

SRS Software Requirement and Specification Online Fuel Delivery

In the name of Allah, the most beneficent, and the most merciful.



vumalik.blogspot.com



All CS Free Services



- LMS Handling
- Quiz
- GDB's
- Assignment
- Project



Contact Us

VU Malik

0317 6291353

To seek knowledge is the duty of every Muslim man and woman

Online On Demand Fuel Delivery

Software requirement specification

Version 1.0



Group Id : MC123456789

Supervisor Name :Hafiz Muhammad Azeem Sarwar

Date	Version	Description	Author
02/06/2022	1.0	The purpose of this project is to develop a website which will offer online delivery of fuel on demand. At some areas, the fuel stations are situated at very long distances. In such cases, if a vehicle gets stopped on the road due to shortage of fuel, the owner of the vehicle have to manually drag his vehicle towards the nearest fuel station, then it will be very hard for him. This situation can become hard if the location is new for the vehicle owner.	MC123456789

Revision History

Table of Contents

1.	Scope (of the project)	
2.	Functional Requirements Non Functional requirements	
3.	Use Case Diagram	
4.	Usage Scenarios	
5.	Adopted Methodology	
6.	Work Plan (Use MS Project to create Schedule/Work Plan)	

1. Scope of Project:

The purpose of this project is to develop a website which will offer online delivery of fuel on demand. At some areas, the fuel stations are situated at very long distances. In such cases, if a vehicle gets stopped on the road due to shortage of fuel, the owner of the vehicle have to manually drag his vehicle towards the nearest fuel station, then it will be very hard for him. This situation can become hard if the location is new for the vehicle owner.

The proposed system is to develop a website to deliver the fuel to those users who need to refuel vehicles at any location and time. In this website there will be three modules such as Admin, User and Fuel Station. Admin will approve the fuel stations registered in the website. Fuel Stations can register themselves to provide fuel delivery service to the users. Fuel Station will add its complete information such as its address, Google map location, and the kind of fuel it provides. Users can search for fuel stations by their location, fuel station name and can book the fuel for their vehicles online.

2. Functional and Nonfunctional Requirements:

Functional Requirements:

User Module

- ✓ Register
- ✓ Login
- ✓ Search Fuel Station
- ✓ Place order
- ✓ Make Payment
- ✓ Get Delivery

Fuel Station Module

- ✓ Register
- ✓ Login
- ✓ Create Fuel Station
- ✓ Update/Delete Fuel Station
- ✓ Receive order
- ✓ Approve Status

Admin Module

- ✓ Login
- ✓ Approve/Reject Fuel Station
- ✓ Approve/Reject Users

Non-Functional Requirements:

Any requirement that specifies how a system performs certain functions **Nonfunctional requirement** describes the general properties of a system or application. It is also known as *quality attributes*.

- ☐ **Performance:**
- ☐ **Reliability:**
- ☐ **Availability:**
- ☐ **Usability:**
- ☐ **Quality:**
- ☐ **Maintainability:**
- ☐ **Safety:**

Fast and appropriate search results

System user should have sound knowledge of SQL Server for retrieving required information in less time.

Bugs free product

Minimum 5 to 6 test cases will be developed for this system to help in removing bugs.

Reliability

The system should be able to handle all the workload of all the defined systems with less time. It means the extent to which program, performs with required precision. The website developed should be extremely reliable and secure so that information about any question etc. is not leaked. The system shall not be down more than 2 times in a year.

Availability

The software will be available only to authorized users like students to view their enrolled course, admin to add an update/delete student details. Checking that the system always has something to function and always pop up error messages in case of component failure. In that case the error message appear when something goes wrong so to prevail availability problems.

Maintainability

The application is to be designed so that it easily maintained. Also it should allow incorporating new requirements in any module of system. Backups for database are available.

Usability

While designing the forms we will keep in mind that the beginner users will also be using the system. So we will use standard icons on the forms.

Portability

There is a web based application and is built in PHP and my SQL. This system will perform well on all Microsoft O/S environments. The application will be easily portable on any allow window based system. The term defines an application's flexibility when used on multiple platforms or instantly accessed from the Internet, a desktop or network.

