



SOFTWARE DEVELOPMENT LIFE CYCLE

NAME: VO NGOC MAI

LEARNER'S NAME: NGUYEN VUONG THAI BAO

LEARNER'S ID: GCS190574

CLASS: GCS0805





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:

- 1. Search for music in our digital music archive.
- 2. Listen to music samples.
- 3. Purchase individual downloads at a fixed fee per download.
- 4. Establish a customer subscription account permitting unlimited downloads for a monthly fee.
- 5. 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:

- 6. \$757,500 in sales from individual music downloads
- 7. \$950,000 in sales from customer subscriptions
- 8. \$205,000 in additional in-store or website CD sales
- 9. \$153,000 in sales from music download gift cards

Special Issues or Constraints:

- 10. 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.
- 11. 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.
- 12. 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.





Learning Outcomes and Assessment Criteria							
Pass	Merit	Distinction					
LO1 Describe different software							
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							
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 conten

	RIBE TWO ITERATIVE AND TWO SEQUENTIAL SOFTWARE LIFECYCLE S	7
	LC - Waterfall:	
1.1.	Waterfall Model - Design	
1.2.	Waterfall Model – Application:	
1.3.	Waterfall Model – Advantages:	
1.4.	Waterfall Model – Disadvantages	
	LC - Prototyping Model:	
2.1.	What is prototyping model:	
2.2.	Steps of the prototyping model:	
2.3.	Types of prototype models:	
2.4.	Advantages of the prototyping model	
2.5.	Disadvantages of the prototyping model	
3. SD	LC - Agile Model:	
3.1.	What is Agile?:	
3.2.	The advantages of the Agile Model are as follows:	
3.3.	The disadvantages of the Agile Model are as follows :	14
4. SD	LC - Spiral Model:	15
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 –	
5. Mo	odel for suitable for the project:	
(P2) EXP	LAIN HOW RISK IS MANAGED IN THE SPIRAL LIFECYCLE MODEL	18
1. Ma	nnagement Risk:	18
1.1.	A risk management process involves:	18
1.2.	The types of risk your business faces:	18
1.3.	Strategic and compliance risks:	19
1.4.	Financial and operational risks	19
1.5.	Use preventative measures for business continuity:	19
1.6.	How to manage risks:	20
2. Son	me risks of the Tune Source project:	20
1.1.	Scope Risks:	20
1.2.	Cost Risks:	20





1.3.	Time Risks:	21
1.4.	Communication Risks:	21
P3 EXPI	LAIN THE PURPOSE OF A FEASIBILITY REPORT	21
1. De	efinition Of Feasibility Report/Study. :	21
2. Ke	ey components of a feasibility report	21
3. Pu	urpose Of A Feasibility Report	21
P4 DESC	CRIBE HOW TECHNICAL SOLUTIONS CAN BE COMPARED	22
1. Te	chnical Feasibility:	22
1.1.	Define:	22
1.2.	We can built it? Yes. We Can built it	23
2. Ec	conomic Feasibility:	23
2.1.	Define of Economic Feasibility:	23
2.2.	Should we built it?	23
3. O	ganizational Feasibility:	24
<i>3.1.</i>	Organizational Feasibility: If we build it, will they come? Why?	24
4. Di	scuss alternative technical solutions using the alternative matrix	24
4.1.	.NET technology:	24
4.2.	Java technology:	25
4.3.	Alternative Matrix:	
Defense		20





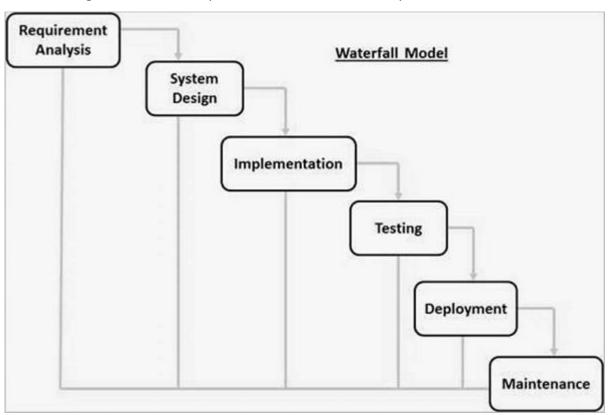
P1 DESCRIBE TWO ITERATIVE AND TWO SEQUENTIAL SOFTWARE LIFECYCLE MODELS.

