ASSIGNMENT 1 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 9: Software Development Life Cycle		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Nguyen Dong Anh	**Student ID**	GCC19072
Class	GCC0801-1631	**Assessor name**	Nguyễn Hùng Dũng
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	Anh

Grading grid

P1	P2	P3	P4	M1	M2	D1	D2

❒ Summative Feedback:	❒ Resubmission Feedback:	
Grade:	**Assessor Signature:**	**Date:**
Internal Verifier's Comments:		
Signature & Date:		

Submission Format:	
Format:	The submission is in the form of 1 document. 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.
Submission:	You **must** submit the assignment **by the due date** and follow the submission method specified by the Tutor. The submission form is **soft copy**, which is to be uploaded to the following URL: http://cms.greenwich.edu.vn.
Note:	Your assignment *must* be your own work, and not copied by or from another student or from other sources, such as book etc. If you use ideas, quotes or data (such as diagrams) from books, journals or other sources, you must reference the sources, using the Harvard style. Make sure that you know how to reference properly and that you understand the plagiarism guidelines. **Plagiarism is a very serious offence**, which will result in a failing grade.

Unit Learning Outcomes:

LO1 Describe different software development lifecycles.
LO2 Explain the importance of a feasibility study**.**

Assignment Brief and Guidance:

System 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 underutilised.
- 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. (P1) Describe the following SDLC models: waterfall, v-model, prototyping, agile and spiral. Choose one that you think suitable for the project and explain why.
    - 350 - 500 words for each model
    - Explanation: 400 – 600 words

(M1) Discuss the suitability of each of the SDLC models for the project. For each model, specify whether it is most, moderately or least suitable.

- Discussion and arguments: 800 - 1000 words

(D1) Discuss the merits of applying the waterfall model to a large software development project.

- Discussion: 800 – 1200 words

2. (P2) 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 Plan to manage risks of TuneSource project

Task 2 – Feasibility study

1. (P3) Discuss the purpose of conducting a feasibility study for the project.
   - Discussion: 400 – 1600 words
2. (P4) Discuss how the three feasibility criteria (technical, economic, organisational) are applied to the project. Discuss whether the project is feasible.

   Discuss alternative technical solutions using the alternative matrix.

   Discussion: 1200 – 1500 words
3. (M2) Explain the components of a feasibility report.
   - Discussion economic feasibility study: 350 – 500 words
   - Discussion organizational feasibility study: 350 – 500 words
4. (D2) Assess the impact of each feasibility criterion on a software investigation.
   - Discussion and represent as feasibility alternatives matrix for: 700 – 900 words

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

TABLE OF CONTENTS

Chapter 1: Describe different software development lifecycles

A – Describe the following SDLC models: waterfall, v-model, prototyping, agile and spiral, suitable model for the project and explanation (P1)