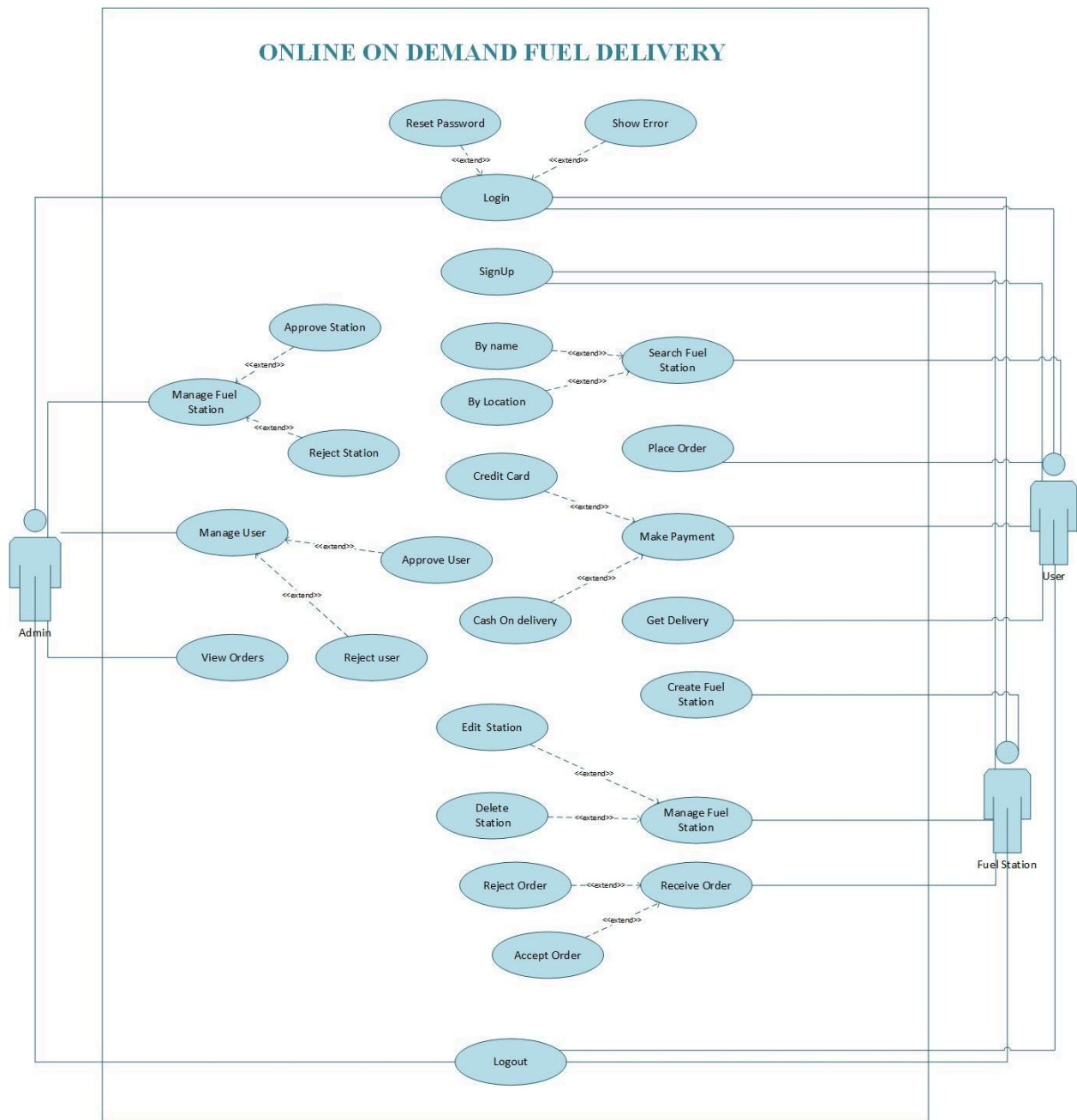
Supportability

The system would use windows pattern for menus making, icons etc. or any current standards operating in open source projects.

Tools:

- HTML, CSS
- JavaScript, PHP
- WAMP server,
- __SQL, Dream viewer, Notepad++

3. Use Case Diagram(s):



4. Use Case's Scenario:

5.

Functional Requirement title	Login
Functional Requirement Id	UC-1
Actor	Admin, Fuel Station, User
Actions	All authorized users of this application will login.
Description	All users will login to system by using email and password for login to the systems. Users of this application will provide with email and password to enter in the required fields. Application will authenticate and allow only registered users for login
Pre-Conditions	All users must have application installed.
Post Conditions	All users must login after authentication
Exceptions	Application crash
Author	MC123456789

Functional Requirement title	View users
Functional Requirement Id	UC-2
Actor	Admin
Actions	Admin will login into the system application. He will open user panel. He will view the registered users
Description	Admin will open the application. Admin will login to the system application by providing the credentials into the required fields. Admin will click on view user button link. Admin will be shown with a panel where he will be shown with all added user in the system. Admin will view all the user details. He will view the details of the user by using this panel.
Pre-Conditions	Admin must be logged in.

