

Software developoment life cycle

Final Assignment



January 30, 2023

HTU

**Abdallah Daradkeh**

**ID:211104446**

Contents

[Part1 2](#_Toc127312110)

[SDLC Models 2](#_Toc127312111)

[How risk is being manage in SDLC 9](#_Toc127312112)

[discuss using example(s) a particular lifecycle model for a specific development environment: 11](#_Toc127312113)

[Benefits of Waterfall model in large software development project 12](#_Toc127312114)

[Discussing the suitability of software behaviour design techniques using examples. 12](#_Toc127312115)

[Analayze range of tools(software behavior design techniques) and too techniques: 13](#_Toc127312116)

[Two techniques: 13](#_Toc127312117)

[Two tools: 14](#_Toc127312118)

[Differentiate between Finite FSM and extended FSM 15](#_Toc127312119)

[Data-driven improve reliability and effectiveness of software: 16](#_Toc127312120)

[Compared two technical solutions and how 17](#_Toc127312121)

[Compare two technical solutions 17](#_Toc127312122)

[Explain feasibility report purpose. 23](#_Toc127312123)

[Discuss the components of Feasibility report 23](#_Toc127312124)

[Assessing the impact of criteria feasibility on investigation of the project 24](#_Toc127312125)

[Part 2: 25](#_Toc127312126)

[Feasibility study of the online food service: 25](#_Toc127312127)

[Requirements: 25](#_Toc127312128)

[Non functional requirements: 26](#_Toc127312129)

[SRS document: 26](#_Toc127312130)

[Introduction: 26](#_Toc127312131)

[ The purpose 26](#_Toc127312132)

[ The scope 26](#_Toc127312133)

[ Overview 27](#_Toc127312134)

[Requirements: 27](#_Toc127312135)

[Non functional requirements: 27](#_Toc127312136)

[SDS Document 28](#_Toc127312137)

[Design overview 28](#_Toc127312138)

[The software major functionalities: (high level overeview of the design of software). 28](#_Toc127312139)

[Detailed Design 28](#_Toc127312140)

[User interface Design: 30](#_Toc127312141)

[Data design: 33](#_Toc127312142)

[Test plan: 35](#_Toc127312143)

[Logout 35](#_Toc127312144)

[Login 35](#_Toc127312145)

[Sign up 37](#_Toc127312146)

[Conclusion 38](#_Toc127312147)

[How software requirements can be traced throughout software lifecycle: 38](#_Toc127312148)

[Discuss two approach to improve quality 38](#_Toc127312149)

[Evaluating the process of undertaking a systems investigation taking into consideration: effectiveness in improving a software quality. 39](#_Toc127312150)

[RTM 39](#_Toc127312151)

Graphical user interface, text, application, email

Description automatically generated

# Part1

## SDLC Models

* + **Two sequential SDLC:**

1. **Waterfall Model:**

Waterfall Model is very easy in order to use and understand. Each phase has to be done before moving to the next phase.

Waterfall Model SDLC is as the following:

Requirements > Design > Development > Testing > Deployment > Maintenance

**Advantages of this Model:**

* It is very simple and very simple to understand and use.
* Can gives structure to staff which is inexperienced.
* All tasks are understood very well
* The requirements which sets are stability.
* Very useful to companies and project since it is very benefit to control the management( the plan the staff the track)
* It is suitable when we want to focus on quality more than the cost and suitable time frame.

**Disadvantages of this Model:**

* You must know all the requirements at the beginning.
* You can’t fix errors if you don’t notice them at the same phase. So, you can’t fix error if you don’t detect them during the phase.
* Not recommended for complex project, where there are changes frequently on requirements.
* The testing phase, it comes late after the development process.
* You cannot get benefit of the customers feedback on the current development phase.
* Errors may lead to huge problems and small changes too.

**When to use this Model:**

* When requirements are very clear and clearly known and identified.
* When the product in its definition is stable and not changeable.
* When we have well understood to the technology.
* When we want to make a new version of a product which is existing.
* Whet we want to port a product that is existing to a platform which is new.

1. **Prototype Model:**

It is a very popular used SDLC. We use it when the customer does not have a clear requirements (the exact requirements for the project) in the beginning. What we do in this model? We develop a prototype(of the final product(end product)) based on requirements get from customers and continue developing it and testing it and make changes on it repeatedly until the customer accept the final prototype. And in order to develop the final product, this prototype is the basis.

Prototype Model is as the following:

Requirement gathering

Quick Decision

Refine requirement based on customer feedback

Build prototype

Evaluate the prototype by the customer

The acceptance by the customer

Implementation

Testing

Design

Quick Maintenance

**There are many types of prototyping models:**

1. **Rapid throwaway prototypes**: we make and develop prototype based on initially requirements. Then every customer’s feedback or interaction, the prototype is recreated until the requirement is baselined. The previous prototypes will not be a part of the final prototype which accepted by the customer. To search thought, this model is very useful.

2. **Evolutionary prototype**: since it is very disappointed to through the prototype per each interaction of the customer. The protyped which being developed is refined and change incrementally according to feedbacks from customer until the customer accept it. It save a lot of time because you don’t have to recreated prototype again from zero. If you use a new technology, and this technology is not well understood. This model is very useful if the requirement are not stable at the beginning phase and the project is complex where we have to check once every functionality.

3) **incremental prototype**: In this model, we decimate the final product into various short prototypes and each prototype is developed lonely(invdividually). Then, in the end, All these various prototypes are combined together into a single product. It is very useful that the time between development team of the application and the client reduce.

4) **Extreme prototype**: We use it mainely for web development. It is a three sequential stages:

1) We develop a basic prototype which the whole page is represented by HTML format.

2) Then, the simulation of how the system is worked.

3) Then we implement and integrated the services needed to the final prototype.

