**Higher Nationals**

Internal verification of assessment decisions – BTEC (RQF)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **INTERNAL VERIFICATION – ASSESSMENT DECISIONS** | | | | | | |
| **Programme title** | HND in Computing | | | | | |
| **Assessor** | Ms. Iresha Jayarathne | | **Internal Verifier** |  | | |
| **Unit(s)** | **Unit 35: System Analysis & Design** | | | | | |
| **Assignment title** | **Web based System for “Auto Cars”** | | | | | |
| **Student’s name** | Mohamed Zaheer | | | | | |
| **List which assessment criteria the Assessor has awarded.** | **Pass** | | **Merit** | | **Distinction** | |
|  | |  | |  | |
| **INTERNAL VERIFIER CHECKLIST** | | | | | | |
| **Do the assessment criteria awarded match those shown in the assignment brief?** | | Y/N |  | | | |
| **Is the Pass/Merit/Distinction grade awarded justified by the assessor’s comments on the student work?** | | Y/N |  | | | |
| **Has the work been assessed accurately?** | | Y/N |  | | | |
| **Is the feedback to the student:**  Give details:   * Constructive? * Linked to relevant assessment criteria? * Identifying opportunities for improved performance? * Agreeing actions? | | Y/N  Y/N  Y/N  Y/N |  | | | |
| **Does the assessment decision need amending?** | | Y/N |  | | | |
| **Assessor signature** | |  | | **Date** | |  |
| **Internal Verifier signature** | |  | | **Date** | |  |
| **Programme Leader signature** (if required) | |  | | **Date** | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Confirm action completed** | | | |
| **Remedial action taken**  Give details: |  | | |
| **Assessor signature** |  | **Date** |  |
| **Internal Verifier signature** |  | **Date** |  |
| **Programme Leader signature** (if required) |  | **Date** |  |

Higher Nationals - Summative Assignment Feedback Form

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name/ID** | Mohamed Zaheer / E171743 | | |
| **Unit Title** | **Unit 35: System Analysis & Design** | | |
| **Assignment Number** | **1** | **Assessor** | Ms. Iresha Jayarathne |
| **Submission Date** | 2024/04/03 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessor Feedback:**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **LO1 Evaluate the strengths and weaknesses of the traditional and agile systems analysis methodologies** | | | | | | | **Pass, Merit & Distinction Descripts** | **P1** | **M1** | D1 |  |  |  | | **LO2 Produce a feasibility study for a system for system to be developed to solve a business-related problem** | | | | | | | **Pass, Merit & Distinction Descripts** | **P2** | **M2** |  |  |  |  | | **LO3 Assess systems analysis methodologies to effectively solve business-related problems** | | | | | | | **Pass, Merit & Distinction Descripts** | **P3** | **M3** | **D2** |  |  |  | | **LO4 Design the system to meet user and system requirements.** | | | | | | | **Pass, Merit & Distinction Descripts** | **P4** | **M4** |  |  |  |  | |  | |

|  |  |  |
| --- | --- | --- |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Internal Verifier’s Comments:** | | |
| **Signature & Date:** | | |

\* Please note that grade decisions are provisional. They are only confirmed once internal and external moderation has taken place and grades decisions have been agreed at the assessment board.

**Pearson Higher Nationals in**

**Computing**

Unit 35: Systems Analysis & Design

Assignment 01

**General Guidelines**

1. A Cover page or title page – You should always attach a title page to your assignment. Use previous page as your cover sheet and make sure all the details are accurately filled.
2. Attach this brief as the first section of your assignment.
3. All the assignments should be prepared using a word processing software.
4. All the assignments should be printed on A4 sized papers. Use single side printing.
5. Allow 1” for top, bottom , right margins and 1.25” for the left margin of each page.

**Word Processing Rules**

1. The font size should be **12** point, and should be in the style of **Time New Roman**.
2. **Use 1.5 line spacing**. Left justify all paragraphs.
3. Ensure that all the headings are consistent in terms of the font size and font style.
4. Use **footer function in the word processor to insert Your Name, Subject, Assignment No, and Page Number on each pag**e. This is useful if individual sheets become detached for any reason.
5. Use word processing application spell check and grammar check function to help editing your assignment.

