ASSIGNMENT 01 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 09: Software Development Life Cycle		
Submission date	30/07/2023	**Date Received 1st submission**	
Re-submission Date	01/09/2023	**Date Received 2nd submission**	
Student Name	Dinh Dinh Khoi	**Student ID**	GCC210345
Class	GCC1003	**Assessor name**	Nguyen Kim Khanh
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	

Grading grid

P1	P2	P3	P4	M1	M2	D1	D2

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

Assignment Brief 01 (RQF)

Higher National Certificate/Diploma in Business

Student Name/ID Number:	
Unit Number and Title:	**Unit 09: Software Development Life Cycle**
Academic Year:	
Unit Assessor:	
Assignment Title:	**Plan a software development life cycle**
Issue Date:	**07/12/2020**
Submission Date:	
Internal Verifier Name:	
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:

- Students are compulsory to submit the assignment in due date and in a way requested by the Tutor.
- The form of submission will be a soft copy posted on http://cms.greenwich.edu.vn/.
- Remember to convert the word file into **PDF** file before the submission on CMS.

Note:

- The individual Assignment must be your own work, and not copied by or from another student.
- 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.
- Make sure that you understand and follow the guidelines to avoid plagiarism. Failure to comply this requirement will result in a failed assignment.

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
LO1 Describe different software development lifecycles	**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.	**D1** Assess the merits of applying the Waterfall lifecycle model to a large software development project.
LO2 Explain the importance of a feasibility study	**P3** Explain the purpose of a feasibility report. **P4** Describe how technical solutions can be compared.	**M2** Discuss the components of a feasibility report.	**D2** Assess the impact of different feasibility criteria on a software investigation.

Table of Contents

I. SDLC models.

1. Waterfall.

a. Definition.

The first Process Model to be introduced was the Waterfall Model. It also called "linear-sequential life cycle model". It is incredibly easy to use and comprehend. There is no overlap between stages in a waterfall model; each step must be finished before the subsequent phase can start.

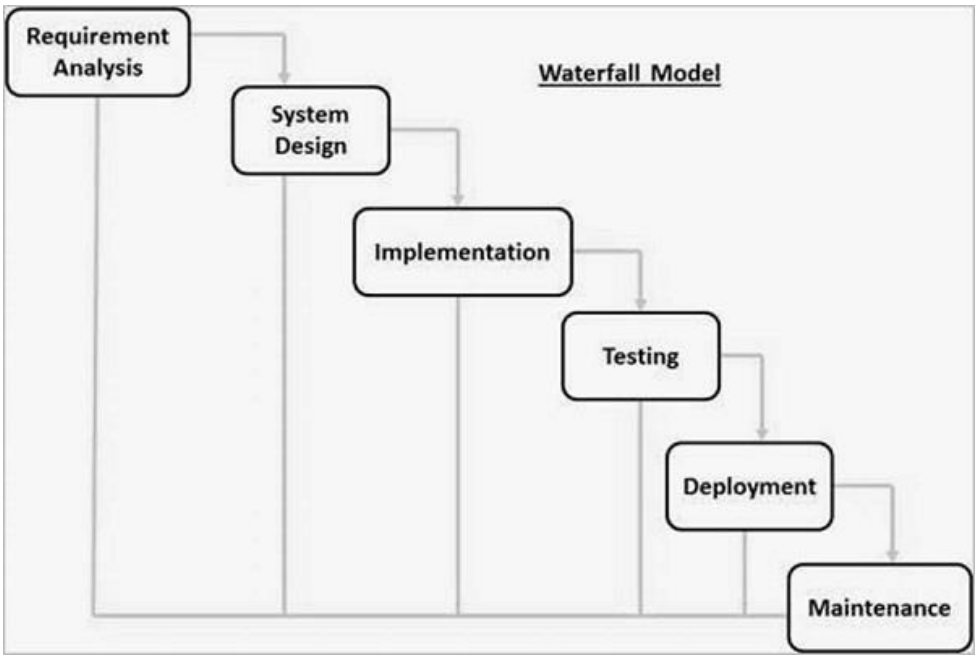
The Waterfall model is the earliest SDLC approach that was used for software development.

