### **• Explanations of purpose of feasibility study:**



The purpose of a feasibility study is to determine the likelihood of successful project completion. Project managers use feasibility studies to differentiate pros and cons, opportunities and threats as presented by environments, resources required through execution, and ultimately project implementation expectations before they invest a lot of time and money into it.

## P4 DESCRIBE HOW TECHNICAL SOLUTIONS CAN BE COMPARED.

### 1. Introduction.

- As same as the other project, Tune Source project also needs to be evaluated in different feasibility criteria. In this part, I will discuss about how these feasibility criteria are applied into the project and how feasible the project is.

### 2. Feasibility study of Tune Source project.

#### 2.1. Technical feasibility.

- Firstly, we will apply technology feasibility into Tune Source project. We will discuss about how much our team familiar with application and technology, together with the project size and the compatibility with this project.

**Familiar with application:** Good.

**Familiar with technology:** High.

**Project size:** Small.

**Compatibility:** High.

- Our team has more than three-year experience working with website. This project aims to add some new functions to an already existed website. The basic Tune Source website is built by Java technology, which we have the longest time working with. Because of the website requires new functions write from Java language as same as the old website, we can easily add more functions without any problem. This is a small project so that our team has 7 members, a suitable number of human resource for this project. To conclusion, we has fulfilled the requirements to start Tune Source project immediately.

#### 2.2. Economic feasibility.

- Secondly, economic criteria is also an important factor that we should not miss. This can make the whole project fail if we are not manage it properly. We will divide into four elements: development cost, annual operating cost, annual benefits and intangible benefits.

#### **Cost Project Development**

- Development team salary: 190.000\$
- Hardware and software: 260.000\$
- Data conversion cost: 80.000\$

#### **Annual Operating cost**

- Software licensing fee: 14.000\$
- Communication change: 48.000\$
- Operational team salary: 90.000\$

#### **Annual Benefits**

- Increase sale: 2.308.000\$
- Reduction in staff: 800.000\$

#### **Intangible benefits and cost**

- Increase market share: 650.000\$
- Improve customer service: 200.000\$

• According to the table above, this project will bring huge benefits to our company. Because this is only a small project, our total cost is not too high. With this new website, the money we take back is much higher the money we have to pay in the development and operating process. The table above only conclude the money we will take back in a short-time period. According to the newest survey, there are many customers waiting for the release of this new website. Our benefits may be higher than what we expect.

### 2.3. Organisational feasibility

- Thirdly, I will discuss about the organisational feasibility of this project. There are five topics to talk about in this criteria: project champions, senior management, users, other stakeholders and strategic business align with this project.

**Project champion:**

Carly Edwards, Assistant Vice President, Marketing.

**Senior management:**

Adam Smith, Charles Jones.

**Users:**

Tune Source customers.

**Other stakeholders:**

Internet Service Provider (ISP)

- The project champion and project sponsor is Carly Edwards, Assistant Vice President, Marketing. Our team senior management team is Adam Smith and Charles Jones. Both of them are experience senior who had managed many projects from small to large. The users of this website is Tune Source customers. These customers are waiting for our new website release date. We have one business align with this project: Internet Service Provider (ISP). This company is cooperated with Tune Source IT department to maintain the website.

## 3. Provide an alternative matrix for Tune Source project.

### 3.1. Introduction.

- Alternative evaluation matrix is a matrix specific in comparing alternatives for numerous requirements such as hardware, software, database, etc. In this part, I will apply weighted alternative evaluation matrix to choose the most suitable programming language for this project.

### 3.2. ASP.NET technology.

#### 3.2.1. Definition of ASP.NET technology.

- ASP.NET is an open-source web framework for building web apps on the .NET (dotNET) framework. It is created by Microsoft and version 1.0 was released in 2002 to allow developers to build dynamic web apps, services, and sites. ASP.NET is the successor to the ASP (Active Server Pages) technology and was a significant upgrade in terms of flexibility and power. It is an extension of the .NET platform with additional tools and libraries specifically for building things on the web, including web apps and websites.



- ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices.
- ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation.

### **3.2.2. Advantages of ASP.NET**

- The .NET framework provides a consistent, object-oriented programming environment.
- It promotes the safe execution of code.
- Eliminates the performance problems of scripted or interpreted environments.
- It minimizes software deployment and versioning conflicts.
- A consistent experience for both developers and users across various types of Windows-based and Web-based applications on multiple devices.
- Communication built on the industry standards to ensure that code based on the .NET Framework can integrate with any other code.

### **3.2.3. Disadvantages of ASP.NET**

- It doesn't support many platforms like Java.
- It doesn't support many CMS.
- It does not provide the same JavaScript on the server and client-side which could be easy for coding.
- ASP.NET core is not good
- Hard to make changes in the app

## **3.3. JavaScript technology**

### **3.3.1. Introduction**

JavaScript is a lightweight, cross-platform, and interpreted compiled programming language which is also known as the scripting language for webpages. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments. JavaScript is both imperative and declarative type of language. JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.

### **3.3.2. Advantages of JavaScript technology**

- Client-side JavaScript is very fast because it can be run immediately within the client-side browser.
- JavaScript is relatively simple to learn and implement.
- JavaScript is used everywhere on the web.
- JavaScript plays nicely with other languages and can be used in a huge variety of applications.
- Being client-side reduces the demand on the website server.
- Gives the ability to create rich interfaces.