Post Conditions	User must be Approve by the admin
Author	MC123456789

Functional Requirement title	Manage Signup Request
Functional Requirement Id	UC-4
Actor	Admin
Actions	Admin will login into the system application. When the User or Fuel sending a request for registration. Admin can be able Reject and accept request
Description	Admin will open the application. Admin will login to the system application by providing the credentials into the required fields. Admin will click on registration request button link. Admin will be shown with a panel where he will be able to add new user or fuel station' details. Admin will register the user or fuel station.
Pre-Conditions	Admin must be logged in.
Post Conditions	User or Fuel must be registered
Author	MC123456789

Functional Requirement title	Manage User
Functional Requirement Id	UC-5
Actor	Admin
Actions	Admin will login into the system application. He will click on manage user panel. All registered user will be shown in this panel to Admin. Admin will select any user and update/delete and block

Description	Admin will open the application. Admin will login to the system application by providing the credentials into the required fields. Admin will click on manage click button link. Admin will be shown with a panel where he will be shown with all registered user in the system. Admin will select any of the registered users. He will update the information and submit the data. Client information will be updated and will be saved. For deleting the user information admin will select the user and he will click on the delete button. Client data will be deleted from the system. For blocking, admin select the user and blocked him.
Pre-Conditions	Admin must be logged in.
Post Conditions	User data must be update and deleted respectively.
Author	MC123456789

Functional Requirement title	Manage Fuel Station
Functional Requirement Id	UC-6
Actor	Admin
Actions	Admin will login into the system application. All registered fuel station will be shown in this panel to admin. Admin will select any fuel station and update/ delete/block the candidate.
Description	Admin will open the application. Admin will login to the system application by providing the credentials into the required fields. Admin will click on manage fuel station button link. Admin will be shown with a panel where he will be shown with all registered fuel station in the system. Admin will select any of the registered fuel station. He will update

	the information and submit the data. Fuel station information will be updated and will be saved. For deleting the fuel station information admin will select the user and he will click on the delete button. Fuel station data will be deleted from the system. For blocking, admin select the candidate and blocked him.
Pre-Conditions	Admin must be logged in.
Post Conditions	Fuel Station data must be update and deleted respectively.
Author	MC123456789

Functional Requirement title	Signup
Functional Requirement Id	UC-8
User	User, Fuel Station
Actions	The User/Fuel Station should be able to sign up through a registration form. The registration form should client's basic information and requirement. The basic information about the user/fuel station should have the fields such as Name, Email, Phone, Address, etc.
Description	The user/fuel station should be able to sign up through a registration form. The registration form should get the fuel station/user basic information and requirement. The basic information about the users should have the fields such as Name, Email, Phone, Address, etc. it sends the request to admin for signup. Then admin accept or reject the signup request.
Pre-Conditions	Admin must be logged in.
Post Conditions	User, Fuel Station must be signup.
Author	MC123456789

Functional Requirement title	View Fuel Station
Use case Id	UC-9
User	User
Actions	User will login into the system application. He will click on search Button and find fuel station
Description	Userwill open the application. Client will login to the system application by providing the credentials into the required fields. Client will click on view fuel station button link. Admin will be shown with a panel where he will be shownall fuel station.
Pre-Conditions	User must be logged in.
Post Conditions	View fuel station must be shown
Author	MC123456789

Functional Requirement title	Logout
Functional Requirement Id	UC-13
Actor	User, Fuel Station, Admin
Actions	User can be logout any time.
Description	User can be logout any time.
Pre-Conditions	All Users have an account.
Post Conditions	All users must be login
Exceptions	NA
Author	MC123456789

Adopted Methodology

Though many people may not have even heard of an NFR (non-functional requirements), NFRs are among the most important things to define when building the specification for an e-commerce platform, as most have a direct impact on the experience and satisfaction of your users.

We are all used to defining functional requirements, such as what will happen when a user clicks the “add to basket button” or when they input a search term. These are tangible, easy to quantify and understand, and typically define the behavior of a system based on user interactions.

A non-functional requirement is one which is a little less tangible, or does not specifically define system behavior, but is no less important.