**Important Points:**

1. It is strictly prohibited to use textboxes to add texts in the assignments, except for compulsory information. eg: Figures, tables of comparison etc. Adding text boxes in the body except for the before mentioned compulsory information will result in rejection of your work.
2. Carefully check the hand in date and the instructions given in the assignment. Late submissions will not be accepted.
3. Ensure that you give yourself enough time to complete the assignment by the due date.
4. Excuses of any nature will not be accepted for failure to hand in the work on time.
5. You must take responsibility for managing your own time effectively.
6. If you are unable to hand in your assignment on time and have valid reasons such as illness, you may apply (in writing) for an extension.
7. Failure to achieve at least PASS criteria will result in a REFERRAL grade .
8. Non-submission of work without valid reasons will lead to an automatic RE FERRAL. You will then be asked to complete an alternative assignment.
9. If you use other people’s work or ideas in your assignment, reference them properly using HARVARD referencing system to avoid plagiarism. You have to provide both in-text citation and a reference list.
10. If you are proven to be guilty of plagiarism or any academic misconduct, your grade could be reduced to A REFERRAL or at worst you could be expelled from the course.

**Student Declaration**

I hereby, declare that I know what plagiarism entails, namely, to use another’s work and to present it as my own without attributing the sources in the correct form. I further understand what it means to copy another’s work.

1. I know that plagiarism is a punishable offence because it constitutes theft.
2. I understand the plagiarism and copying policy of Edexcel UK.
3. I know what the consequences will be if I plagiarise or copy another’s work in any of the assignments for this program.
4. I declare therefore that all work presented by me for every aspect of my program, will be my own, and where I have made use of another’s work, I will attribute the source in the correct way.
5. I acknowledge that the attachment of this document signed or not, constitutes a binding agreement between myself and Pearson , UK.
6. I understand that my assignment will not be considered as submitted if this document is not attached to the assignment.

**Mhdzaheer2003@gmail.com**

**Student’s Signature: Date:**

**(*Provide E-mail ID*) (*Provide Submission Date*)**

**Higher National Diploma in Computing**

Assignment Brief

|  |  |
| --- | --- |
| Student Name /ID Number | Mohamed Zaheer /E171743 |
| **Unit Number and Title** | **Unit 35: Systems Analysis & Design** |
| Academic Year | 2022/23 |
| Unit Tutor |  |
| **Assignment Title** |  |
| Issue Date | 2024/02/05 |
| Submission Date | 2024/04/03 |
| IV Name & Date |  |