### **3.3.3. Disadvantages of Java Script technology.**

- It can be exploited for malicious purposes.
- JavaScript is sometimes interpreted differently by different browsers.

## **3.4. PHP technology.**

### **3.4.1. Introduction.**

- PHP(short for Hypertext PreProcessor) is the most widely used open source and general purpose server side scripting language used mainly in web development to create dynamic websites and applications. It is not only used to build the web apps of many tech giants like Facebook but is also used to build many CMS (Content Management System) like WordPress, Drupal, Shopify, WooCommerce etc.
- PHP can actually do anything related to server-side scripting or more popularly known as the backend of a website. There are also many hash functions available in PHP to encrypt user's data that makes PHP secure and reliable to be used as a server-side scripting language. So these are some of the abilities of PHP that makes it suitable to be used as server-side scripting language.

### **3.4.2. Advantages of PHP technology.**

- It is developed and maintained by a large group of developers.
- It is relatively fast, since it uses not much system resources.
- It uses a C like syntax, so for those who are familiar with C, it's very easy for them to pick up and easy to create website scripts.
- Since it is maintained by many developers, bugs are rather found and fixed quickly, making it a stable software.
- You can easily find functional modules you need such a PDF, graph etc.
- You can connect to databases easily using PHP, since many websites are data/content driven, so we will use database frequently, this will largely reduce the development time of web apps.

### **3.4.3. Disadvantages of PHP technology.**

- Since it is open sourced, all people can see the source code. If there are bugs in the source code, it can be used by people to explore the weakness of it.
- It will be difficult to use it for programming huge applications.
- Implicit conversion may surprise unwary programmers and lead to unexpected bugs.
- The framework has a bad error handling method.
- The technology is helpless to support a bunch of apps.

## **4. Weighted Alternative Matrix.**

		<b>ASP.NET technology</b>		<b>Java Script technology</b>		<b>PHP technology</b>	
Criteria	Weighting Factor	Score	Weighting Score	Score	Weighting Score	Score	Weighting Score
Cost	5	2	10	3	15	2	10
Development Time	3	2	6	3	9	1	3
Maintenance Cost	2	3	6	2	6	1	3
Integration	1	1	1	2	2	3	3
Total Score			23		32		19

## **5. Solution for TS and Explain.**

### **5.1 Introduction.**

Here there are 4 factors i have sorted in descending order equivalent to most important being largest here i rank cost as most important because it is needed to maintain the system, followed by time development time because the system can crash if we miss the deadline, followed by the maintenance cost here we consider it not as important as the cost to build eg you spend a lot of money to buy one car but it costs you very little to maintain your car if the maintenance cost is as high as or higher than the cost of buying the car then you should buy a new one, in the end integrating it is not important in our project.

### **5.2 Tune Source Project.**

Looking at the table we can see that Java outperforms ASP.NET and PHP. it can save money and resources while still producing great results, besides we have a sufficient knowledge base to speed up our time when we use ASP and PHP. In terms of maintenance cost we have enough knowledge about Java than PHP or ASP so we can minimize the chance of errors as much as possible in other words PHP and ASP we have less knowledge than Java. Finally, while the integration element in this project is small, it is not ignored, and JavaScript responded to that request. So I choose Java through the above criteria.

## Reference

- Geeksforgeeks.org (2022) What is php and why we use it [online] Available at: <https://www.geeksforgeeks.org/what-is-php-and-why-we-use-it/> [Accessed at: June 12, 2022]
- Deepak (2019) advantages and disadvantages of php [online] phpbabu.com Available at: <https://www.phpbabu.com/advantages-and-disadvantages-of-php/> [Accessed at: June 12, 2022]
- Indeed.com (2021) What is a feasibility report? [online] Available at: <https://www.indeed.com/career-advice/career-development/feasibility-report> [Accessed at: June 12, 2022]
- Sunanda (2022) Project feasibility study [online] blog.masterproject.com. Available at: <https://blog.masterofproject.com/project-feasibility-study/> [Accessed at: June 12, 2022]
- Webeduclick.com (2022) advantages and disadvantages of asp.net [online] Available at: <https://webeduclick.com/advantages-and-disadvantages-of-asp-net/> [Accessed at: June 12, 2022]
- SDLC - Waterfall Model [Online] Tutorialspoint. Available at: [SDLC - Waterfall Model \(tutorialspoint.com\)](https://www.tutorialspoint.com/sdgc/sdgc_waterfall_model.htm) [Accessed at: June 12, 2022]
- SDLC - Prototyping Model [Online] Seachcio. Available at: [What is the Prototyping Model? \(techtargget.com\)](https://www.seachcio.com/what-is-the-prototyping-model/) [Accessed at: June 12, 2022]
- SDLC - Agile Model [Online] Tutorialspoint. Available at: [SDLC - Agile Model \(tutorialspoint.com\)](https://www.tutorialspoint.com/sdgc/sdgc_agile_model.htm) [Accessed at: June 12, 2022]
- Feasibility study [Online] wikipedia . Available at : [Feasibility study - Wikipedia](https://en.wikipedia.org/wiki/Feasibility_study) [Accessed at: June 12, 2022]