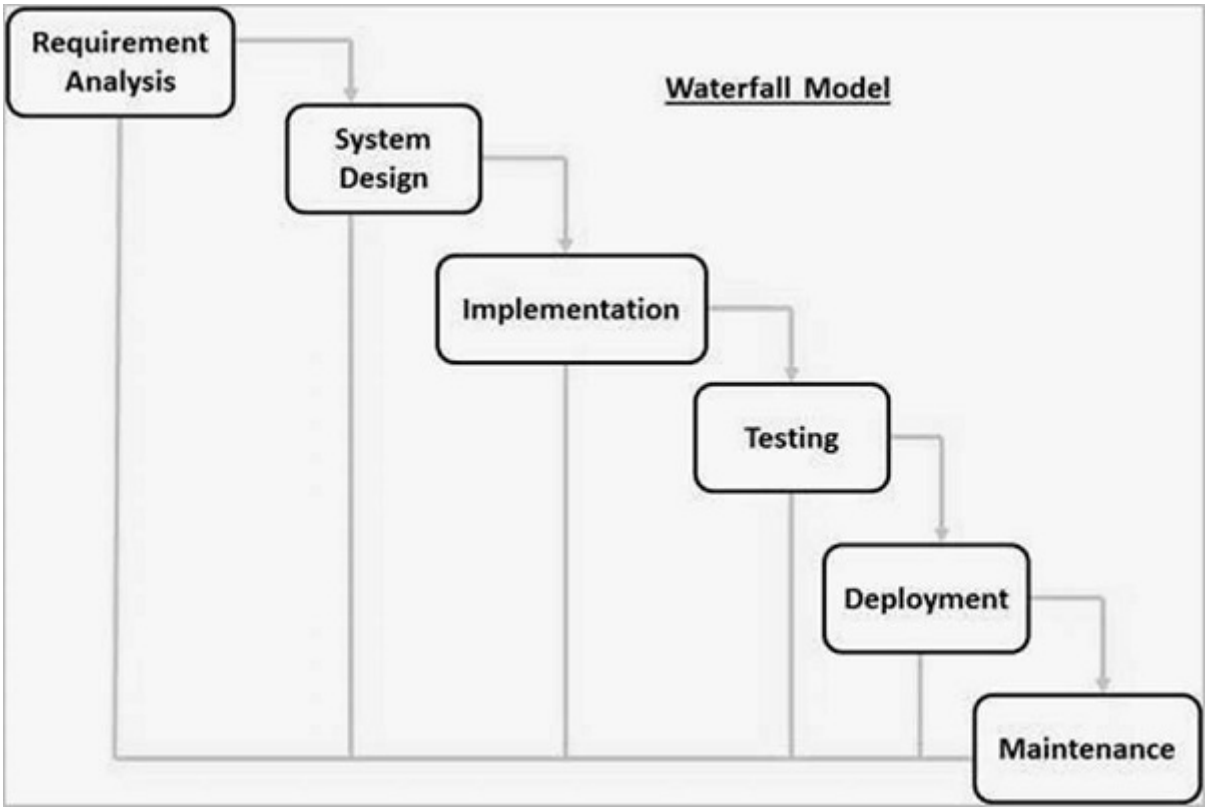
I will be a project manager of a company named ABC. My company has been hired by Tune Source to carry out a project that helps them develop a software for the requirements specified in the system request. As the first step, I will:

Describe these following SDLC models: waterfall, v-model, prototyping, agile and spiral. And I will choose one that I think suitable for the project and explain why.

1. Waterfall:

The 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(Balaji, 2012)

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.

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.

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.

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.

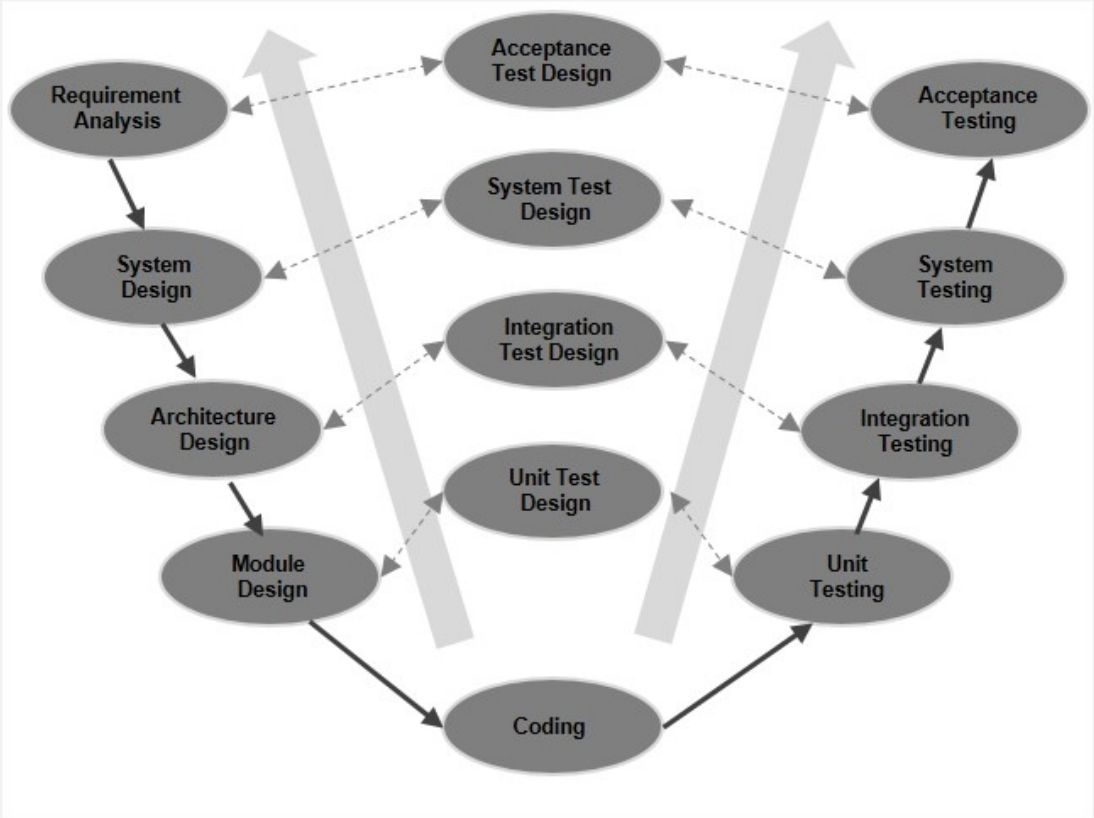
The major disadvantages of the Waterfall Model are as follows:

- 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. V-model:

The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as Verification and Validation model. The V-Model is also an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage.

Under the V-Model, the corresponding testing phase of the development phase is planned in parallel. So, there are Verification phases on one side of the 'V' and Validation phases on the other side. The Coding Phase joins the two sides of the V-Model(Balaji, 2012)



There're three phases for the V-model activities:

- Verification Phases:

- Business Requirement Analysis: This phase involves detailed communication with the customer to understand his expectations and exact requirement.
- System Design: Once you have the clear and detailed product requirements, it is time to design the complete system. The system design will have the understanding and detailing the complete hardware and communication setup for the product under development.
- Architectural Design: Architectural specifications are understood and designed in this phase. Usually more than one technical approach is proposed and based on the technical and financial feasibility the final decision is taken. The system design is broken down further into modules taking up different functionality. This is also referred to as High Level Design (HLD).
- Module Design: The detailed internal design for all the system modules is specified, referred to as Low Level Design (LLD).

- Coding Phase:

The best suitable programming language is decided based on the system and architectural requirements.The coding is performed based on the coding guidelines and standards. The code goes through numerous code reviews and is optimized for best performance before the final build is checked into the repository.

- Validation Phases:

- Unit Testing: Unit testing is the testing at code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.
- Integration Testing: Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.
- System Testing: System testing is directly associated with the system design phase. System tests check the entire system functionality and the communication of the system under development with external systems.
- Acceptance Testing: Acceptance tests uncover the compatibility issues with the other systems available in the user environment. It also discovers the non-functional issues such as load and performance defects in the actual user environment.

Advantages

The advantage of the V-Model method is that it is very easy to understand and apply. The simplicity of this model also makes it easier to manage.

The advantages of the V-Model method are:

- This is a highly-disciplined model and Phases are completed one at a time.
- Works well for smaller projects where requirements are very well understood.

- 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.

Disadvantages

The disadvantage is that the model is not flexible to changes and just in case there is a requirement change, which is very common in today's dynamic world, it becomes very expensive to make the change.

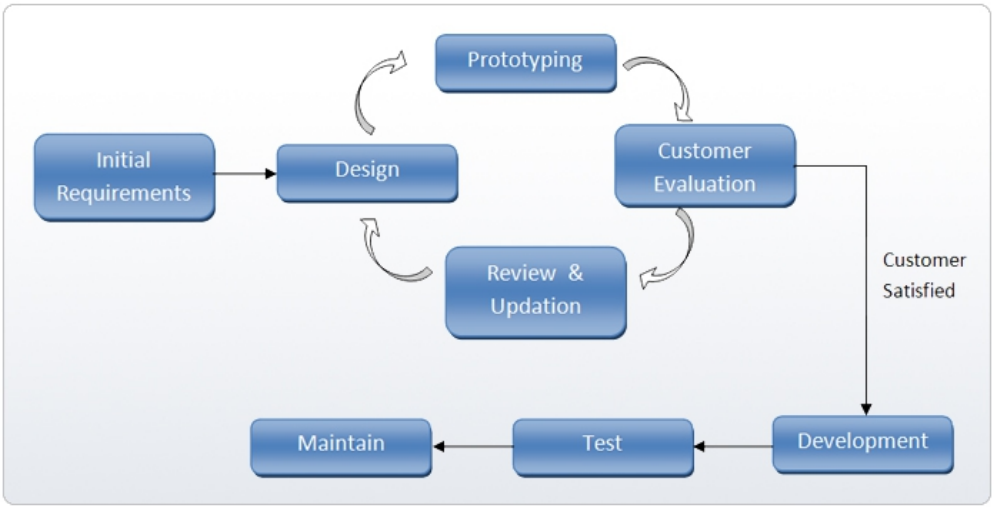
The disadvantages of the V-Model method are:

- High 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.
- Once an application is in the testing stage, it is difficult to go back and change a functionality.
- No working software is produced until late during the life cycle.