**Advantages**:

* You involve users in the development.
* You provide to the user a model which is worked. So, the understanding more about the system that you developed for him.
* You can descover errors earlier.
* You can benefit of quicker customers feedback which lead to optimal sloutions
* Any functionalitiy which is missing can be detect easily and identified.

**Disadvantages**:

* slaw process(prototyping) and it is take a lot of time.
* It the final prototype has been thrown all the cost is waste and go for nothing.
* You convineance that the prototype which you have created is the optimal solustion so you repairing what you have implemented.
* And it increase complexity because the scope can go furhtert than the original plans.

**When to use:**

* When a system needs to include customers feedback and get a lot of feedback and interaction with users.
* So it is very suitable for online systems or web interfces where a huge of interaction are needed.
* It ensures that the system is usable and work with customers due to adapt the feedback from the customers and make it good for him.

**When to use specific prototypeing models:**

Rapid prototyping: when fastly develop software

Evoluationary prototyping: unclear requirements of the software

Incremental prototyping: Huge(enterprise-level) projects and software

Extreme prototying: website developing.

* + **Two iterative SDLC:**
* **Spiral**: risk driven: according to the special patterns of risk for a project. It is has the incremental and sequential elements which guide a team to adopt.(like evolutionary prototying, waterfall, incremental)

**Brief about spiral:**

It include the other life cycle models. And in this model risk handling have been built. And every phase of the project is represented by a spiral(single loop). It combine the nature of prototyping(iterative) and linear of waterfall

**Phases of this model:**

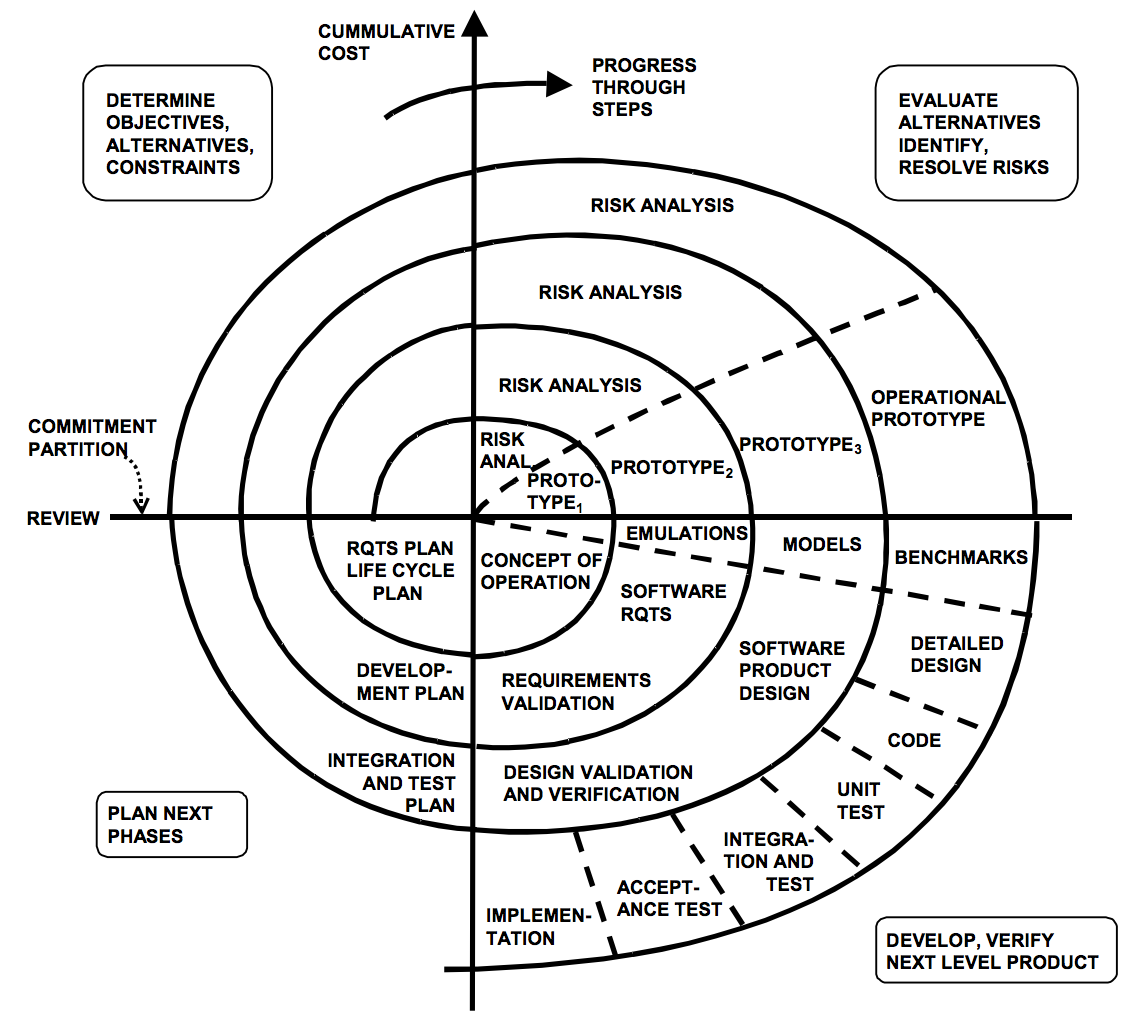
**Planning phase**: gather the requirement(BRS>business requirement specification & SRS>system requirement specification)

**Risk Analysis**: In this phase, identify the risk and identify secondery solutions. So it is a process undertaken. And in the end of this phase a prototype is created and secondery solutions are thinking about and suggested and implemented if any risk detect through the risk analysis.

**Engineering phase**: in this phase we make development and testing. So, the software is being developed and beside that it is tested at the end of it.