1. Adopted Methodology:

5. 1 Existing Methodologies:

A methodology is composed of one of the software development models used in conjunction with one or more techniques, i.e., methodology = model + technique(s). The techniques such as prototyping, cleanroom, and object-oriented are ways to implement the waterfall, spiral and incremental models. These methods may be combined on a single project. Also, part of a technique may be used without using all aspects of that technique. It means that a project using the spiral model may combine prototyping with object-oriented analysis and design and also use cleanroom testing techniques. Using the METHODOLOGY = MODEL + TECHNIQUE statement, there are much more methodologies than we have time to categorize. Therefore, the remainder of the discussion will deal with models and techniques.

1. Models:

- **Waterfall Model:**

In the earliest days of software development, code was first written and then debugged. There was no proper design or analysis. This first code and then debug approach quickly became less than optimal as complex software systems were required. As the approach to developing complex hardware systems was well understood, it delivered a model for developing software. This model, known as the waterfall, is an approach that emphasizes completing a phase of the development before proceeding to the next phase. In combination with certain phase completions, a baseline is established that "freezes" the products of the development at that point. If a need is recognized to adjust these products, a prescribed change process is followed to make the change. The graphic illustration of these phases in software development looks like the downward flow of a waterfall.

The salient features of the waterfall model are:

- A formal method.
- A type of top-down development.
- Composed of independent phases to be done sequentially.
- Used in varied ways.
 - Steps are combined.
 - There are diverse starting and ending points.

a. Incremental Model:

The incremental model performs the waterfall in overlapping sections attempting to compensate for the length of waterfall model projects by producing usable functionality at an early stage. This may involve a whole straightforward set of requirements that are implemented in a series of small projects. Alternatively, a project using the incremental model may start with general objectives. Then these objectives can be partially redefined as requirements and are implemented, followed by the next portion of the objectives until all objectives are implemented.

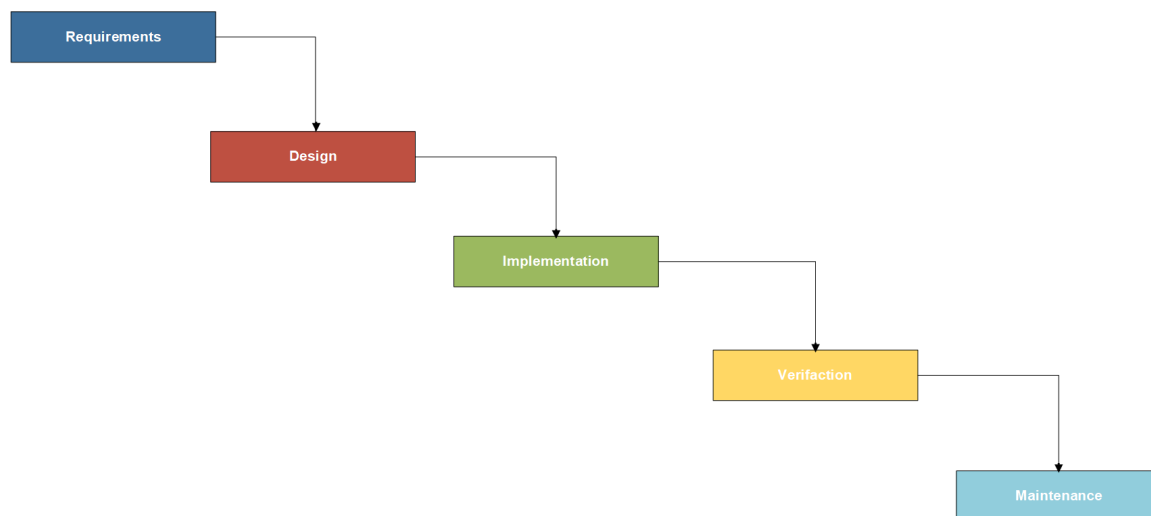


Figure 1 Software Development Models

But, use of general objectives rather than complete requirements can be uncomfortable for management. Because some of the modules will be completed long before others modules,

definite interfaces are required. Also, strict reviews and audits are more difficult to implement on increment models than on a complete system. Finally, there can be an inclination to drive difficult problems to the future to prove early success to management.

b. Spiral Model:

The incremental model can be viewed as a spiral model. The spiral view demonstrates the strength of the incremental model and that is resources can be held constant but the system size rises. The spiral size relates to system size, and the distance between the coils of the spiral indicates resources.

2. Techniques:

a. Prototyping:

Prototyping is the process of building a working mockup of a system.

Strengths of Prototyping:

- Early functionality.
- Provides a process to perfect the requirements definition.
- Provides risk control.
- Documentation focuses on the end product not the evolution of the product.
- Provides a formal specification embodied in an operating replica.