The software development process is depicted using the waterfall model, which follows a linear sequential flow. This implies that a phase of development can only start if the one before it is finished. The stages in this waterfall model do not cross over.

b. Model design.

The first widely used SDLC model in software engineering to assure project success was the waterfall technique. The entire software development process is split into several phases using "The Waterfall" technique. Typically, the results of one step in this waterfall model serve as the input for the subsequent phases in turn.

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 the 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.

(tutorialspoint, 2023)

c. Advantages and Disadvantages.

Advantages	***Disadvantages***
Simple and easy to understand and use	No working software is produced until late during the life cycle.
Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.	High amounts of risk and uncertainty.
Phases are processed and completed one at a time.	Not a good model for complex and object-oriented projects.
Works well for smaller projects where requirements are very well understood.	Poor model for long and ongoing projects.
Clearly defined stages.	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.
Well understood milestones.	It is difficult to measure progress within stages.

Easy to arrange tasks.	Cannot accommodate changing requirements.
Process and results are well documented.	Adjusting scope during the life cycle can end a project.

(tutorialspoint, 2023)

2. V-model.

a. Definition.

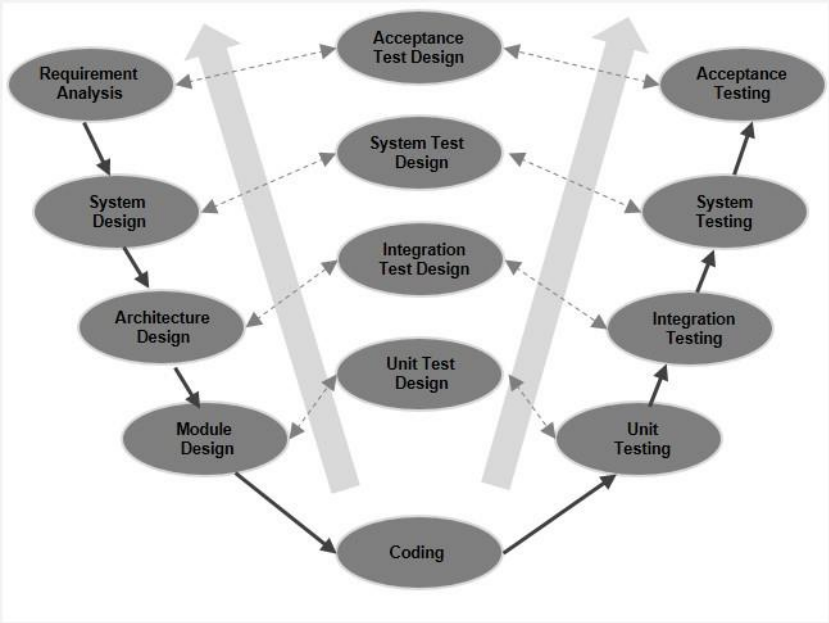
The V-model is an SDLC paradigm in which processes are executed sequentially in the form of a V-shape. Another name for it is the Verification and Validation model.

The V-Model, which is a development of the waterfall model, is predicated on the coupling of a testing phase with each appropriate development stage. As a result, a testing phase has a direct relationship with each and every phase of the development cycle. Since each step must be finished before moving on to the next, this technique requires extreme discipline.

b. Design.

According to the V-Model, the associated testing phase of the development phase is scheduled concurrently. Therefore, the 'V' represents phases of verification and the 'V' represents phases of validation. The V-Model's two sides are connected during the coding phase.

The following illustration depicts the different phases in a V-Model of the SDLC.