**Evaluation phase**: The customer in this phase evaluate the output before we continue and complete next spiral.

**Spiral model is as the following:**



**The spiral steps model generalized:**

1) Requirements of the new system are specified in most possible detail. This contain usually making interview with users which these group of users represent users that intern and extern and other things of the current system.

2) Initially design is done for new system.

3) We create a prototype firstly of the system which is new foom the initially design. It is a small system usually. And it contaion some of the characteristic of final software/product.

4) 4 steps are taken to evolve second prototype: 1) the first prototype is evaluated, take into consideration the strength the weaknesses the risks. 2) the second prototype requirements are defined. 3) the second prototype is planned and designed. 5) the second prototype is constructed and tested.

5) The project can stop if a risk was very huge.

6) The new prototype is again evaluated similar to the previous one, and another (if needed) prototype is developed based on the 4 steps.

7) The previous steps continuous repeated until the client accept and please and convince that the existing prototype represents final product.

8) Construction of the final system based on final accepted prototype.

9)Evaluation and testing for the final system. Maintenance regularly is made to prevent failures on systems and to prevent system work stop.

Advantages:

* There is a lot of risk analysis which is helpful to reduce and prevent risk
* It is very useful for large projects and important projcets
* We can added new functionality(additional) later.
* The software in this model is out early and produced.

**Disadvantages:**

* It is cost in order to use
* The risk analysis, this phase requires: a very and expertise which is high specific
* The success of project is almost depend on the phae of the risk analysis.
* It is not suitable for small projects

**When to use:**

* When the evaluation or risk and costs are paramount
* For projects which have high risk or medium risk
* When customers are not sure of what they want
* When the requirements of the project or software are complex
* When we want to make a new product line where it is very useful
* When the project can contain huge changes
* Agile:

It is a set of principles and a set of values, which agile software development project is applied on. It is flexible, it is good for an environment where there are fact changes happened. According to the optimal practices, so it model and document software.

Principles that this model have:

Customer involvement: The customer must be involve in the development process. He take his role in provide system requirements and priorities them and evaluate the system iterations.

Incremental delevary: Incrementally developed of the software. With customer feedback for the next interation requirements.

People not process: Development team has skills which must take an advantage of. And they must let they work on a way that they develop it without an obligation processes.

Embrace change: There is must be and expectation that the requirements of the system, that they are expected to change. So The system must be design to these refinements accommodation.

Maintain simplicity: Elemintate complexity as you can, and ensure simplicity in the software and in the development process.

Agile brief:

* It is a model which is an incremental.
* Rapid cycles and incremental which the software is developed in. Which lead to releases which are small and incremental and that each release building is base on the functionality of the previous release.
* To ensure that the quality of the software is maintained, So Test is done thoroughly each release.
* It is very suitable for applications which are needed to be done on time frame(time critical).
* There is one of the most famous agile SDLC is extreme programming(XP)

Extreme programming(Ajile)

* Every day there are a build of new versions many times
* Every 2 weeks increments are delivered to the customers.
* To accept the build all tests have to be run for it and for every build and it is just accept the build which the tests successfully run.

A cycle of the extreme programming.

Select user stories for this release

Break down stories to tasks

Plan release

Evaluate system

Release software

Develop/integrate/test software

Advantages:

* Since there is a delivery of a software which is useful and continuously delivered, we got te satisfaction of the customer.
* We focus on the interactions and people rather than focus on the tools and focus on the process.
* Permanently interact between testers and the customers and the developers.
* Frequently delivered of a software which is working(weeks)
* The conversation is face-to face- which is the optimist.
* Developers and people have coopearation every day.
* There is always attention and it is continuous and it is to good design and to the excellence of the technical.
* If the circumstances has changed, there is no problem. It is adaptive them.
* There is no problem of late changes which happened in requirements.

Disadvantages:

* It is not easy to evaluate and assess, how much effort it need at the beginning of the SDLC for the software.( for some software deliverables(large software).
* Designing and documentation, Some of necessary of them there is a lack of emphasis on them
* And also if the customer interactions and represent not clear what the customer want the outcome to be the project go beyond the origin plans easily.
* There is a decisions through the development process. Only those who are capable of taking the kind of it required are the senior/big programmers
* It is not safe for the programmers which are beginners, just only if there is an expertise between them and expertise resources.

When to use:

* When you want to see delivered of software which is useful
* When you focus on the time(schedule).(weeks deliverables)
* High interaction and involved of the customers.
* Expertise resources are exist.
* Systems want to be established quickly

## How risk is being manage in SDLC

What is risk management on SDLC firstly?

During the SDLC there are issues maybe appear during the project and have negative effect on the project progress and impact its successes. So SDLC models are managed these issues (risks) by identify them, and analyze them, then assess them, then found strategies to decrease and prevent them. This process have to be taken at all phases by the development team In the project. So ,mitigation strategies are always updated. The risk management process is a continuous process which improve the development process by decrease and prevent risks.

The risks are increase or decrese based on the isuues of like size of the project: if the project is larger, the risk is increase. The cost issue: Have we had the budget for this project? The length of the project can we finish the project with the time we have. The economic value: does the business effort value the value returns to the company.

At SDLC models risk is listed(brainstorming/past experiences of projects) . And the assess the impact of risk. The come up with strategies to mitigate and prevent it. Throughout the development process risks will always be monitored and tracked. So a new risk will appear and then identify them and control. Then the result of risk management is reported. So it becomes an experience that you can return to and learn from on future projects.