|  |  |
| --- | --- |
| **Submission format** | |
| The submission should be in the form of an individual written report written in a concise, formal business style using single spacing and font size 12. You are required to make use of headings, paragraphs and subsections as appropriate, and all work must be supported with research and referenced Please provide in-test citations, reference list and bibliography using Harvard referencing system. Please also provide a bibliography using the Harvard referencing system.  The recommended word limit is not less than 5000 words, although you will not be penalised for exceeding the total word limit. | |
| **Unit Learning Outcomes:** | |
| LO1 Evaluate the strengths and weaknesses of the traditional and agile systems analysis methodologies.  LO2 Produce a feasibility study for a system to be developed to solve a business-related problem.  LO3 Assess systems analysis methodologies to effectively solve business-related problems.  LO4 Design the system to meet user and system requirements. | |
| **Assignment Brief and Guidance:** | |
| \*Please note that assignment guidance is for reference only and should be more specific in detail to meet customized needs.  **Assignment brief**  **Case study**  “Auto Cars” is a taxi service in Sri Lanka which has been in operation for several decades, serving the customers in city and urban areas. As a result of the increased use of information technology such as mobile phones and the internet, “Auto Cars” decided to launch a web-based system that enables customers to reserve vehicles conveniently. This also helps the company to gain a competitive advantage.  The main system features of the web-based system are required for “Auto Cars” to reserve taxis for customers are:   1. Customers and drivers can use the online system to register. They will receive an email with their username and password once they have registered. 2. Customers can book taxis via the website. When they reserve a vehicle, they will receive an SMS on their phone with the driver's contact information and vehicle details. 3. Customers can rate drivers based on their experiences with them on trips. 4. In addition, the company employs a phone operator who manually reserves drivers from the system. This is the same as a customer reserving a driver, except there is no logged in customer in the application. 5. When the operator makes a taxi reservation, she enters the customer's phone number into the system, and the customer receives an SMS. 6. The driver closest to the customer must be assigned by the application.  When a customer or the operator reserves a vehicle, the system should display a list of available drivers. From this list, the customer can then select the best driver. 7. The driver’s location is manually (by the driver) entered into the web-based system or via the telephone operator. Similarly, when reserving a taxi, the customer's location is specified (by the customer or through the operator).   Assume that you are the newly appointed system analyst for “Auto Cars” and assigned to work on this project. Produce a professional report including the answers to the following tasks.  **Activity 01**  Discuss and critically evaluate the strengths and weaknesses of the traditional and agile systems analysis methodologies. Compare the strengths and weaknesses of the two methodologies by referring to the proposed web system for “Auto Cars”.  **Activity 2**  Conduct the feasibility study for the web-based system for “Auto Cars” and produce a feasibility report. Further, evaluate the importance of the feasibility criteria used to investigate the feasibility of the proposed system.  **Activity 3**  Review the user and system requirements for the web-based system suggested in the scenario using a suitable software development methodology and asses the effectiveness of the methodology selected. Justify the chosen methodology for the web-based system with relevant examples.  **Activity 4**  Design a system specification for the web-based system to meet end user system requirements and assess the effectiveness of your design and the methodology used with reference to how it meets the user requirements.  Your system design specification should include,   * Data Flow diagrams (context diagram, Level 0, level 1 , level 2 DFDs)/ Use case diagram * Entity Relationship diagram (ERD) * Flow charts * Prototypes to illustrate system interfaces and functions. * Release plan and tools | |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Grading Criteria** | **Achieved** | **Feedback** |
| **LO1** Evaluate the strengths and weaknesses of the traditional and agile systems analysis methodologies. |  |  |
| **P1** Discuss the strengths and weaknesses of the traditional and agile systems analysis methodologies. | 22-39 |  |
| **M1** Compare and contrast the strengths and weaknesses of the traditional and agile systems analysis methodologies with reference to a specific business-related problem. |  |  |
| **LO2** Produce a feasibility study for a system to be developed to solve a business-related problem. |  |  |
| **P2** Produce a feasibility study for a system for a business related problem. |  |  |
| **M2** Evaluate the importance of the feasibility criteria in the systems investigation for the business related problem. |  |  |
| **LO1 & LO2**  **D1** Critically evaluate the strengths and weaknesses of the traditional and agile systems methodologies and feasibility study in solving a specific business-related problem. |  |  |
| **LO3** Assess systems analysis methodologies to effectively solve business-related problems |  |  |
| **P3** Review a system using a suitable methodology for a business-related problem. |  |  |
| **M3** Asses the effectiveness of the methodology used in providing a solution for a given business context. |  |  |
| **LO4** Design a system to meet user and system  Requirements |  |  |
| **P4** Design a fully functional system to meet user and system requirements for the business-related  problem. |  |  |
| **M4** Assess the effectiveness of the system design, with reference to the methodology used and how the design meets user and system requirements. |  |  |
| **LO3 & 4**  **D2** Justify the choice of the methodology used in  the context of the business problem. |  |  |

# Acknowledgement

I am grateful for the divine guidance and strength that enabled me to complete this assignment. Heartfelt thanks to my lecturer for their guidance and to my family and friends for their unwavering support. May God bless them with success and honor.

Thankyou!