There are several Verification phases in the V-Model, each of these are explained in detail below:

- ***Business Requirement Analysis*** - This is the first phase in the development cycle where the product requirements are understood from the customer's perspective. This phase involves detailed communication with the customer to understand his expectations and exact

requirement. This is a very important activity and needs to be managed well, as most of the customers are not sure about what exactly they need. The acceptance test design planning is done at this stage as business requirements can be used as an input for acceptance testing.

- ***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. The system test plan is developed based on the system design. Doing this at an earlier stage leaves more time for the actual test execution later.
- ***Module Design*** - In this phase, the detailed internal design for all the system modules is specified, referred to as Low Level Design (LLD). It is important that the design is compatible with the other modules in the system architecture and the other external systems. The unit tests are an essential part of any development process and helps eliminate the maximum faults and errors at a very early stage. These unit tests can be designed at this stage based on the internal module designs.

Coding Phase:

- The actual coding of the system modules designed in the design phase is taken up in the 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 tests designed in the module design phase are executed on the code during this validation phase. 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. Most of the software and hardware compatibility issues can be uncovered during this system test execution.
- ***Acceptance Testing*** - Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment. 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.

(tutorialspoint, 2023)

c. Advantages and Disadvantages.

Advantages	*Disadvantages*
This is a highly-disciplined model and Phases are completed one at a time.	High risk and uncertainty.
Works well for smaller projects where requirements are very well understood.	Poor model for long and ongoing projects.
Simple and easy to understand and use.	Once an application is in the testing stage, it is difficult to go back and change a functionality.
Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.	No working software is produced until late during the life cycle.

(tutorialspoint, 2023)

3. Spiral.

a. Definition.

In the spiral model, the notion of iterative development is combined with the waterfall approach's methodical, managed elements. This spiral approach, which places a strong focus on risk analysis, is a hybrid of the waterfall model and the sequential linear development model, or iterative development process model. With each spiral-shaped iteration, it enables for incremental product launches or gradual product refinements.

b. 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:***

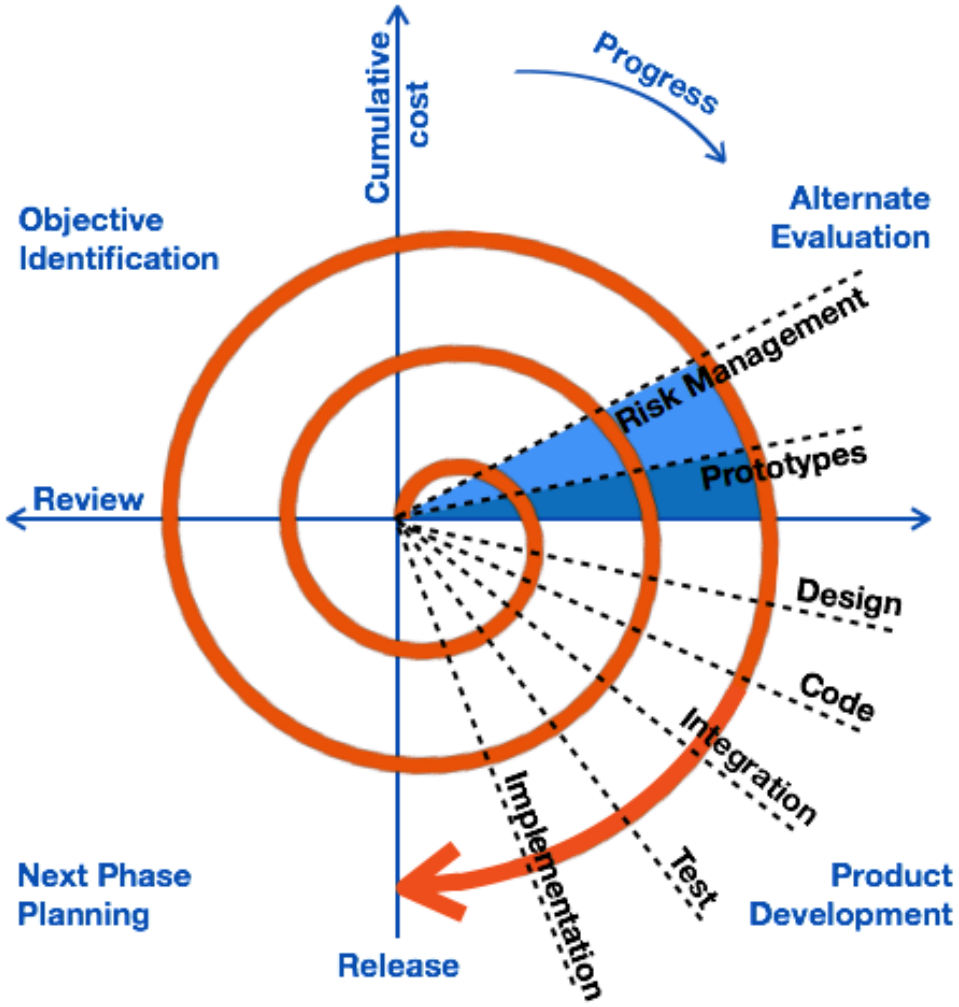
  The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