3. Prototyping:

Prototype is a working model of software with some limited functionality. The prototype does not always hold the exact logic used in the actual software application and is an extra effort to be considered under effort estimation(Carter, 2001)

It is becoming very popular as a software development model, as it enables to understand customer requirements at an early stage of development. It helps get valuable feedback from the customer and helps software designers and developers understand about what exactly is expected from the product under development.



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

1) 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.

2) A preliminary, simple design is created for the new system.

3) 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.

4) 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.

5) 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 then evaluated in the same manner as was the first prototype.

6) 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.

7) 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.

Types of prototype models

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.

Increment: 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.

Advantages

- Increased user involvement in the product even before its implementation.
- Since a working model of the system is displayed, the users get a better understanding of the system being developed.
- Reduces time and cost as the defects can be detected much earlier.
- Quicker user feedback is available leading to better solutions.
- Missing functionality can be identified easily.
- Confusing or difficult functions can be identified.

Disadvantages

- Risk of insufficient requirement analysis owing to too much dependency on the prototype.

- Users may get confused in the prototypes and actual systems.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- Developers may try to reuse the existing prototypes to build the actual system, even when it is not technically feasible.
- The effort invested in building prototypes may be too much if it is not monitored properly.

4. Agile:

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(Balaji, 2012) Every iteration involves cross functional teams working simultaneously on various areas like:

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing
- Acceptance Testing.

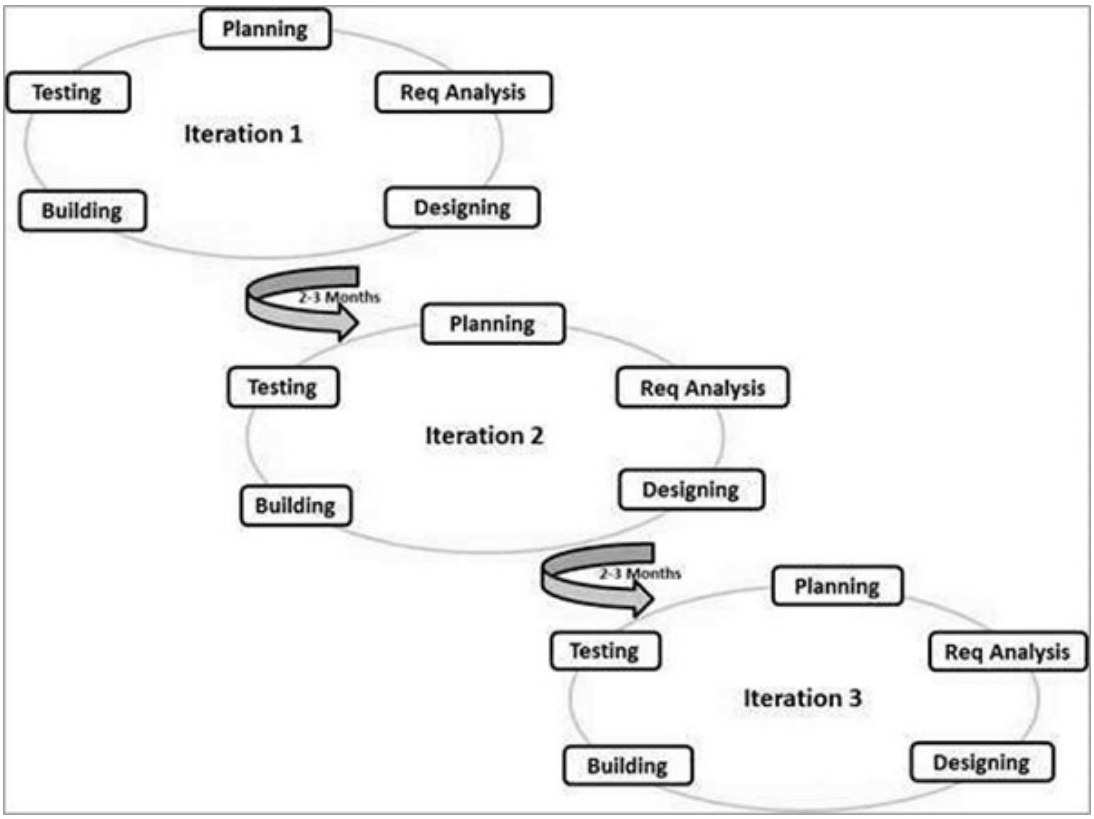
At the end of the iteration, a working product is displayed to the customer and important stakeholders.