For example Agile SDLC model:

Risk is being manage in Agile during the planning phase. So during the planning phase the risks are identified and assessed. Then strategies are made(developed) to decrease and prevent these risks during the review phases and the meatings that held to reflect on the work and identify areas for improvement phases. This allows for continuous risk management during the development process. For example: during the planning phase: the team identifies possible risks and then they assess their likelihood and impact. During the review and the meetings which held to reflect on the work and identify areas for improvement phases, the team get out with a strategies(mitigation) to the identified risks to be addressed. Ajile SDLC have many things which help to manage risk, it is rapid cycles of incremental developed for the software and this lead to incremental releases and each release is base on and building on the previous functionality. And test is made for every release thoroughly. And the incremental nature make it better to manage risk. And the involvement of customers during the development process and the permenant cooperation between developers and testers and customers. And the adaptive for changes. All these are help in manage risk. In ajile the project. Risk management is integrated into its management process. We start it by identify the potential risks early in the software, then we continue that throughout the software, we do that through the continuous monitor and assess for the risks. Also, The Ajile team’s using plan for risk management which is specific to to the software(project), and it is review and update continuously. The plan contains assessment process for the risk which is used in order to idenftify risk evaluate risk and prioritize risks. And through implement the plan of risk management we metegate the risk, this plan contain strategies like designing the system and testing the system to decrease and minimize risks. Put a team for risk management, and a dynamic process management to ensure identify risks and evaluate and manage risks. There is cycle which is continuous of assessing in Ajile in the risk management process, and monitoring and controlling the risks, that enable the team to quickly take decicsions. All of this helps to reduce risks and their impacts on the project.

In sequential models such as waterfall:

Identified of risks and assessing early in the project then evaluating and prioritizing these risks. Then strategies(mitigation) are got out with before continue to the next phase. A plan of management for risk is developed and implemented, which the strategies for mitigate and prevent identified risks. It is continuous reviewed and updated regularly during the project. It is completed . The risk management is completed typically before the actual development phase begins. The risks must be will understood on the Waterfall model.. So, it is more systematic in risk management process. For an instance, in Waterfall, the team (development) identify risks and assess the likelihood of them and their impact. Then they developed stratieges of mitigation and they implement them during the design phase. .Waterfall model, in this model you have to fully complete the phase and try to get out of perfectly. And during the development process, if a risk arised on previous finished phases, you can’t return and fix it. You have to complete the cycle. It is based on that you finished each phases perfectly.

Risk management in spiral model:

It is start at the initial of software development process and conitiues through it to the end. Spiral includes risk analysis and there is a step for management for risk on each single spiral(iteration), in which we identify the potential risk and evaluate it, then develop plan for risk management, and implement them to prevent or decrease risks from take place. In addition to this, the risk management plan is updated and reviewed regularly by the team to ensure the effectivety of it throughout the project. In other words, the spiral model enable us to proactive know the risk to manage them(proactive approach) in order to manage the risks.

## discuss using example(s) a particular lifecycle model for a specific development environment:

SDLC model is selected for environment based on many factors: the size, the cost, the time, the stability of requirements, and that the requirements are clear or not…

For example:

If we want to make a new version of the latest version of Iphone. In this project we have all the requirements, and we understand the technology and we just have to update the device to a new version. So Waterfall is the most suitable SDLC model for this project since all requirements are well know and clear and stable and have familiar with technology.

While if there is a change in requirements and complexity we will start think about using Ajile SDLC or Spiral SDLC…

…………………………………………..

## Benefits of Waterfall model in large software development project

1) A phases which are clear and Excellent(distinct): It is clear phases such as: requirements gathering, design, implementation , testing, and maintenance, so this help to keep on budget and time since that will help us on ensure this.

2) It is good for well-defined requirements progects

3)The easy of managing and controlling: Since each phase must be fully completed before moving into the next phase. And identify and correct problems by this way is become easy.

4) Documentation is good in this model: It is a model which is encourage on documentation on a good way, for that each phase on the Waterfall model is completely documented, and this can be beneficial for the mainenace and improves of future.

5) It is suitable for the fixed budget for the project

6) It is can deal with large teams easily since every teams will know in any phase we are.

## Discussing the suitability of software behaviour design techniques using examples.

They are used for that be ensure that the needs and requirements are met by the software(users and stakeholders). Modeling and designing the software systems behavior, this is what they do. So, they are very suitability in that aspects.

An examples :

Use case diagram: This is a technique which is used for modelling the software system functionality. In addition to this, they users >>>>>> the tasks that he need to do and perform using the system(software), it involves identify them. And represent them by create models. To ensure that functionality that users expected and need are there and it meet the requirements that they require.

State machine diagram: It is another diagram which helps to model the system behavior. But this is different; It is doing that in a forms of states and transitions. So, it is about creating models, these models represent the following: the various states that the system can go through and what happen between these states of transitions. And this is also help that the behaves of the software is meet the needs and requirements of users.

For examples if we take the state of human walk into consideration and want to represent its behaviour using design techniques (finite state machine(FSM):

Sprint(increase speed)

ran

create

Walked

Decrease speed immediately to zero

Decrease speed

Increase speed

Stop moving

Decrease speed

And another examples are use case diagram: it provides better understanding to requirements of system. For example if we have for example ‘pay’ key and there is to actor to the system (customer and merchant). So :

Diagram

Description automatically generated with medium confidence

So these are very help techniques since they help us to understand requirements and know that this what the user want(the specific behavior of the software).

## Analayze range of tools(software behavior design techniques) and too techniques:

## Two techniques:

* State machine diagram: As we say, It is another diagram which helps to model the system behavior. But this is different; It is doing that in a forms of states and transitions. So, it is about creating models, these models represent the following: the various states that the system can go through and what happen between these states of transitions. And this is also help that the behaves of the software is meet the needs and requirements of users.
* Use case diagram: As we say, This is a technique which is used for modelling the software system functionality. In addition to this, they users >>>>>> the tasks that he need to do and perform using the system(software), it involves identify them. And represent them by create models. To ensure that functionality that users expected and need are there and it meet the requirements that they require.