- ***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.

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



(tutorialspoint, 2023)

The Spiral Model consists of the following phases:

- ***Planning***: The first phase of the Spiral Model is the planning phase, where the scope of the project is determined and a plan is created for the next iteration of the spiral.
- ***Risk Analysis***: In the risk analysis phase, the risks associated with the project are identified and evaluated.

- ***Engineering***: In the engineering phase, the software is developed based on the requirements gathered in the previous iteration.
- ***Evaluation***: In the evaluation phase, the software is evaluated to determine if it meets the customer's requirements and if it is of high quality.

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.

c. Advantages and Disadvantages.

Advantages	*Disadvantages*
Changing requirements can be accommodated.	Management is more complex.
Allows extensive use of prototypes.	End of the project may not be known early.
Requirements can be captured more accurately.	Not suitable for small or low risk projects and could be expensive for small projects.
Users see the system early.	Process is complex
Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.	Spiral may go on indefinitely.

(tutorialspoint, 2023)

4. Scrum.

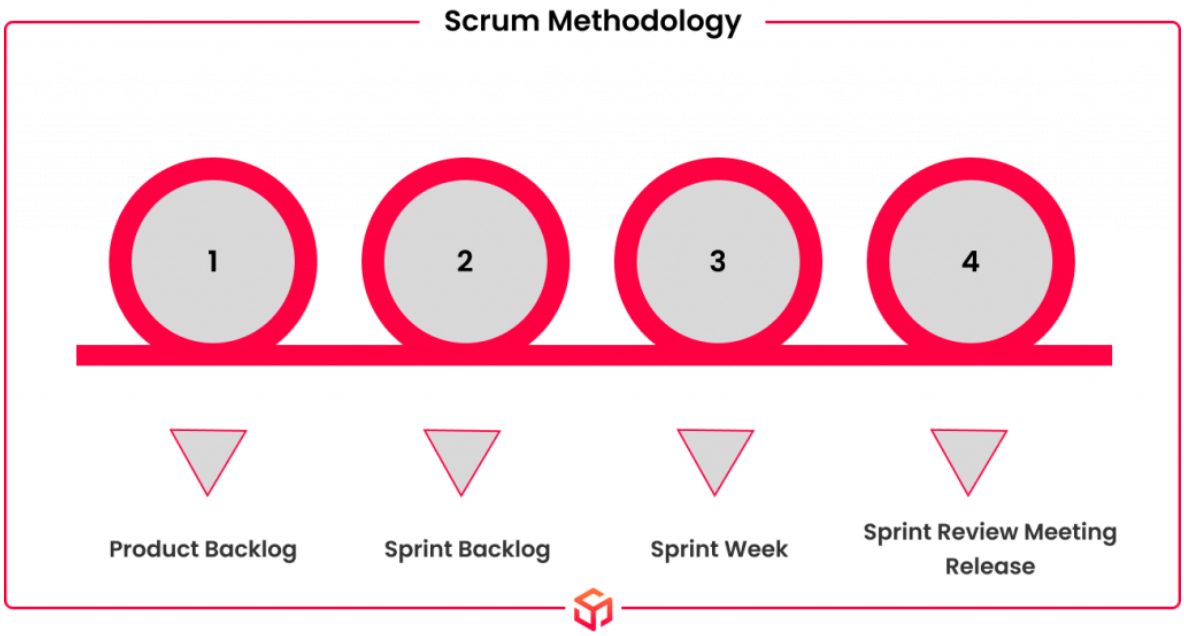
a. Definition.