# Table of Contents

[Acknowledgement 15](#_Toc163080385)

[Table of Contents 16](#_Toc163080386)

[Figures 21](#_Toc163080387)

[Tables 22](#_Toc163080388)

[Activity 01 23](#_Toc163080389)

[1.1. Traditional Methodology 23](#_Toc163080390)

[1.2. Types of Traditional Methodology 23](#_Toc163080391)

[1.2.1. Waterfall Model 23](#_Toc163080392)

[1.2.2. Spiral Model 26](#_Toc163080393)

[1.2.3. Prototyping Model 28](#_Toc163080394)

[1.3. Strengths and Weaknesses of Traditional Methodology 30](#_Toc163080395)

[1.3.1. Strengths of Traditional Methodology 30](#_Toc163080396)

[1.3.2. Weaknesses of Traditional Methodology 31](#_Toc163080397)

[1.4. Agile Methodology 31](#_Toc163080398)

[1.5. Types of Agile Methodology 32](#_Toc163080399)

[1.5.1. Scrum 32](#_Toc163080400)

[1.5.2. Extreme programming (XP) 34](#_Toc163080401)

[1.5.3. Kanban 35](#_Toc163080402)

[1.5.4. Lean development 36](#_Toc163080403)

[1.6. Strengths and Weaknesses of Agile Methodology 39](#_Toc163080404)

[1.6.1. Strengths of Agile Methodology 39](#_Toc163080405)

[1.6.2. Weaknesses of Agile Methodology 40](#_Toc163080406)

[1.7. Comparison of Strengths and Weaknesses of Traditional vs. Agile System Analysis Methodologies 41](#_Toc163080407)

[1.8. Similarities Between Traditional vs. Agile System Analysis Methodologies 42](#_Toc163080408)

[1.9. Comparison in the Context of “Auto Cars” Web System 44](#_Toc163080409)

[1.10. Recommended Approach 46](#_Toc163080410)

[Activity 02 48](#_Toc163080411)

[2.1. Feasibility Study 49](#_Toc163080412)

[2.1.1. Importance of Feasibility Study 49](#_Toc163080413)

[2.1.2. Types Of Feasibility Studies 50](#_Toc163080414)

[2.2. Feasibility Study Report for the Web-Based System of "Auto Cars" 51](#_Toc163080415)

[Project Scope 51](#_Toc163080416)

[Current Analysis 51](#_Toc163080417)

[Requirements 52](#_Toc163080418)

[The Approach 52](#_Toc163080419)

[Evaluation 53](#_Toc163080420)

[Review 53](#_Toc163080421)

[2.3. Feasibility study for “Auto Car” 54](#_Toc163080422)

[2.3.1. Technical Feasibility 54](#_Toc163080423)

[2.3.2. Economic Feasibility 54](#_Toc163080424)

[2.3.3. Legal Feasibility 56](#_Toc163080425)

[2.3.4. Operational Feasibility 56](#_Toc163080426)

[2.3.5. Scheduling Feasibility 57](#_Toc163080427)

[2.4. Importance of Feasibility Criteria in Evaluating the Proposed System for "Auto Cars" 58](#_Toc163080428)

[- Financial Feasibility 58](#_Toc163080429)

[- Technical Feasibility 59](#_Toc163080430)

[Operational Feasibility 59](#_Toc163080431)

[Legal and Regulatory Feasibility 60](#_Toc163080432)

[Schedule Feasibility 60](#_Toc163080433)

[2.5. Strength and weakness of Traditional and Agile Methodology 61](#_Toc163080434)

[2.5.1. Traditional Methodologies Strengths 61](#_Toc163080435)

[2.5.2. Traditional Methodologies Weaknesses 62](#_Toc163080436)

[2.5.3. Agile Methodologies Strengths 63](#_Toc163080437)

[2.5.4. Agile Methodologies Weakness 64](#_Toc163080438)

[2.6. Strength and weakness of Feasibility study 64](#_Toc163080439)

[2.7. Critical Evaluation 66](#_Toc163080440)

[Activity 03 67](#_Toc163080441)