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.

Agile Methods:

- **Scrum:** is an agile development methodology used in the development of Software based on an iterative and incremental processes. Scrum is adaptable, fast, flexible and effective agile framework that is designed to deliver value to the customer throughout the development of the project.
- **eXtreme Programming(XP):** is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.



The advantages:

- 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.

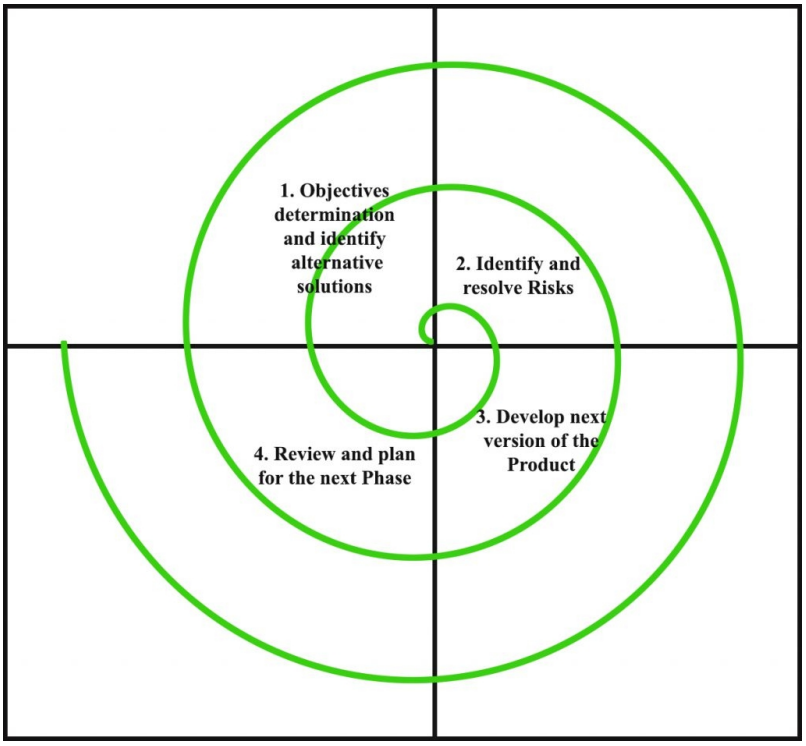
The disadvantages:

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.

- An overall plan, an agile leader and agile project management 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.

5. Spiral:

Spiral model is one of the most important SDLC models, which provides support for Risk Handling. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a Phase of the software development process(Boehm, 1988) The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using spiral model(Boehm, 1988)



Each phase of Spiral Model is divided into four quadrants as shown in the figure below.

1. **Objectives determination and identify alternative solutions:** Requirements are gathered from the customers and the objectives are identified, elaborated and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.

2. **Identify and resolve Risks:** During the second quadrant all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution is identified and the risks are resolved using the best possible strategy. At the end of this quadrant, Prototype is built for the best possible solution.

3. **Develop next version of the Product:** During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.

4. **Review and plan for the next Phase:** In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

Advantages

- **Risk Handling:** The projects with many unknown risks that occur as the development proceeds, in that case, Spiral Model is the best development model to follow due to the risk analysis and risk handling at every phase.
- **Good for large projects:** It is recommended to use the Spiral Model in large and complex projects.
- **Flexibility in Requirements:** Change requests in the Requirements at later phase can be incorporated accurately by using this model.
- **Customer Satisfaction:** Customer can see the development of the product at the early phase of the software development and thus, they habituated with the system by using it before completion of the total product.

Disdvantages

- **Complex:** The Spiral Model is much more complex than other SDLC models.
- **Expensive:** Spiral Model is not suitable for small projects as it is expensive.
- **Too much dependable on Risk Analysis:** The successful completion of the project is very much dependent on Risk Analysis. Without very highly experienced expertise, it is going to be a failure to develop a project using this model.
- **Difficulty in time management:** As the number of phases is unknown at the start of the project, so time estimation is very difficult.