Scrum is an Agile methodology that supports team collaboration in the creation of superior software solutions. It is a simple, gradual, and iterative technique with a strong emphasis on providing value to the client. The Software Development Life Cycle (SDLC) uses the Scrum model, an agile methodology, to combine Scrum concepts.

Planning, designing, coding, testing, and deployment are only a few of the processes that go into software development, which are described in detail by the SDLC process. Teams may take advantage of Agile methodology while using a structured approach to software development by incorporating Scrum within the SDLC. By following the SDLC and Scrum, Agile and structured development may be achieved in software development.

(Amuthan, 2023)

b. Design.



- ***Product Backlog***: All of the features, functions, and specifications for the software product are included in the product backlog. It is developed and prioritized by the product owner, who is in charge of making sure the team concentrates on the most important tasks first.
- ***Planning Sprint***: A sprint backlog is made up of the tasks necessary to finish the chosen items from the product backlog for which the development team has estimated the amount of work.
- ***Sprint***: The sprint is a time-boxed iteration that usually lasts two to four weeks. The team works on finishing the tasks on the sprint backlog during the sprint. Daily stand-up meetings are held by the team to discuss advancement and any problems.
- ***Sprint Review***: The team meets for a sprint review meeting to present the finished product to stakeholders through prototyping. The team gets comments, which are added to the product backlog for the subsequent sprint. Typically, they use programs like Figma to illustrate it.
- ***Sprint retrospective***: A sprint retrospective is a meeting when the team reviews the sprint's progress, analyzes areas for improvement, and decides what steps to take moving forward.
- ***Repeat***: Every sprint until the product is finished, the team repeats the aforementioned procedures.

(Amuthan, 2023)

c. Advantages and Disadvantages.

× **Advantages:**

- ***Adaptability and flexibility***: Scrum is a fluid and adaptable methodology in and of itself, making it appropriate for a broad range of settings and circumstances that call for a flexible approach, such as when it may be challenging to define unambiguous requirements.

- ***Creativity and innovation***: Scrum prioritizes innovation and creativity. Scrum teams foster novel and creative solutions because they collaborate closely and evaluate ideas from all members.
- ***Time to market***: Scrum frequently speeds up delivery compared to other methods, shortening the time it takes for new goods or services to hit the market.
- ***Lower costs***: Compared to other project management methodologies, the Scrum methodology often involves less paperwork and oversight, therefore it may be less expensive for a firm.
- ***Increased transparency often leads to higher quality work***: By fostering openness between teams, as well as between teams, internal stakeholders, and clients, the Scrum approach helps to guarantee that team members are attentive and productive. No matter how little, every participant may see any modification made to any aspect of a project. All parties become more trustworthy as a result. At the same time, because every team member assumes complete accountability for their task, this frequently fosters a productive environment and produces high-quality products.