[3.1. User requirements 67](#_Toc163080442)

[3.1.1. User Requirements for “Auto Cars” Web system. 67](#_Toc163080443)

[3.2. Organizational Requirements 68](#_Toc163080444)

[3.2.1. Organizational Requirements for “Auto Cars” Web system. 68](#_Toc163080445)

[3.3. System Requirements 68](#_Toc163080446)

[3.3.1. System Requirements for “Auto Cars” Web system. 69](#_Toc163080447)

[3.4. Functional Requirements 69](#_Toc163080448)

[3.4.1. Functional Requirements for “Auto Cars” Web system. 70](#_Toc163080449)

[3.5. Non-Functional Requirements 70](#_Toc163080450)

[3.5.1. Non-Functional Requirements for “Auto Cars” Web system. 70](#_Toc163080451)

[3.6. Analysis of the Most Suitable Methodology for “Auto Cars” 71](#_Toc163080452)

[3.7. Reason for choosing scrum methodology for “Auto Cars “Web based System. 73](#_Toc163080453)

[3.8. Advantages and Disadvantages of Scrum Methodology 74](#_Toc163080454)

[3.9. Justification of the Chosen Methodology for the Auto Cars 74](#_Toc163080455)

[3.10. Effectiveness and suitability of the chosen methodology 75](#_Toc163080456)

[Activity 04 77](#_Toc163080457)

[4.1. Effectiveness of the design and the methodology used with reference to how it meets the user requirements. 77](#_Toc163080458)

[System Design Specification for “Auto Cars” Web-Based Taxi Reservation System 78](#_Toc163080459)

[4.2. System Specification Requirements (SRS) 79](#_Toc163080460)

[4.2.1 System Specification for Auto Cars 80](#_Toc163080461)

[Introduction 80](#_Toc163080462)

[Objectives 80](#_Toc163080463)

[Project Scope 80](#_Toc163080464)

[Overall Description 80](#_Toc163080465)

[Product Function 80](#_Toc163080466)

[User Characteristics 81](#_Toc163080467)

[Constraints 81](#_Toc163080468)

[Assumptions and Dependencies 81](#_Toc163080469)

[Specific Requirements 81](#_Toc163080470)

[Application Maintenance 82](#_Toc163080471)

[4.3. Data Flow Diagram (DFD) 83](#_Toc163080472)

[Dataflow diagram for Auto cars 84](#_Toc163080473)

[4.4. Use case Diagram. 85](#_Toc163080474)

[Use case Diagram for Auto cars. 86](#_Toc163080475)

[4.5. Flowchart Diagram 86](#_Toc163080476)

[Flowchart Diagram for Auto cars 87](#_Toc163080477)

[4.6. Entity Relationship Diagram 88](#_Toc163080478)

[Database Diagram For Auto cars 89](#_Toc163080479)

[ER Diagram for Auto Cars 90](#_Toc163080480)

[4.7. Prototypes 91](#_Toc163080481)

[Login Page 91](#_Toc163080482)

[Customer Registration 92](#_Toc163080483)

[Drivers Registration 92](#_Toc163080484)

[Drivers Home Page 93](#_Toc163080485)

[Customer Home Page 93](#_Toc163080486)

[Taxi Booking Page 94](#_Toc163080487)

[Rating and Feedback Page 94](#_Toc163080488)

[Recent trip page 95](#_Toc163080489)

[Settings Page 95](#_Toc163080490)

[4.8. Release Plans and Tools 96](#_Toc163080491)

[Release Plan Gant Chart 96](#_Toc163080492)

[Tools 96](#_Toc163080493)

[4.9. Assessment of Effectiveness 97](#_Toc163080494)

[4.10. Justification of Methodology in the Business Problem Context 98](#_Toc163080495)

[References 99](#_Toc163080496)

# Figures

[Figure 1 : Traditional Project Management (Kashyap, 2024) 23](#_Toc163080345)

[Figure 2: Waterfall Model 24](#_Toc163080346)