Weaknesses of Prototyping:

- Less applicable to existing systems as compared to original, new development.
- Disliked among conservatives by way of a "quick and dirty" method.
- Suffers from bad documentation.
- Sometimes produces a system with poor performance.
- Inclination for difficult problems to be drive to the future so that the initial potential of the prototype is not met by subsequent products.

b. Cleanroom:

The cleanroom technique attempts to keep contaminants (software bugs) out of the product. The impression is to control cost by spotting bugs as early as possible, when it is less expensive to remove them. In spite of using natural languages like English, more proper notations are used to produce specifications on which all software design and requirements validation is based. Software is intended to execute properly the first time. Testing is done by means of statistical examination to emphasize on the detection of the errors which are most likely to cause operational failures.

Strengths of Cleanroom

- Production of reliable software.
- Cleanroom may be gradually introduced to an organization.

Weaknesses of Cleanroom

- Needs a complete set of requirements.
- Disciplined style may suppress creativity.

c. Object-Oriented:

The object-oriented approach to software development focuses on real-world objects. It is built on the idea that there exists a fundamental human limitation to manage more than seven different objects or concepts at one time.

Strengths of Object-Oriented:

- Owners of a problem can participate in deriving the solution.
- Lower maintenance costs because the object-oriented analysis encourages a complete solution.
- The model expresses the user's view of reality.

Weaknesses of Object-Oriented

- Can be difficult for those with a structured analysis background.

5.2 Adopted Methodology:

For the sake of the current project, VU process model along with prototyping has been chosen. The complete adopted model is shown in the following diagram.

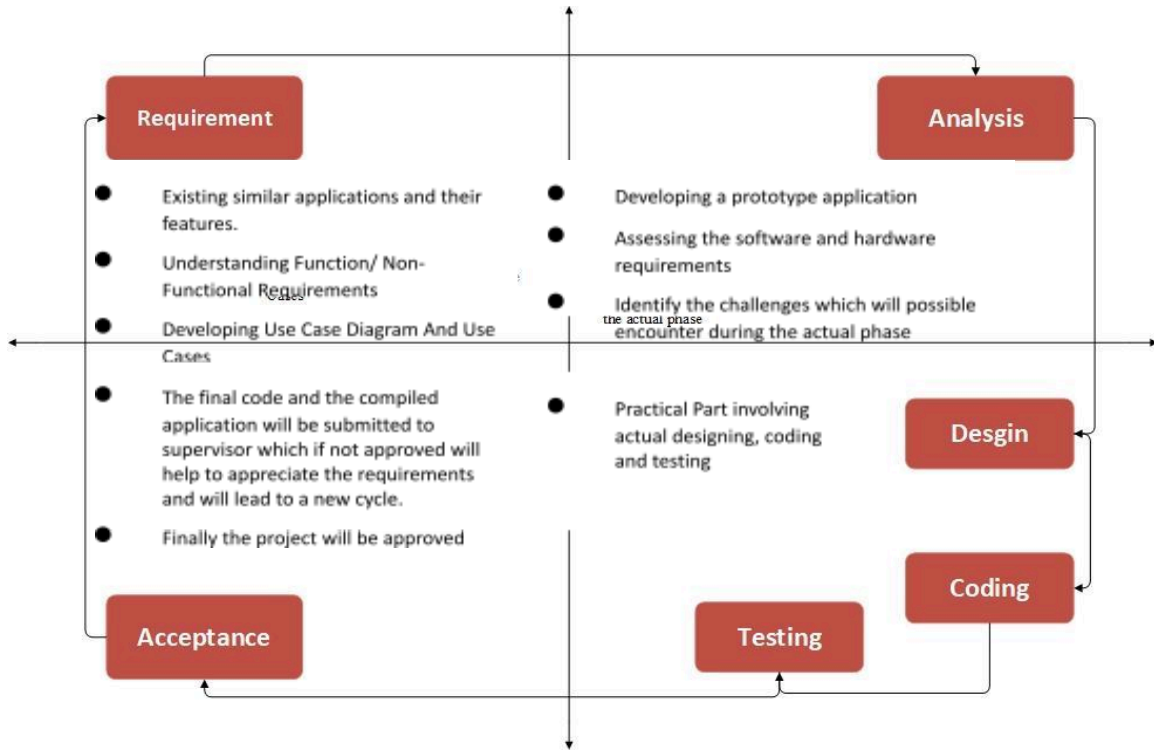


Figure 2 Adopted Model for the current Project

5.3 Reasons for choosing the Methodology:

As we have seen earlier that a methodology is simply a model with a mechanism. Also the review of existing models shows that no single model is perfect. It has been summarized in the following table which describes the strengths and weaknesses of three models. Another model is VU process model proposed by virtual university of Pakistan and is nothing except a combination of waterfall and spiral models. It's evident from this table that where there is weakness in the waterfall model, the alternative spiral model is strong and vice versa so by combining these two models together as VU process model what we have is a unique process model with greater strength and minor weakness.