× **Disadvantages:**

- ***You need experienced team members***: All members of your team must be capable of doing their individual jobs successfully since Scrum requires lengthy periods of intensive labor. To be able to achieve this, you typically need to have significant experience. Each member of the team has to work well and offer insightful criticism of the procedure and outcomes.
- ***It requires lots of training***: Your team needs training if they lack experience. This takes time as each team member must understand the advantages, phenomena, thought processes, and practices. Expect this to take several months if you start from scratch before your team is fully operational.
- ***It only really works with small teams***: Teams of little more than ten work best for scum in most cases. Larger firms can find it difficult to successfully divide their workforce into small, independent teams since they are not used to doing so, which would hold down their adoption of Scrum.
- ***It can be difficult to scale***: The implementation of Scrum on a larger scale typically necessitates intensive training and coordination, which is related to the aforementioned problem. Due to the complexity of properly implementing and managing scrum on larger projects with larger teams, this seldom happens.
- ***It may require major transformation within an organization***: Adopting the Scrum framework may necessitate organizational change to support the practice, which is related

to the preceding drawback. Different departments may need to work differently on some process steps, which might result in a substantial organizational shift.

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

Usefulness in Developing System	Waterfall	Parallel	V-model	Iterative	System prototyping	Throwaway prototyping	Agile Development
clear structure	Excellent	Poor	Good	Poor	Poor	Poor	Poor
determines the end goal early	Excellent	Poor	Poor	Poor	Poor	Poor	Poor
keeps a project to a specific timescale.	Good	Poor	Poor	Poor	Poor	Poor	Good
the phases of the waterfall model are predictable and don't overlap	Excellent	Poor	Good	Poor	Poor	Poor	Good
with unclear user requirements	Poor	Poor	Poor	Good	Excellent	Excellent	Excellent
with unfamiliar technology	Poor	Poor	Poor	Good	Poor	Excellent	Poor

The use of waterfall model in large software development project brings a lot of value. Here are a few reasons the waterfall model is used:

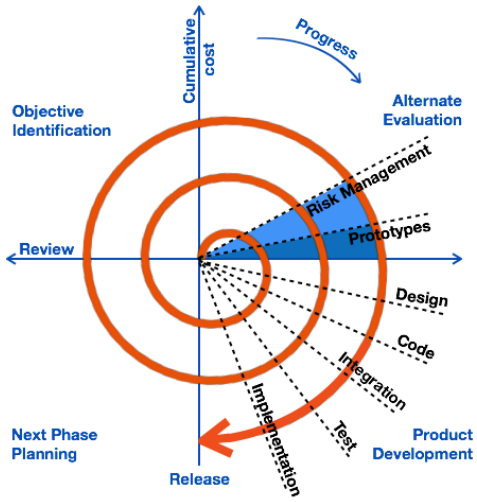
- ***Easy to understand, easy to use***: Like a map or a setup system, it's easy to track project progress. Know what stage the project is at in the project development flowchart. Easy to manage, smooth software development.
- ***Easy-to-understand work structure***: The waterfall model focuses on the exact set of stages (Requirement -> Design -> Deployment -> Testing -> Installation -> Maintenance), making the work structure easy to understand for inexperienced staffs. Team projects must complete this phase before moving on to other phases, so if risks are identified can be addressed immediately.
- ***Costs can be predicted***: The waterfall model requires a detailed outline of each stage, the more accurate the cost estimate will be, incurring additional costs during the project implementation. Project teams estimate time, labor and costs at each stage with higher specifications.

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

1. Risk management process.

Evaluation and Risk Analysis.

The phase of risk analysis includes the identification, estimation, and monitoring of technical feasibility as well as the management of risks such project delays or incurred expenses. At the conclusion of the first loop, following build testing, the client assesses the application and provides unbiased comments.



Activities of each phase. (tutorialspoint, 2023)

Based on customer reviews in the first iteration, software development enters the next iteration and then follows a linear approach to implementing client-recommended feedback. Iterative process occurs throughout the life of the software.