## Two tools:

* Draw.io(<https://app.diagrams.net/>): It is a very beneficial tools to draw and implement different techniques such as use case diagram and finite state machine… It is very simple and clear. Providing users a ready framework which have different shapes and the ability to control properities for these shapes. It is very useable and reliable. It gives range of templetes and range of examples that we can use It as point to start from. Which save time and save effort. And multiple users can work simultaneously on the same diagram. It integrates with other tools such as OneDrive and Dropbox make sharing and storing diagram easy. And the compatibility of it which mean that you can install it on windows Linux MacOS and more. So it is accessible to large numbers of users.

Graphical user interface, application, table

Description automatically generated

* Wondershare EdrawMax: It is a very excellent offline tools, which enable you to draw diagrams which represents software behavior. It is very simple and clear. It is usable and reliable. It is provide you many shapes and a framework where you can draw diagrmas easily and fastly. It is user friendly so the user interface is easy and it is also provide range of examples and templetes which users can use and consider it their start point and provide funcionlaity such as drag and dropp so that making it easy to create diagrams and also the permissions of multiple user to collaborate simultaneously on the same diagrams. And the compatibility(supported by many platforms) and the integrations(with tools such as google drive and OneDrive.

Graphical user interface, table, Excel

Description automatically generated

## Differentiate between Finite FSM and extended FSM

Human walk(Finite FSM))Trafic light control system also)

Sprint(increase speed)

ran

create

Walked

Decrease speed immediately to zero

Decrease speed

Increase speed

Stop moving

Decrease speed

A character in a game which has only 5 moves(Extended FSM)(computer keyboard with limited allow key pressed).

Graphical user interface, application

Description automatically generated

So the Finite FSM mention the state that the system go through, but the Extended FSM mention the A states where it is based on condition such as these states will be allowed or unallowed until that condition is occurred or achieve.

## Data-driven improve reliability and effectiveness of software:

The data-driven means that to depends on data to the software as the base source of input and making decision. And this improve reliability and effectiveness as the following:

1) It is more accurate to depend on date analysis to make decisions than make decisions based on human opinions and their decision making.

2) It enhance the performance, Since we make informed decisions by use the data, such as the predict of the usage resource and the adjustments that we make based on.

3) It is suitable for large scale applications, since the process for data that processed by data driven are increase so the software can expan based on that and continue the effective of perform.

4) It is more reliable in detect faliures and respond to it in real time more that manual interference

5) It is definitely better on the aspect of make of the decisions, since it does not based on guess and intuition. It is based on facts. So this make it better.

## Compared two technical solutions and how

We can compare between technical solutions based on many things: funacitnolity,and performance and the scalability and the security and the integration and the support(maintain) and the satisfaction of user and the cost and the ease of use…