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.



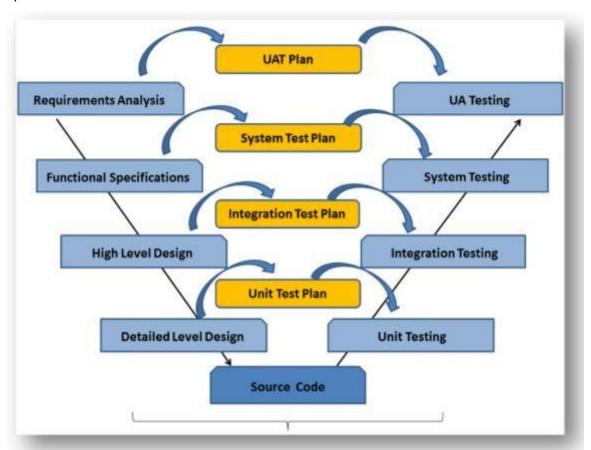


• Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

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:





- Customers get a say in the product early on, increasing customer satisfaction.
- Missing functionality and errors are detected easily.
- Prototypes can be reused in future, more complicated projects.
- It emphasizes team communication and flexible design practices.
- Users have a better understanding of how the product works.
- Quicker customer feedback provides a better idea of customer needs.

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.

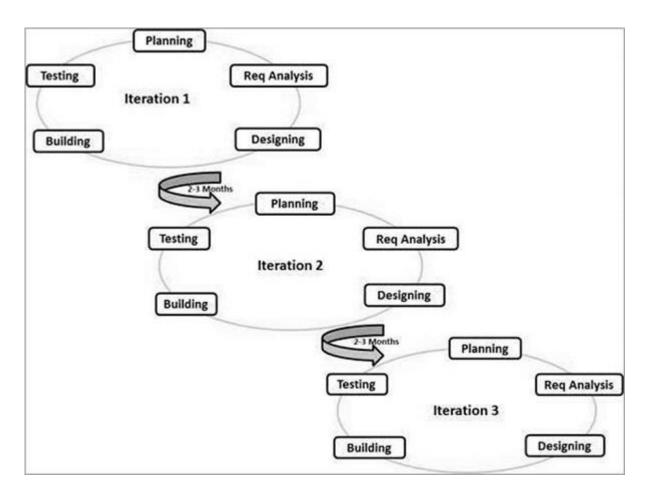
At the end of the iteration, a working product is displayed to the customer and important stakeholders.

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

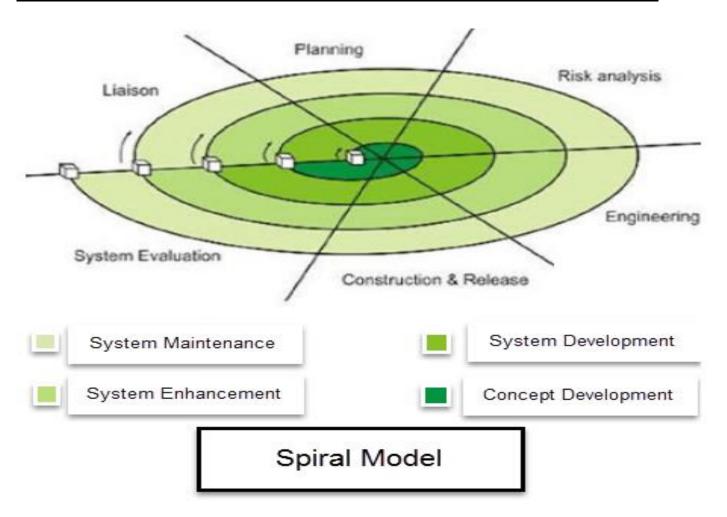
The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

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. Management Risk:

1.1. A risk management process involves:

- Businesses face many risk, therefore risk management should be a central part of any business' strategic management. Risk management helps you to identify and address the risks facing your business and in doing so increase the likelihood of successfully achieving your businesses objectives.
- ♣ A risk management process involves:
 - Methodically identifying the risks surrounding your business activities
 - Assessing the likelihood of an event occurring
 - Understanding how to respond to these events
 - Putting systems in place to deal with the consequences
 - Monitoring the effectiveness of your risk management approaches and controls
- ♣ As a result, the process of risk management:
 - o Improves decision-making, planning and prioritisation
 - o Helps you allocate capital and resources more efficiently
 - Allows you to anticipate what may go wrong, minimising the amount of firefighting you have to do or, in a worst-case scenario, preventing a disaster or serious financial loss
 - Significantly improves the probability that you will deliver your business plan on time and to budget
- Risk management becomes even more important if your business decides to try something new, for example launch a new product or enter new markets. Competitors following you into these markets, or breakthroughs in technology which make your product redundant, are two risks you may want to consider in cases such as these.

1.2. The types of risk your business faces:

- ♣ The main categories of risk to consider are:
 - o Strategic, for example a competitor coming on to the market
 - o Compliance, for example the introduction of new health and safety legislation
 - Financial, for example non-payment by a customer or increased interest charges on a business loan
 - Operational, for example the breakdown or theft of key equipment
- ♣ These categories are not rigid and some parts of your business may fall into more than one category. The risks attached to data protection, for example, could be considered when reviewing your operations or your business' compliance.
- Other risks include:
 - o Environmental risks, including natural disasters
 - Employee risk management, such as maintaining sufficient staff numbers and cover, employee safety and up-to-date skills





- o Political and economic instability in any foreign markets you export goods to
- Health and safety risks

1.3. Strategic and compliance risks:

- Strategic risks are those risks associated with operating in a particular industry.
- They include risks arising from:
 - Merger and acquisition activity
 - o Changes among customers or in demand
 - Industry changes
 - Research and development

1.4. Financial and operational risks

- Financial risks are associated with the financial structure of your business, the transactions your business makes and the financial systems you already have in place.
- ♣ Identifying financial risk involves examining your daily financial operations, especially cash flow. If your business is too dependent on a single customer and they are unable to pay you, this could have serious implications for your business' viability.
 - The way you extend credit to new customers
 - Who owes you money
 - o The steps you can take to recover it
 - o Insurance that can cover large or doubtful debts
- Financial risk should take into account external factors such as interest rates and foreign exchange rates.
- ♣ Rate changes will affect your debt repayments and the competitiveness of your goods and services compared with those produced abroad.
- Operational risks
- Operational risks are associated with your business' operational and administrative procedures. These include:
 - Recruitment
 - Supply chain
 - Accounting controls
 - IT systems
 - Regulations
- Board composition

1.5. Use preventative measures for business continuity:

- ♣ Risk management involves putting processes, methods and tools in place to deal with the consequences of events you have identified as significant threats for your business. This could be something as simple as setting aside financial reserves to ease cash flow problems if they arise or ensuring effective computer backup and IT support procedures for dealing with a systems failure.
- ♣ Programs which deal with threats identified during risk assessment are often referred to as business continuity plans. These set out what you should do if a certain event happens, for example, if a fire destroys your office. You can't avoid all risk, but business continuity plans can minimize the disruption to your business.
- ♣ Risk assessments will change as your business grows or as a result of internal or external changes. This means that the processes you have put in place to manage your business risks should be





regularly reviewed. Such reviews will identify improvements to the processes and equally they can indicate when a process is no longer necessary.