2. Risk management Matrix.

No.	***Risks***	***Describe***	***Impact***	***Solution***
1.	Technology	Using outdated technology, employees have not updated and learned new technology.	The software uses outdated technologies, so it is difficult to access new technologies that reduce the performance of the software, it is difficult to meet the increasing problems of customers, and may even incur additional costs during development.	Access to new technology, train employees to use and learn new technology. Apply new technologies during project development. Send employees to research and learn more about the latest technologies.
2.	Costs incurred	In the process of project implementation, it is difficult to avoid arising (errors arising during project implementation, project extending beyond the prescribed time limit...) which generate	Causing budget deficits, delaying project completion. Impact on the profitability of the project.	Detailed planning for the project, reasonable cost management, cost recognition during project implementation.

		additional costs compared to the initial statistics.		
3.	Scope creep	Regular feedback from customers, stakeholders, or product owners will often lead to a change in the scope of the project.	Causing serious risks to project. When scope changes, it significantly affects the developer's ability to stick to the original timeline of the project.	Regularly check customer feedback to easily manage and change the most appropriate scope. Prioritize the implementation of important functions, avoid performing many functions at the same time.
4.	Manpower shortage	Sometimes a stakeholder or development team member has to leave the project unexpectedly. This can create project risks, especially if project knowledge is not well documented.	Insufficient human resources lead to effective project completion, slow progress, inexperienced new employees find it difficult to adapt to the project environment, and difficult to complete projects.	Regularly monitor the project team's schedule, maintain updated documents related to the project. complete Provide documentation for those new to the project.

III. Feasibility study.

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

A project feasibility study analyzes several factors, including market demand, technological problems, financial projections, and organizational capabilities. It provides project managers with project insights that can aid in improved decision making. In addition, a feasibility study provides project stakeholders with a thorough understanding of any potential risks, advantages, and costs.

A feasibility study's goal is to decide if a project should be continued, amended, or abandoned. Knowing a project's chances of success makes good economic sense for a company. A project feasibility study analyzes a number of factors, including market demand, technological issues, projected financial results, and organizational capability. It offers project managers useful project insights that can aid in improved decision-making. A feasibility study also gives stakeholders a thorough idea of the project's possible risks, advantages, and costs a project before investing money in it. A feasibility study's goal is to decide if a project is worthwhile, whether it needs to be adjusted, or whether it should be abandoned. An organization should ascertain a project's likelihood of success before allocating resources to it for business reasons.

Some reasons why it is advisable to conduct a feasibility study before starting a project:

- The study helps identify pathways and risks early in the process, allowing for proactive problem solving.

- It determines project viability by focusing on technical, operational and financial feasibility.
- It helps stakeholders understand the potential impact of the project and helps gain their support.
- Project managers have more information to make better decisions and avoid wasting resources.
- It ensures the project aligns with the organization's goals and objectives.
- By detecting hidden wake-ups early in the process, feasibility reports can help reduce the risk of unexpected costs or delays in the future.
- Research acts as a roadmap for the project.

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

a. Discuss how the three feasibility criteria are applied to the project.

A feasibility study in software engineering is a study to determine the viability of a given project or system. One of the crucial four phases in the software project management process is the feasibility study. As the name implies, a feasibility study is an investigation of the viability of a project or a measurement of a software product's potential value to an organization from a practical standpoint. A feasibility study is conducted for a variety of reasons, including to determine whether a software project will be appropriate in terms of its development, implementation, and value to the company.

Technical feasibility:

Technical feasibility is the process of formally determining whether it is technically feasible to create a good or service. It is crucial to plan and organize every aspect of the business before introducing a new service or starting a client project. Analyzing the process, including the tools, technology, materials, labor, and logistics, may assist establish the technical feasibility of a given strategy.

The questions that test the technical feasibility of a system:

- 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?
    - Some organizations like to use state-of-the-art technology … but most prefer to use mature and proven technology.

  - A mature technology has a larger customer base for obtaining advice concerning problems and improvements.
- 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?

(FLM, 2023)

Operational feasibility:

It is a measure of how well the proposed solution will attempt to solve the stated problem and take advantage of the loopholes discovered during the scoping phase. It also decides how it will fulfill each need found during the requirements analysis phase.