	Waterfall	Incremental	Boehm Spiral
STRENGTHS			
Allows for work force specialization	X	X	X
Orderliness appeals to management	X	X	X

Can be reported about	X	X	X
Facilitates allocation of resources	X	X	X
Early functionality		X	X
Does not require a complete set of requirements at the onset		X(4)	X
Resources can be held constant		X	
Control costs and risk through prototyping			X
WEAKNESSES			
Requires a complete set of requirements at the onset	X		
Enforcement of non-implementation attitude hampers analyst/designer communications	X		
Beginning with less defined general objectives may be uncomfortable for management		X	X
Requires clean interfaces between modules		X	
Incompatibility with a formal review and audit procedure		X	X
Tendency for difficult problems to be pushed to the future so that the initial promise of the first increment is not met by subsequent products		X	X
(4) The incremental model may be used with a complete set of requirements or with less defined general objectives.			
Table 2: Strengths and Weaknesses of Models.			

Table 1 Comparison of Various Models

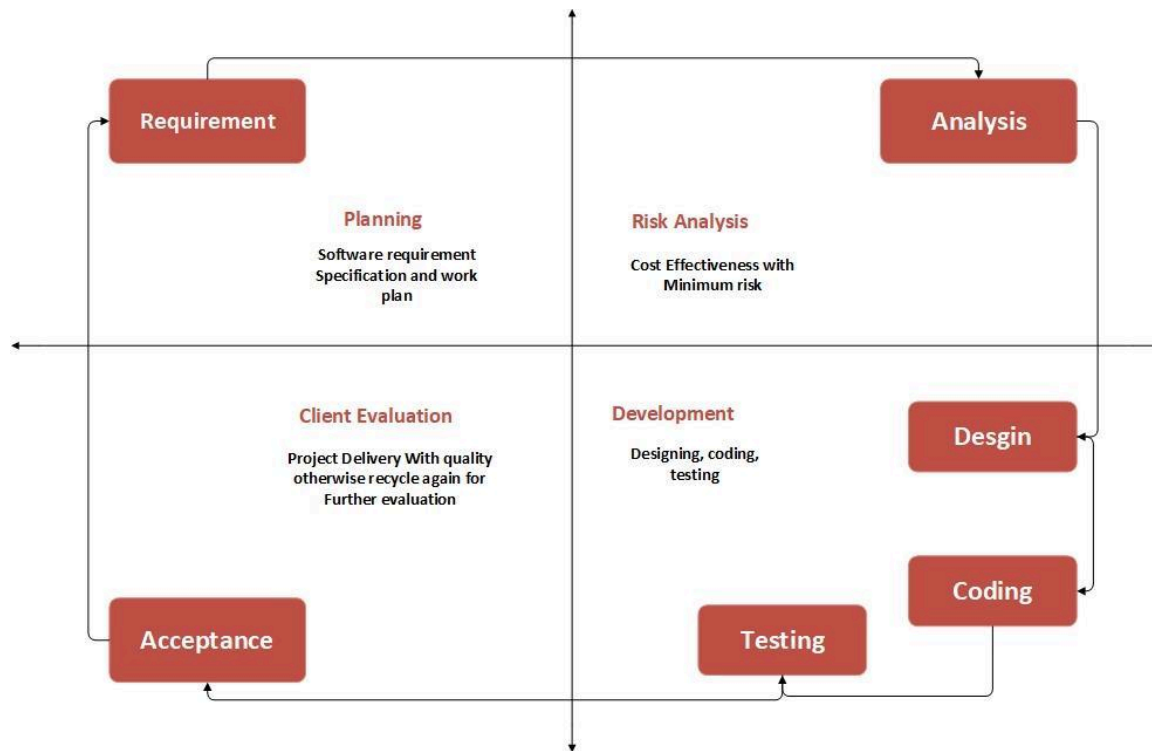


Figure 3 VU Process Model

6. Schedule/Work plan)