1.6. How to manage risks:

- ♣ There are four ways of dealing with, or managing, each risk that you have identified. You can:
 - Accept it
 - Transfer it
 - Reduce it
 - Eliminate it
- For example, you may decide to accept a risk because the cost of eliminating it completely is too high. You might decide to transfer the risk, which is typically done with insurance. Or you may be able to reduce the risk by introducing new safety measures or eliminate it completely by changing P a g e | 15 the way you produce your product. When you have evaluated and agreed on the actions and procedures to reduce the risk, these measures need to be put in place.
- ♣ Risk management is not a one-off exercise. Continuous monitoring and reviewing are crucial for the success of your risk management approach. Such monitoring ensures that risks have been correctly identified and assessed and appropriate controls put in place. It is also a way to learn from experience and make improvements to your risk management approach.
- ♣ All of this can be formalized in a risk management policy, setting out your business' approach to and appetite for risk and its approach to risk management. Risk management will be even more effective if you clearly assign responsibility for it to chosen employees. It is also a good idea to get commitment to risk management at the board level.
- ♣ Good risk management can improve the quality and returns of your business

2. Some risks of the Tune Source project:

1.1. Scope Risks:

Scope risks are tasks that endanger project objectives, deliverables, or timeline. This is the most well-known project risk of entrepreneurs. Even though there are various steps to be taken to avoid the odds of this happening.

Mitigation plans: Create consistent concrete goals and task dates for each phase and team member working on a given project. This will ensure your team doesn't try and add additional tasks or take up any extra free time on tasks for any given project.

1.2. Cost Risks:

This type of risk is the most common as it shows the biggest threat to businesses and their financials. Staying within budget is crucial for business owners to make a sufficient profit of their client's projects while still making them happy.

Mitigation plans: To avoid going into the red, have your team stagger their work by moving non-essential tasks to different weeks, months, or years to best suit your





businesses project focus. Along with that, stay in touch with your expenses by tracking them on an app or spreadsheet to double-check your project's finances are still in the green.

1.3. Time Risks:

If you and your team think there is enough time to complete your project comfortably, or even with a touch of time to spare, you most likely aren't scheduling your tasks effectively.

Mitigation plans: Have you or your management team take on a couple more projects or simply wrap up projects sooner rather than later if you think there is too much time to put towards one given project. That way, if something were to take a turn for the worse on a deadline, you were allocated enough time to take control and get it back on track.

1.4. Communication Risks:

In a world filled with technology, sometimes effective communication gets put to the side. Make sure your team is communicating any delays or shifts in any given project to ensure the reallocation or changes are being made to best suit your deliverable. Along with that, foster a positive team working environme

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. Technical Feasibility:

1.1. Define:

- 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





1.2. We can built it? Yes. We Can built it ...

- ♣ This Tuin source project is similar to 2 other projects that we have developed (Analysis Systems and Design, Inc.) completed for 2 other small businesses. Our team will successfully Tuin source to provide the functionality to the users.
- Familiarity with Technology
 - The application will be built in Microsoft Access, which SAD analysts/programmers have extensive experience with.
 - The employees at VW have virtually no experience using MS Access or an automated rental processing system but all are comfortable with using PCs.
 Training should be sufficient to overcome their inexperience.
- ♣ This is not a big project, so with the experience we have, we believe that this project will be completed well.

2. Economic Feasibility:

2.1. Define of 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

2.2. Should we built it?

Development cost:

- 5 networked PCs and sofrware \$7500
- \$11000 for us, Inc development and implement
- \$2000 for user manuals and training

Annual Operating costs

- \$500 technical support fee to us once system is installed
- \$750 for hardware/software replacement and upgrades.

Annual Benefit:

- 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

Intangible Costs and Benefits

- Save time for customers
- Increase customer satisfaction with Tuin source.
- Free some of employees' time to provide more assistance tp customers.

3. Organizational 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. Two key aspects to consider include management ability and resource sufficiency.

3.1. Organizational Feasibility: If we build it, will they come? Why?

Yes they will come

This project is designed based on tuning strategy's development source. So it is very suitable for long-term development with other projects in the future.

- This project is done through a signed agreement between the software development company and the client. Elements surrounding the project include the developer and steakhoder.
- In conclusion, the Tuin Source project is a small project, but it will bring many benefits to the product owners of this project as the Tuin Source project is feasible in both Technical, Economic and Organizational terms..