## Compare two technical solutions

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Uber | Technical feasibility | Economical feasibility | Organizational feasibility | Scheduling feasibility | Legal and contractual feasibility | Political feasibility |
|  | Uber has a wide accessible through a website and mobile app | The business model:  Passengers satisfaction: since it connect drivers with passengers through platform, which make it easy to find services to transportation for passengers. | The structure of management: the centralized management structure that Uber has, with clear lines of authority and making decision processes | The availability of driver: it must ensure from provides enough drivers for the places where there are high demands | It must comply with various laws: releated to the : services of trasportation and employment… | Navigtions transportation services complix regulatory. Can involve federal regulations/ local and state |
|  | The user experience on Uper that it is user-friendly platform. Since it has features like estimated arrival times and in-app payment. | Costs of operating: Uber incurs cost: the driver pay, the expenses of marketing, the technology development, but it benefits from the scale grows for it | The culture: it is has a fast paced and innovative culture company, that encuourage rapid innovation and growth | The deman of ride: It must predict accurately where there are a demand ride and mapping it with the availability of drivers. So riders can find drivers when they need | The necessary licenses and permits must be obtained by Uber to operate in each region.(time consuming process) | Engage in lobbying efforts in order to shape the regulatory environment and its interest promoted.(affect the public opinion) |
|  | It has a technical infrastructure, so it include GPS-bases mapping,real-time communication between passengers and drivers, and cloud-based data storage | The strategy of pricing: Uber ensure that It generates enough revenure to cover the operating costs, through uses dynamic pricing which edit fares based on supply and demand. So it is economically feasible in that aspect. | The satisfaction of the employee: In the past it was a topic of debate but the company has implemented measures to optimise the conditions of working and employee engagement. | The matching of real time: it has an efficient scheduiling system to ensure accurate matching and quick matching ,so that real time uses to match riders with drivers. | The contractual agreements: complex legal can involve and commercial considerations,since Uber must negotiate and execute contrats with suppliers drivers partners | The public perception or transportation services impact it and negative perceptions. |
|  | The integration with different systems(google maps, and different gateway for payment which make it easy for users to access) | The market demand: There is a lot of demand on transportation services. | The resource allocation:It is well funded organization; so that allowing it to allocate resources towards its operations, technology development, and marketing efforts | The optimization of the route :They have to optimize the routes of drivers to reduce travel time | It must manage the liability for accidents which occur during driving. | Manage relationships with different stakeholders(government regulators, the providers of transportation, the local communities) . |
|  | The scalability of Uber: it can be easily expand to new regions and increase its customers numbers. | Competition: there is a lot of competition from other transportation services, such as taxi companies and other public transportation services, but it made a a name for itself and make it known for a lot of people by recoginze its brand. | The legal and ethical considerations: Uber must navigate complex legal and ethical considerations, such as complying with regulations and protecting the privacy of its users. | The dynamic pricing: it must be integrated with the scheduling system, to ensure accurately fare calculated for the journey fees. | Must comply with different Labor laws related to employee benefit, minimum wage, over time… | The stability of political environment. Because instability political can lead to more regulations and negative impacts to its services |
|  | The security of uber: excellent security measures: the data storage is encrypted, and check the background for drivers; so (passengers safety and their data) | The regulations: Uber is compliance to local regulations and obtain necessary licenses to operate | The stakeholder relations: it must manage relationships with different stakeholders, its riders and its drivers and the regulators and communities that it operates with | The reliability: it must ensure that its schedduiling system is reliable to reduce failures and that passengers can find drivers when they need them. | It has to ensure the privacy, and has to comply with data privacy regulations. | It is supported by economic consideration, job creation, tax revenue, local economic development |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Careem | Technical feasibility | Economical feasibility | Organizational feasibility | Scheduling feasibility | Legal and contractual feasibility | Political feasibility |
|  | careem has a wide accessible through a website and mobile app | The business model:  Passengers satisfaction: since it connect drivers with passengers through platform, which make it easy to find services to transportation for passengers. But it take commision on eadh ride. | The structure of management: the centralized management structure that Careem has, with clear lines of authority and making decision processes | The availability of driver: it must ensure from provides enough drivers for the places where there are high demands | It must comply with various laws: releated to the : services of trasportation and employment…  And data protection laws and customer protection laws… | Have to complay with all applicable laws and rigulations for every county it has a branches there. |
|  | It is user friendly providing smooth user experience and allow riders to reques a ride and track their driver’s progress quickly and easily | Costs of operating: Careem incurs cost maintaining its mobile app, server infrastructure, payment processing systems, and customer support | The culture: it is has a fast paced and innovative culture company, that encuourage rapid innovation and growth | The deman of ride: It must predict accurately where there are a demand ride and mapping it with the availability of drivers. So riders can find drivers when they need | The necessary licenses and permits must be obtained by Careem to operate in each region.(time consuming process) | The opinion of people to Careem can vary widely. |
|  | Integrates a secure payment system into its platform. Enabling seamless trnsactions. | The strategy of pricing: Careem ensure that It generates enough revenure to cover the operating costs, through uses dynamic pricing which edit fares based on supply and demand. So it is economically feasible in that aspect. | The satisfaction of the employee: The employee are very satisfied | The matching of real time: it has an efficient scheduiling system to ensure accurate matching and quick matching ,so that real time uses to match riders with drivers. | The contractual agreements: complex legal can involve and commercial considerations,since Careem must negotiate and execute contrats with suppliers drivers partners | The public perception or transportation services impact it and negative perceptions. |
|  | The integration with different systems(google maps, and different gateway for payment which make it easy for users to access) | The market demand: There is a lot of demand on transportation services. | The resource allocation:It is well funded organization; so that allowing it to allocate resources towards its operations, technology development, and marketing efforts | The optimization of the route :They have to optimize the routes of drivers to reduce travel time | It must manage the liability for accidents which occur during driving. | Manage relationships with different stakeholders(government regulators, the providers of transportation, the local communities) . |
|  | Integrates with third-party(google maps) | Competition: there is a lot of competition from other transportation services, such as taxi companies and other public transportation services, but it made a a name for itself and make it known for a lot of people by recoginze its brand. | The legal and ethical considerations: Careem must navigate complex legal and ethical considerations, such as complying with regulations and protecting the privacy of its users. | The dynamic pricing: it must be integrated with the scheduling system, to ensure accurately fare calculated for the journey fees. | Must comply with different Labor laws related to employee benefit, minimum wage, over time… | The stability of political environment. Because instability political can lead to more regulations and negative impacts to its services |
|  | Gps using and mapping technology to track location of passengers and drivers and in order to provide the real time of navigation. Mapping providers and algorithms or matching riders with drivers which impact accuracy and efficiency of their service overall. | The regulations: Careem is compliance to local regulations and obtain necessary licenses to operate | The stakeholder relations: it must manage relationships with different stakeholders, its riders and its drivers and the regulators and communities that it operates with | The reliability: it must ensure that its schedduiling system is reliable to reduce failures and that passengers can find drivers when they need them. | It has to ensure the privacy, and has to comply with data privacy regulations. | It is supported by economic consideration, job creation, tax revenue, local economic development/  Competition(there are a lot of cometittors in that fields) |

Careem and Uber is approximately similar technical solutions, there is little different between them, There is some little difference in some aspects, but in general they have the same feasibility. They both have an excellent technical feasibility that they have excellent platforms and integration with third parties which make it easier to use by customers(google maps). They both have excellent economic feasibility, but Careem take commission for each ride. They both have excellent Organizational feasibility but everybody know that Uber has a reputation issue on the past about its customers satisfaction because of the bad drivers and journeys conditions, but now Uber work very hard to treat these things. And both companies should manage relationships with different stakeholders. And the problem on the legal issues in some counties which had faced. Both have excellent scheduling feasibility but they must take care of drivers availability on the places where there are high demands and reduce failures and enhance reliability that riders can find drivers when they want. And by mention legal and contractual feasibility, both have to comply with all applicable laws and regulations for each country, which some of them are complex and take time. And they have the same political feasibility, so they affect with the public opinion and politician stability and they have a lot of competition. In general, they both are excellent technical solutions And Each one is better in some aspects of feasibility, but in general they are excellent and equal. Uber maybe better than Careem in Economic feasibility, since Careem have commission for each ride which can lead to lose some customers. But Uber has issues about customers satisfaction on the past, so Careem has the higher hand on Organizational feasibility. They are equal for the rest aspects approximately. But as I said, in general, they are approximately look like having the same functionality and the same performance and scalability and other features.

## Explain feasibility report purpose.

The feasibility report: a study that evaluate what the software/project is capable of(its potential): the practical and the financial behavior of it. This is base on factors: study technical feasibility, economic feasibility, organizational feasibility, Scheduling and legal and contractual feasibilities, and political feasibility. The purpose of this report is to if we should continue into the project or not, it is consider, and to increase the success possibilities and decrease the fail possibilities.

## Discuss the components of Feasibility report

1) Economic feasibility: It is answer a questions: Do we have budget to carry out this project? What is the value of the project and benefits.