[Figure 3: Spiral Model (sitesbay., n.d.) 26](#_Toc163080347)

[Figure 4: Prototyping Model (naimishsahu08, 2023) 28](#_Toc163080348)

[Figure 5: Prototyping model phases (Martin, 2024) 29](#_Toc163080349)

[Figure 6: Scrum Framework (mijacobs, et al., 2022) 32](#_Toc163080350)

[Figure 7: Extreme programming methodology (lucidchart, n.d.) 34](#_Toc163080351)

[Figure 8: Kanban Board (aadityasharma0115, 2023) 35](#_Toc163080352)

[Figure 9: Lean Development principles (Staff, 2024) 37](#_Toc163080353)

[Figure 10: Feasibility study. 49](#_Toc163080354)

[Figure 11: Types of Feasibility study. 51](#_Toc163080355)

[Figure 12: User Requirements 67](#_Toc163080356)

[Figure 13: System Requirements 69](#_Toc163080357)

[Figure 14: Functional Requirements 70](#_Toc163080358)

[Figure 15: Data Flow Diagram Levelo 0 84](#_Toc163080359)

[Figure 16: Data Flow Diagram Level 1 84](#_Toc163080360)

[Figure 17: DFD Diagram Level 2 85](#_Toc163080361)

[Figure 18: Use case Diagram for “Auto car” (Author Developed) 86](#_Toc163080362)

[Figure 19: Flowchart Diagram (Author Developed) 87](#_Toc163080363)

[Figure 20: Flowchart diagram for auto cars (Author Developed) 88](#_Toc163080364)

[Figure 21: Auto Cars For Database Diagram 89](#_Toc163080365)

[Figure 22: ER Diagram 90](#_Toc163080366)

[Figure 23: Login Page (Author Developed) 91](#_Toc163080367)

[Figure 24: Customer Registration (Author Developed) 92](#_Toc163080368)

[Figure 25: Drivers Registration (Author Developed) 92](#_Toc163080369)

[Figure 26: Drivers Home Page 93](#_Toc163080370)

[Figure 27: Customer Home Page (Author Developed) 93](#_Toc163080371)

[Figure 28: Taxi booking page (Author Developed) 94](#_Toc163080372)

[Figure 29: Rating and Feedback Page (Author Developed) 94](#_Toc163080373)

[Figure 30: Recent Trip page (Author Developed) 95](#_Toc163080374)

[Figure 31: Settings Page (Author Developed) 95](#_Toc163080375)

# Tables

[Table 1: Comparison of Strengths and Weaknesses of Traditional vs. Agile System Analysis Methodologies 41](#_Toc163080376)

[Table 2: Similarities Between Traditional vs. Agile System Analysis Methodologies 42](#_Toc163080377)

[Table 3:Comparison in the Context of “Auto Cars” Web System 44](#_Toc163080378)

[Table 4: Cost for Auto Cars 55](#_Toc163080379)

[Table 5: Project Timeline for Auto Cars 58](#_Toc163080380)

[Table 6:Strength and weakness of Feasibility study 64](#_Toc163080381)

[Table 7: Advantages and Disadvantages of Scrum Methodology 74](#_Toc163080382)

[Table 8: Application Maintenance (Author Developed) 82](#_Toc163080383)

[Table 9: Gant Chart 96](#_Toc163080384)

# Activity 01

## Traditional Methodology

Traditional project management is a well-established methodology that follows a sequential process: starting with initiation, then moving on to planning, execution, monitoring, and finally closure. This approach places importance on linear workflows, thorough documentation, upfront planning, and setting priorities. In traditional project management, time and budget are typically considered flexible, while requirements are seen as fixed, which can lead to challenges with staying within budget and meeting deadlines. The PMBOK® standard outlines specific tools and techniques that project managers adhere to at each stage of the process. Ols and techniques that project managers follow at each step. (Kashyap, 2024)