4. Discuss alternative technical solutions using the alternative matrix

4.1. .NET technology:

- NET is a software framework that is designed and developed by Microsoft. The first version of the .Net framework was 1.0 which came in the year 2002. In easy words, it is a virtual machine for compiling and executing programs written in different languages like C#, VB.Net, etc.
- It is used to develop Form-based applications, Web-based applications, and Web services. There is a
 variety of programming languages available on the .Net platform, VB.Net and C# being the most
 common ones. It is used to build applications for Windows, phone, web, etc. It provides a lot of
 functionalities and also supports industry standards.
- .NET Framework supports more than 60 programming languages in which 11 programming languages
 are designed and developed by Microsoft. The remaining Non-Microsoft Languages which are
 supported by .NET Framework but not designed and developed by Microsoft.





4.1.1. Advantages of .NET:

- **Object Oriented** Everything that you see in the .NET framework is an object. It is the same for what you write within the framework. This means that you get a powerful tool to not just access but also control your apps. This also makes it simpler for you to respond to recurring events.
- Caching The caching system that .NET includes is extremely robust and easy-to-use.
- *Easy Maintenance* Pages, with .NET, are extremely simple to write and maintain. This is because the source code and HTML are both together. In addition to that, the source code executes on the server. What does this mean? This makes your web pages more powerful and flexible.
- **Simplicity** Performing common tasks with .NET is extremely simple and straight forward. Submission of forms is a breeze and so is site configuration, deployment, and client authentication.
- **Feature-Rich** There are a range of features that can be explored by the developers in order to create powerful apps. Consider the case of its rich toolbox as also the designer in the visual studio. They let you access such features as automatic deployment, WYSIWYG editing, and drag-anddrop controls.
- **Consistency** The management and monitoring of all the processes is performed by the framework.

 If one of the processes is dead, a new process can be created just as easily. This lets your app be consistently available for handling requests.
- **Monitoring** Finally, .NET also stands for its automatic monitoring. It will promptly notice any problems like infinite loops, memory leaks, etc. Not just this, it will also destroy these activities automatically and restart itself

4.2. Java technology:

- → Java is a set of computer software and specifications developed by James Gosling at Sun Microsystems, which was later acquired by the Oracle Corporation, that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers. Java applets, which are less common than standalone Java applications, were commonly run in secure, sandboxed environments to provide many features of native applications through being embedded in HTML pages.
- ♣ Writing in the Java programming language is the primary way to produce code that will be deployed as byte code in a Java virtual machine (JVM); byte code compilers are also available for other languages, including Ada, JavaScript, Python, and Ruby. In addition, several languages have been designed to run natively on the JVM, including Clojure,



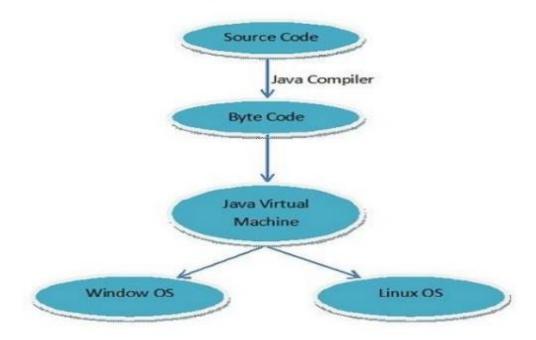


Groovy, and Scala. Java syntax borrows heavily from C and C++, but object-oriented features are modeled after Smalltalk and ObjectiveC.[12] Java eschews certain low-level constructs such as pointers and has a very simple memory model where objects are allocated on the heap (while some implementations e.g. all currently supported by Oracle, may use escape analysis optimization to allocate on the stack instead) and all P a g e | 33 variables of object types are references. Memory management is handled through integrated automatic garbage collection performed by the JVM.

4.2.1. Java features:

Simple, easy and familiar:

- Java is easy to learn and familiar because java syntax is just like c++.
- It is simple because:
 - It does not use header files.
 - Eliminated the use of pointer and operator overloading.
- **Platform Independent:** Write once, run anywhere (WORA)



Object-Oriented:

- o Java is Object oriented throughout language- that mean no coding outside of
- Class definitions, including main().

Robust

- Robust means inbuilt capabilities to handle errors/exceptions.