So we make cost-benefit analysis. It is studied continuously during the SDLC for the project which we based on decide to kill the projet or to continue the project or to redirect the project.

There are Tangible benefits and intangible benefits: the Tangible benefits( the items which we can measured it using dollars). Such as if we get higher profit than the previous month. (improve an aspect of the project work management…)

The intangible benefits cant be measure using dollars easily. Such as the satisfaction of employees and the increase in their morlas.

And the tangible costs: it is which can measured by dollars such as the costs of hardware,Labor…

The intangible costs are which can’t measured by dollars easily such as the loss of morale of employees and customers.

So we take different things cost and benefit in consideration in order to decide to kill or redirect or continue the project. Calculate Return on investment.

2) Technical feasibility: Can we make this project? It is an assess for the ability of the organization to construct the project(system). Based on many factors: the size of the project(the size of team and the size of the project how much effort it cost), the structure of the project, the development group( are they familiar with technology…), users group(are they familiar with the similar system), so we analyze the risk of the project based on them

3) Organizational feasibility: Are there users for the system If I complete it? How well the users will accepted the system ultimately? Are they going to come if we build it. In order to enhance it we can ask for regularly users feedback, make presentations about the benefits of project, create prototypes.

4) Scheduling feasibility: Can we finish the project by the time frame we have? If not, there is no meaning of the whole work? How much time will take by the project? If we can’ the project will fail, so must take into consideration.

5) Legal and contractual feasibility: We must check the compliance of the project meets to the laws and regulations, and avoid these risks such as : avoid the copyright and other issues.

6) Political feasibility: This study the key stakeholders in company, how they view the project that you proposed to them. Maybe the project will be block if they not supporting it, or disrupt it or change its focus.

## Assessing the impact of criteria feasibility on investigation of the project

They have a lot of impact such as: The ROI and The BEP, The ROI(return on investment) is how much the percentage of profits. And the continue of the project or the decline of it is based on the flow of ROI, and also the break even point(BEP) how much time I need to return the original investment(0), If I will take a lot of time, this also can cancel or redirect the project. The size of the project and the familiarity of development goup to the technology. This can decide if we going to cancel or continue or redirect the project. If we find there is a lot of risk, development team is not familiar with project, so this will affect too. The structure of the project, will be a very criteria which can impact the project progress. This will have a lot of impact that we fail to integrate the new system with the procedures and equipments available, and cost and time will be incaurrately estimated, and the expected benefit will be failure to attain. Also, there target market(there is no user interest on use the system), So the target market and the study of the markets demands and user interests will lead to cancel or proceed the project or redirect it. The time frame we have, if we can’t finish the project by the time frame we have, there is no meaning of the project, so Fail!!. Also If there is a low or regulations that the project is non compliance to, this will make the company to cancel or redirect the project. And if the stakeholder are not support the project or are support it, this is a huge issue that will determine the result of proceed or cancel or redirect the project.

# Part 2:

The Idea is an online food service. We want to make a system where deliver food to customers and provides many services.

## Feasibility study of the online food service:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| The idea | Technical feasibility | Economical feasibility | Organizational feasibility | Scheduling feasibility | Legal and contractual feasibility | Political feasibility |
|  | Lack of experience employee(technical) | No good budget to support the system | The restaurant Is not very popular, so, it is not expected that the website will attract them | There is no good time to finish the website we have a few days | We must comply to some local and national regulations | The owner of the restaurant is not very encourage about the website |
|  | There is no enough employees to improve quality | Gifts employees who are doing well and achieve progressing | There are a presentation which tell people about benefits of the website and features. |  | We must comply with regulations according to payment process and service product and taxes. | The website have some type of culture which can cause to be decline |
|  | We have clear steps about what we had to do |  |  |  |  |  |

There is a high risk to continue building this website, since we don’t have enough time to make the website, and we don’t have enough employees and expertises, but we have clear steps about what to do. There is no good budget to support system, but we gift employees who doing well. The restaurant is not very popular, but we make presentations above people showing them the features and benefits of website. There are not a lot of regulations except the payment procsess regulations. The owner of the restaurant is not encourage, and some type of culture which the website have can lead to cancel. It has high risk to complete. We will try to make the minimum main requirements and features, which can decrease the risk.

## Requirements:

1) Login page

2) Home page

3) Menu page contain food menu

4) Payment Page

5) sign up page

## Non functional requirements:

1) useability: simple and use to use. Must be simple and clear and include clear user interfaces.

2) portability: Since it is a website, it can be open using many devices and platforms. It should open in multiple platform

3) **reliability: reliable, since it is simple and few number of users which can be increase with the scaling of the customers numbers. A department will be specialize to control the call center to monitor and continued customers comments and problems. And a fast new cars will be provided to ensure the velocity of operations.**

**4) performance: a high performance, and speed and fast service and operations. It should be competitive with other restaurant and online food services, and high quality.**

**5) Security and privacy: user information which found in database should be back up and regularly monitor.**

# SRS document:

## Introduction:

* The purpose of this website is to give online food service which people can request food online from there home and pay directly.
* The scope of this website Is food, A service that people buy food online, it is a restaurant service, they go to the website and choose their needs. The software will introduce (Shawerma, Burger) as a food menu, the software will not give service to other restaurant it is a service just for our resturent.

This software if for online food sale service, it save people time and effort buy get the food to their home without go to bring it.

There are will be login page to enter to the website, and a sign up to register customers. There are will be a home page, which contain a link to the menu page, which contain food which we serve, and if customer press on type of food he will go to the payment page.

* Definitions:

1) login page: is the page that user use it to register to the website

2)Menu page: is the page that user use It to view food menu

3) Sign up: the page which user use to register

* Overview: This SRS is to document user and system specification using IEEE(830) templetes, we will cover overall description which include website functions and chsaracteristics of customers and constraints and perspective for product.