The most suitable SDLC model for the project and the reasons:

The one that I think is suitable for the project is the Agile model. Unlike Waterfall, the Agile method prioritizes shorter planning phases, staged delivery, and regular customer interaction. Because Agile model is model with focus on process adaptability and customer satisfaction by rapid delivery of working software product. It is also easy to manage, we don't need to invest too much resources like the other models, the time to deliver is early and the flexibility for our project will also be higher than the other models.

Also since the customers might have a number of music download options available to them from a different company elsewhere so we need to bring this system to the market as soon as possible, this agile model will be be the best choice.

Agile model requires back and forth communication between developers and customers. This model has good communication channels and advanced collaborative tools in place to ensure a smooth flow of information.

If we are looking for any Agile drawback, then let it be the fact that it requires special skills. The method does not have a rigorous planning stage, so it also pays less attention to the design and documentation of requirements.

For the method of Agile model we will be using Crum, which is an agile development methodology used in the development of Software based on an iterative and incremental processes. Scrum is adaptable, fast, flexible and effective agile framework that is designed to deliver value to the customer throughout the development of the project. Scrum teams work in "sprints," which usually last two to four weeks, to complete assigned tasks. Daily Scrum meetings help the whole team monitor progress throughout the project.

In my opinion about Agile model, a "fast failure" is a good thing. The approach produces ongoing release cycles, each featuring small, incremental changes from the previous release. At each iteration, the product is tested. The Agile model helps teams identify and address small issues on projects before they evolve into more significant problems, and engage business stakeholders and get their feedback throughout the development process.

B – Identify some risks and discuss an approach to manage them for the suitable model mentioned above(P2)

What is Risk Management?

To identify risks and discuss an approach to manage those risks we need to know what is Risk Management. Risk management is an important business practice that helps businesses identify, evaluate, track, and mitigate the risks present in the business environment. Risk management is practiced by the business of all sizes, small businesses do it informally, while enterprises codify it(Edwards, 2013)

Why is Risk Management Important?

Risk management is important because it tells businesses about the threats in their operating environment and allows them to preemptively mitigate risks. In the absence of risk management, businesses would face heavy losses because they would be blindsided by risks(Edwards, 2013)

Risk Management Process

The risk management process is a framework for the actions that need to be taken. There are five basic steps that are taken to manage risk; these steps are referred to as the risk management process. It begins with identifying risks, goes on to analyze risks, then the risk is prioritized, a solution is implemented, and finally, the risk is monitored(Edwards, 2013) In manual systems, each step involves a lot of documentation and administration.

Step 1: Identify the Risk

The first step is to identify the risks that the business is exposed to in its operating environment. There are many different types of risks – legal risks, environmental risks, market risks, regulatory risks, and much

more. 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. The advantage of this approach is that these risks are now visible to every stakeholder in the organization with access to the system. Instead of this vital information being locked away in a report which has to be requested via email, anyone who wants to see which risks have been identified can access the information in the risk management 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. In a manual risk management environment, this analysis must be done manually. When a risk management solution is implemented one of the most important basic steps is to map risks to different documents, policies, procedures, and business processes. This means that the system will already have a mapped risk framework that will evaluate risks and let you know the far-reaching effects of each risk.

Step 3: Evaluate or Rank the Risk

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 and every 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.

Even under a digital environment, the basics of the risk management process stay the same. What changes is how efficiently these steps can be taken, and as it should be clear by now, there is simply no competition between a manual risk management system and a digital one.

Risk Management Plan

A risk management plan is a document prepared by a project manager to help foresee any risks, estimate their impact and to plan responses to issues that could occur. Also contained in the document is a risk assessment plan.

A risk is inherent with any project and should be assessed and planned for continually. The risk management plan contains the analysis of potential risks, both high and low impact, along with mitigation strategies to allow the project not to be derailed if an issue arises. The plans should be reviewed from time to time by the project team to keep them up to date and relevant showing a truly potential project risk(Edwards, 2013)

The most crucial part of a risk management plan is the risk strategy. Generally, there are four potential strategies that can be chosen from. Projects can choose to:

- **Avoid the risk:** Plans within the project can be changed around in order to avoid the risk occurring completely.
- **Control/Mitigate the risk:** This describes when steps are taken to reduce the risk's potential impact and/or likelihood.
- **Accept the risk:** This is when teams simply accept the risk and what effect (positive or negative) it could have on the project. The fact that the risk is identified is enough preparation for it in this strategy.
- **Transfer the risk:** This can be done by transferring the risk to a third party that can manage its outcome. This can be done through insurance companies or outsourcing an activity that carries a high risk.