- Java is robust because of following:
 - Built-in Exception handling.
 - Strong type checking i.e. all data must be declared an explicit type.
 - Local variables must be initialized.
 - Automatic garbage collection.
 - First checks the reliability of the code before Execution etc.
- **Secure:** Java is secure because it provides:
 - o Access restrictions with the help of access modifiers (public, private etc).
 - Byte codes verification checks classes after loading.
 - Class loader confines objects to unique namespaces.
 - Security manager determines what resources a class can access such as reading and writing to the local disk.
- ♣ Distributed: Java provides the network facility. i.e. programs can be access remotely from any machine on the network rather than writing program on the local machine. HTTP and FTP protocols are developed in java
- **♣ Compiled and interpreted:** Java code is translated into byte code after compilation and the byte code is interpreted by JVM (Java Virtual Machine). This two steps process allows for extensive code checking and also increase security.
- ♣ Portable: Means able to be easily carried or moved. Write once, run anywhere (WORA) feature makes it portable.
- ♣ Architecture-Neutral: Java code is translated into byte code after compilation which is independent of any computer architecture, it needs only JVM (Java Virtual Machine) to execute.
- **High performance:** JVM can execute byte codes (highly optimized) very fast with the help of Just in time (JIT) compilation technique.
- **Re-usability of code:** Java provides the code reusability With the Help of Inheritance.
- **Multithreading:** Java provides multitasking facility with the help of lightweight processes called threads
- **Dynamic:** Java have the capability of linking dynamic new classes, methods and objects.

4.3. Alternative Matrix:





		.NET TECHNOLOGY		JAVA TECHNOLOGY		OPEN SOURCE TECHNOLOGY	
Criteria	Weighting Factor	Score	Weighting Factor	Score	Weighting Factor	Score	Weighted Score
Cost	5	1	5	2	10	2	10
Development	3	3	9	4	12	2	6
Maintenance Cost	3	1	3	1	3	2	6
Integration	1	3	4	2	2	2	2
Total Score		8	21	8	27	8	24

- **♣** Based on the above report and the "Substitution Matrix" table assesses the project elements and problems. Most likely will choose Java technology to implement the project.
- ♣ Through the assessment on the report and the careful calculation of the risks occurring is very low so according to our company, this is a viable and doable project.





Reference

- SDLC Waterfall Model [Online] Tutorialspoint. Available at: https://www.tutorialspoint.com/sdlc/sdlc waterfall model.htm#:~:text=The%20Waterfall%20mod el%20is%20the,the%20previous%20phase%20is%20complete. [Accessed November 12.2020].
- 3. SDLC Agile Model [Online] *Tutorialspoint*. Available at: https://www.tutorialspoint.com/sdlc/sdlc agile model.htm#:~:text=Agile%20SDLC%20model%20is %20a,builds%20are%20provided%20in%20iterations. [Accessed November 22.2020].
- 4. SDLC Spiral Model [Online] *Tutorialspoint*. Available at: https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm
- 5. Feasibility study [Online] wikipedia . Available at https://en.wikipedia.org/wiki/Feasibility study#Legal feasibility [Accessed November 19.2020].
- 6. **c-sharpcorner.com, 2020. What is .NET Technology? [online].** Available: https://www.csharpcorner.com/blogs/what-is-net-technology [Accessed November 19.2020].
- 7. Wikipedia.org, 2020. Java (software platform) [online]. Available: https://en.wikipedia.org/wiki/Java (software platform) [Accessed November 19.2020].
- 8. Javatutorialpoint.com. Java features[online]. Available: <a href="https://www.javatutorialpoint.com/2015/03/features-ofjava.html#:~:text=Java%20features%3A%201%20Simple%2C%20easy%20and%20familiar%3A%20Java,codin20outside%20of%20class%20definitions%2C%20including%20main%20%28%29[Accessed November 19.2020].
- 9. **Wikipedia.org, 2020. Open source [online].** Available: https://en.wikipedia.org/wiki/Open source [Accessed November 19.2020].
- Thebalancecareers.com, 2019. How Open-Source Software Works [online]. Available: https://www.thebalancecareers.com/what-is-open-source-software-2071941 [Accessed November 19.2020].