Operational feasibility assesses whether the system can be used after the project is established and implemented and depends on the human resources available for the project. It checks the organization's readiness to support the proposed system. Operational feasibility, as opposed to technical and economic feasibility, is difficult to assess.

However, you must understand management's commitment to the proposed initiative to assess operational feasibility. Operational feasibility will probably be accepted and put into use if the management has initiated it.

The questions that test the technical feasibility of a system:

- Does the current operational mode provide enough response and throughput time?
- Is there timely and accurate information to managers and end-users?
- Does the current mode of operation offer cost-effective information to the business?
- Will there be an increased benefit or a cost reduction?
- Does the current system provide an effective process to safeguard against fraud and breach of data?
- Will the system be used once it is developed?
- What are the manpower problems the system will face?
- Will there be labor objections?
- What are the likely organizational policies and conflicts?
- What are the governmental regulations that might affect the system?

(FLM, 2023)

Economic feasibility:

Economic feasibility is a sort of cost-benefit analysis that assesses a project's viability and sense of logic. It entails evaluating and examining the project's assets, possibilities, risks, and weaknesses. The goal of economic feasibility is to aid in decision-making and assess the likelihood that the project will result in financial advantages.

The questions that test the technical feasibility of a system:

- Can the bottom line be quantified yet?
    - Very early in the project … a judgement of whether solving the problem is worthwhile.
    - Once specific requirements and solutions have been identified … the costs and benefits of each alternative can be calculated.
- Cost-benefit analysis:
    - Purpose - answer questions such as:
        - Is the project justified (I.e. will 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?
    - Examples of things to consider:
        - Hardware/software selection.
        - Selection among alternative financing arrangements (rent/lease/purchase).
    - Difficulties:
        - benefits and costs can both be intangible, hidden and/or hard to estimate.
        - ranking multi-criteria alternatives.

(FLM, 2023)

3. Discuss alternative technical solutions using the alternative matrix.

The Alternative Matrix is a tool used in the Feasibility Study phase of the SDLC to evaluate and compare different technical solutions for a project. It is also known as the Weighted Decision Matrix or Criteria Matrix.

The Alternative Matrix is a table that compares several technical project options based on a set of standards. The criteria, which may include elements like cost, response time, training time, simplicity of use, a strong team, and team experience, are often established by the project's stakeholders. Based on the significance of each criterion to the project, a weight is allocated to it.

The technical solutions are then evaluated based on each criterion, and a score is assigned to each solution. The scores are then multiplied by the weight of each criterion and added together to get a total score for each solution. The solution with the highest total score is considered the best option.

		Alternative 1		Alternative 2	
Criteria	**Wt**	**Score**	**Weight**	**Score**	**Weight**
Cost	2	4	8	1	2
Maintenance cost	3	1	3	4	12
Response time	2	2	4	3	6
Ease of use	5	3	15	2	10
Team experience	3	2	6	1	3
Training time	7	2	14	1	7
Total	22	14	50	12	40

IV. References.

Amuthan, T., 2023. *Scrum Model In SDLC : All You Need To Know.* [Online]
Available at: https://www.softsuave.com/blog/scrum-model-in-sdlc/#What_is_Scrum_Model_in_SDLC
[Accessed 30 Aug 2023].

F., 2023. [Online]
Available at: https://flm.greenwich.edu.vn/gui/role/student/SyllabusDetails?sylID=2538

t., 2023. *SDLC - Spiral Model.* [Online]
Available at: https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm
[Accessed 17 Jul 2023].

t., 2023. *SDLC - V-Model.* [Online]
Available at: https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm
[Accessed 17 Jul 2023].

t., 2023. *SDLC - Waterfall Model.* [Online]
Available at:
https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm?gidzl=1WKmGBNolse06J93kUMxQWyNFqQWuh9iHqOvG_UovsiJ4s92fEJlE4GNPXFyvxazJK1Z5Zc1nRmVjVArPm
[Accessed 17 Jul 2023].