Risk management plans are used in many projects and serve as a visualisation of what could happen and what actions are to be taken in certain situations. The scope of the risks is defined and tracked by means of documentation.

Risk Description	Impact	Likelihood	Action Required
Lacking details in task descriptions.	Small	Medium	Make sure all details are present and clear for the team so they know exactly what they are creating.
The TuneSource company keeps changing their mind or adds on new features as the team progresses	Medium	Medium	While the TuneSource company is in charge of making sure the project requirements are being met, they also need to be made aware whenever they request work that deviates from the original plan
Key personnel leave the project taking critical information with them that significantly delays or derails the project.	Medium	Small	Increased collaboration and information sharing on the team. Have them plan together, share knowledge, complete code reviews, and work together on a given project from beginning to end.
Projects going over budget	Medium	Small	The teams should make decisions when they are in the best position rather than presenting very detailed plans at the beginning of the project.

Chapter 2: Explain the importance of a feasibility study

A – Discuss the purpose of conducting a feasibility study for the project(P3)

What is Feasibility Study?

A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, legal, and scheduling considerations, to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it(Shen, 2010)

Feasibility studies also can provide a company's management with crucial information that could prevent the company from entering blindly into risky businesses.

Purpose

The main purpose of a feasibility study is to assess the financial viability of developed land and whether it will be a success or failure. The study aims to highlight problems and risks that a potential plot will face such as gaining planning, local authority, assessing current development supply and demand and the suitability of a site, neighbours and so forth(Shen, 2010)

Types of Feasibility Study

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions(Shen, 2010) There are five types of feasibility study, separate areas that a feasibility study examines, described below.

1. Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system.

2. Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility, helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

3. Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

4. Scheduling Feasibility

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

5. Organizational Feasibility

This assessment is conducted to determine whether a proposed business has sufficient management expertise, organizational competence, and resources to successfully launch its business. Two key aspects to consider include management ability and resource sufficiency.

The importance of Feasibility Study

Feasibility study is an important stage of Software Project Management Process as after completion of feasibility study it gives a conclusion of whether to go ahead with proposed project as it is practically feasible or to stop proposed project here as it is not right, or feasible to develop or to think and analyze about proposed project again(Shen, 2010)

Moreover Feasibility Study helps in identifying risk factors involved in developing and deploying system and planning for risk analysis also narrows the business alternatives and enhance success rate analyzing different parameters associated with proposed project development.

Below are some key benefits of conducting a feasibility study:

- Improves project teams' focus
- Identifies new opportunities
- Provides valuable information for a "go or no-go" decision
- Narrows the business alternatives
- Identifies a valid reason to undertake the project
- Enhances the success rate by evaluating multiple parameters
- Aids decision-making on the project
- Identifies reasons not to proceed

Apart from the approaches to feasibility study listed above, some projects also require other constraints to be analyzed:

- Internal Project Constraints: Technical, Technology, Budget, Resource, etc.
- Internal Corporate Constraints: Financial, Marketing, Export, etc.
- External Constraints: Logistics, Environment, Laws, and Regulations, etc.

B – Discuss how the three feasibility criteria (technical, economic, organisational) are applied to the project. Discuss whether the project is feasible(P4)

Technical Feasibility

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project. The technical requirements are then compared to the technical capability of the organization. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements(Shen, 2010)
The analyst must find out whether it is possible to develop the new system given the current technical

resources. If not, can the system be upgraded or added to in a manner that fulfills the request under consideration? If existing systems cannot be added onto or upgraded, the next question becomes whether there is technology in existence that meets the specifications(Shen, 2010) At the same time, the analyst can ask whether the organization has the staff who are technically proficient enough to accomplish the objectives. If not, the question becomes whether they can hire additional programmers, testers, experts, or others who may have different programming skills from theirs, or maybe outsource the project completely(Shen, 2010)

The essential questions that help in testing the operational feasibility of a system include the following:

- Is the proposed technology or solution practical?
- Do we currently possess the necessary technology?
- Do we possess the necessary technical expertise and is the schedule reasonable for this team?
- Is relevant technology mature enough to be easily applied to our problem?
- What kinds of technology will we need?
- Is the required technology available "in house"?

If the technology is available:

- Does it have the capacity to handle the solution?

If the technology is not available:

- Can it be acquired?