## Requirements:

Functionality and futures:

1) Login page

2) Home page

3) Menu page contain food menu

4) Payment Page

5) sign up page

## Non functional requirements:

1) useability: simple and use to use. Must be simple and clear and include clear user interfaces.

2) portability: Since it is a website, it can be open using many devices and platforms. It should open in multiple platform

3) **reliability: reliable, since it is simple and few number of users which can be increase with the scaling of the customers numbers. A department will be specialize to control the call center to monitor and continued customers comments and problems. And a fast new cars will be provided to ensure the velocity of operations.**

**4) performance: a high performance, and speed and fast service and operations. It should be competitive with other restaurant and online food services, and high quality.**

**5) Security and privacy: user information which found in database should be back up and regularly monitor.**

# SDS Document

## Design overview

The design of the website(system) is going to solve how the user will sign up to the system and what are their permissions inside it, and customer order to the food, and the system manipulations in general. It has many interfaces(login,sign up, Home page, Menu page, payment).

## The software major functionalities: (high level overeview of the design of software).

Use case diagram for online food service

Diagram

Description automatically generated

## Detailed Design

Diagram

Description automatically generated

Flowchart:

Diagram

Description automatically generated

Peudecode:

START

1) Customer sign up

2)Customer Login

3)Customer view menu food

4) Customer choose an item

5) Customer pay for the food

END

## User interface Design:

User interfaces:

Login

Graphical user interface

Description automatically generated

Sign up

Graphical user interface, application

Description automatically generated

Home page

Shape, polygon

Description automatically generated

Menu

Chart, scatter chart

Description automatically generated

Payment

Chart

Description automatically generated

It is userfriendly and simple, but it wants more features and images and other elements which must be taken into consideration on the future maintain.

I have used

## Data design:

DFD(Context level Diagram)

Diagram

Description automatically generated

ERD

Diagram

Description automatically generated

* Many customers order one order
* Many order is ordered to the restaurant management
* Many customers pays
* Many payment stored in one transactions database
* One transactions database send customers data(releated to the product which customers pay) to customers database.

# Test plan:

Graphical user interface, application

Description automatically generated

Logout successfully

A close up of a calculator

Description automatically generated with medium confidence

Done

Graphical user interface, text, application, email

Description automatically generated

## Login

When there is no input

Graphical user interface, text

Description automatically generated

No input entered

So:

Fail access

Text

Description automatically generated

Invalid password or username

Text

Description automatically generated

Fail to access

Graphical user interface, text, application, email

Description automatically generated

Account and password are valid

Text

Description automatically generated

Fail to access

Graphical user interface, text

Description automatically generated

## Sign up

When username and password are blank

Graphical user interface, text, application

Description automatically generated

Result:

Graphical user interface, text, application, email

Description automatically generated

If enter any username and password

A picture containing graphical user interface

Description automatically generated

Result: Success and go to login page

Graphical user interface, text, application

Description automatically generated

## Conclusion

In conclusion, we have covered a design for the software from many aspects, The design have exposed a lot of mistakes and solutions to implement the system, We have reached to optimal solutions and clear, so, Different diagrams have help a lot to implement the code and insperations. We must work more in general to maintain our design, So, payment process must be complete by its detailed, and hardware and software(Database more detailed) must be determine more. We have covered Detailed design, user interface,design overview, and test plan and conclusion.

## How software requirements can be traced throughout software lifecycle:

Different issues and steps:

1) Identify the requirements and documenting it in(clearly) (SRS document)

2) Map the requirements with the design components, by map between the requirements and design phase and specific the functionalities and components which are releated to these requirements(mapping them together)

3) Track the requirements implementation by ensure that they have being correctly implemented based on design.

4)Test the requirements to ensure that they have the expected functions(Testing)

5) Track them through the maintenance phase to ensure that they are up to date. Since any changes mean update and change requirements which need to be tracked.

6) Using technicques such as RTM Requirement traceability matrix) which are make it easy to trace and manage requirements and maaping them to test case.

## Discuss two approach to improve quality

1) The review of code: An approach to improve the quality of software is by review the code. Which includes to have team of developers, testers or quality assurance professionals which who make review to the code before the release of it, so you ensure that there is no bugs on it, and that it meets the requirements, commitments with the coding standards. This improve software quality through detect bugs and problems early in the development process. So, you fix problems when before the development process continue and the cost of fix these problems double to ten times or more.

2) Automated Testing: this help improve quality of the software by automated testing. Which includes using software tools to automate the testing tasks which are repetitive and run tests regularly. This is also help in early detection of bugs and problems, and increase efficiency of testing process, and help in avoiding the possible mistakes that human could made (the human nature can made mistakes or forget somethings).

## Evaluating the process of undertaking a systems investigation taking into consideration: effectiveness in improving a software quality.

It is a very important and crucial step in order to improve a software quality. It effectiveness in improving a software quality are a lot including:

1) Identifying the root cause of software quality issues. Since it gives understanding to problems cause and clear steps and ways to avoiding them

2) Improve the requirements, the investigation process may shown that the requirements were incomplete or other issues in some aspects, This can used to improve requirements and ensure that the development software is more suitable in meeting customer needs.

3) Improve design: the system investigation process may expose problems through the design, for example: it can appear that there is an issue in scalability of the software(poor), or the performance of it is not better than the previous one, by comparing both designs, or there is a security issues. So, having a knowledge about these issues will improve the quality of software and made it better in meeting users expectations and needs.

4) Improved the testing process, the investigation process can lead to detect problems in testing process, for example: there is a problem in the coverage of the test, it is inadequate, or there is not enough automated testing. So improving testing process improve software quality, since developers become able to detect bugs better than before.Also if new bugs appear in software, improving testing process reduce its risk by fast detect it.

# RTM