Economic Feasibility

Economic feasibility is the second part of resource determination. The basic resources to consider are your time and that of the systems analysis team, the cost of doing a full systems study (including the time of employees you will be working with), the cost of the business employee time, the estimated cost of hardware, and the estimated cost of software or software development(Shen, 2010)

The concerned business must be able to see the value of the investment it is pondering before committing to an entire systems study. If short-term costs are not overshadowed by long-term gains or produce no immediate reduction in operating costs, the system is not economically feasible and the project should not proceed any further.

Possible questions raised in economic analysis are:

- Is the system cost effective?
- Do benefits outweigh costs?
- What is the minimal cost to attain a certain system?
- How soon will the benefits accrue?
- Which alternative offers the best return on investment?

Things to consider:

- Hardware/software selection
- Selection among alternative financing arrangements (rent/lease/purchase)

The economical feasibility will review the expected costs to see if they are in-line with the projected budget or if the project has an acceptable return on investment. At this point, the projected costs will only be a rough estimate. The exact costs are not required to determine economic feasibility. It is only required to determine if it is feasible that the project costs will fall within the target budget or return on investment. A rough estimate of the project schedule is required to determine if it would be feasible to complete the systems project within a required timeframe. The required timeframe would need to be set by the organization(Shen, 2010)

Organization Feasibility

Organizational feasibility analysis is conducted to determine whether a proposed business has sufficient management expertise, organizational competence, and resources to successfully launch its business(Shen, 2010)

It defines a business's corporate and legal structure. It can also include information on the founders' professional background, the business's principals, and the skills they will contribute to the functions of the business. An organizational feasibility analysis should include:

- The business structure's description
- The organizational structure's description
- The business's internal and external practices and principles
- Resumes and professional skills.

There are two primary issues to consider in this area:

- **Management prowess:** A proposed business should evaluate the prowess, or ability, of its initial management team, whether it is a sole entrepreneur or a larger group. This task requires the individuals starting the firm to be honest and candid in their self assessments. Two of the most important factors in this area are the passion that the solo entrepreneur or the management team has for the business idea and extent to which the management team or solo entrepreneur understands the markets in which the firm will participate. There are no practical substitutes for strengths in these areas.
- **Resource sufficiency:** It is the second area of organizational feasibility analysis is to determine whether the proposed venture has or is capable of obtaining sufficient resources to move forward. The objective is to identify the most important nonfinancial resources and assess whether they are available. An example is a start-up that will require employees with specialized skills. Another key resource sufficiency issue is the ability to obtain intellectual property protection on key aspects of the business.

Comparing Alternatives

An Alternative is another combination of the project's costs, schedules, resources, and risks that allow achieving the same results as compared to the project base-line. It is one or more ways to produce the project and address the business need while using the same resource base yet operating in a new project environment and facing new working conditions. Another name for it is the Project Option(Lichtenberg, 1983)

Often the project alternative is not a close substitute to the original choice yet sometimes it may offer similar solutions but in a new combination and interpretation. The nature of the project alternative makes two alternatives mutually exclusive. Meanwhile, a combination of two or more project alternatives creates a new project alternative(Lichtenberg, 1983)

The alternative matrix

Criteria	**Weight**	**Alternative 1**	**Alternative 2**	**Alternative 3**
Cost	.30	5	3	4
Response Time	.17	5	5	5
Training Time	.17	3	5	5
Ease of use	.17	5	4	5
Strong Team	.10	3	3	4
Team Experience	.10	5	5	5
Total	1.0	26	25	28
Weighted Total		4.5	4.0	4.6

References

Balaji, S. and Murugaiyan, M.S., 2012. Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, *2*(1), pp.26-30.

Carter, R.A., Antón, A.I., Dagnino, A. and Williams, L., 2001, August. Evolving beyond requirements creep: a risk-based evolutionary prototyping model. In *Proceedings Fifth IEEE International Symposium on Requirements Engineering* (pp. 94-101). IEEE.

Boehm, B.W., 1988. A spiral model of software development and enhancement. *Computer*, *21*(5), pp.61-72.

Edwards, P. and Bowen, P., 2013. *Risk management in project organisations*. Routledge.

Shen, L.Y., Tam, V.W., Tam, L. and Ji, Y.B., 2010. Project feasibility study: the key to successful implementation of sustainable and socially responsible construction management practice. *Journal of cleaner production*, *18*(3), pp.254-259.

Lichtenberg, S., 1983. Alternatives to conventional project management. *International Journal of Project Management*, *1*(2), pp.101-102.