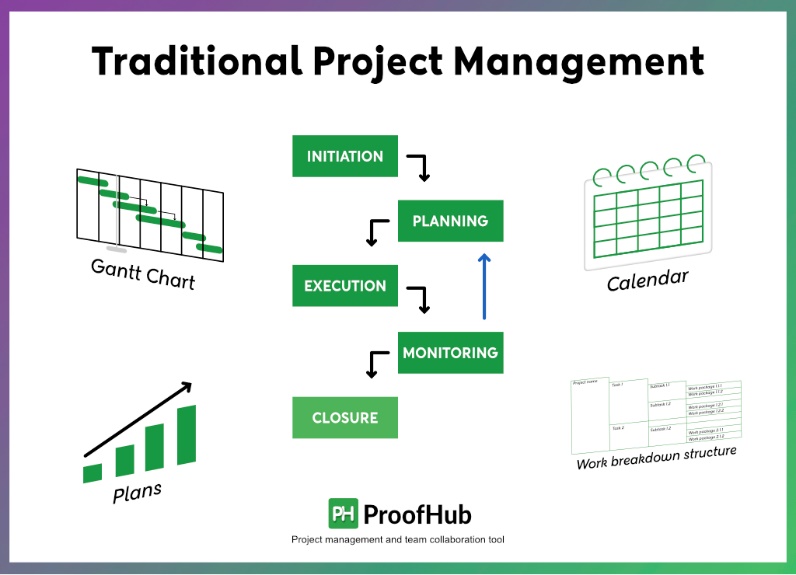


Figure 1 : Traditional Project Management (Kashyap, 2024)

## Types of Traditional Methodology

### Waterfall Model

The waterfall model is a structured approach to software development where each phase flows sequentially, akin to water cascading over a cliff. It involves clear-cut stages with specific objectives, and once a phase is complete, it's not revisited. Initially introduced by Dr. Winston W. Royce in a 1970 paper based on his work with satellite software, although he didn't explicitly name it as such. Despite its origins in software engineering, the waterfall model is also applied in other fields like industrial design and serves as a foundational methodology in project management for complex projects.

(Ben Lutkevich, 2022)



Figure 2: Waterfall Model

#### Phases of Waterfall Model

**- Requirement Gathering Phase**

Requirement Gathering is First Phase of Waterfall model. in This phase involves identifying and documenting all the functional and non-functional requirements of the software. The requirements are collected systematically, including interviews, surveys and workshops conducted with relevant stakeholders.

**- System Design Phase**

System Design is Second phase of waterfall model. In this phase the system architect creates an elaborate design for the software system. The design includes components such as the software architecture, data structures and user interfaces.

**- Implementation Phase**

Implementation is Third phase of Waterfall Model. This phase involves implementing the software by writing the necessary code by the established system design. The program undergoes a modular development approach, dividing it into smaller components that merge to form the final product.

**- Testing Phase**

Testing is the Fourth Phase of Waterfall model. During this phase the software is testing procedures to comply with all defined requirements. The testing process includes unit testing, integration testing and system testing.

**- Deployment Phase**

Deployment is Fifth phase of Waterfall model. During this phase, the software is deployed to the production environment. The process includes installing the software, its configuration and the training of the users.

**- Maintenance Phase**

Maintenance is Sixth phase of Waterfall model. During this phase, the software is a maintenance activity to fix identified bugs and implement new features. The aforementioned phase is characterized by its continuous definition, since it extends over the whole life of the software.

#### Advantages of Waterfall model

* Waterfall model is simple and easy to understand and use.
* Waterfall model easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
* Waterfall model phases are processed and completed one at a time. Phases do not overlap.
* The Waterfall model works well for smaller projects where requirements are clearly defined and very well understood.

#### Disadvantages of Waterfall model

* Once an application is in the Implementation phase, it is very difficult to go back and change something that was not well-thought out in the concept stage.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.

(Anon., n.d.)

### Spiral Model

The Spiral Model is a method for developing software in a structured and iterative way. It’s visualized as a spiral with multiple loops, and the number of loops can differ for each project. Each loop represents a phase of the software development process.

(Anon., 2023)

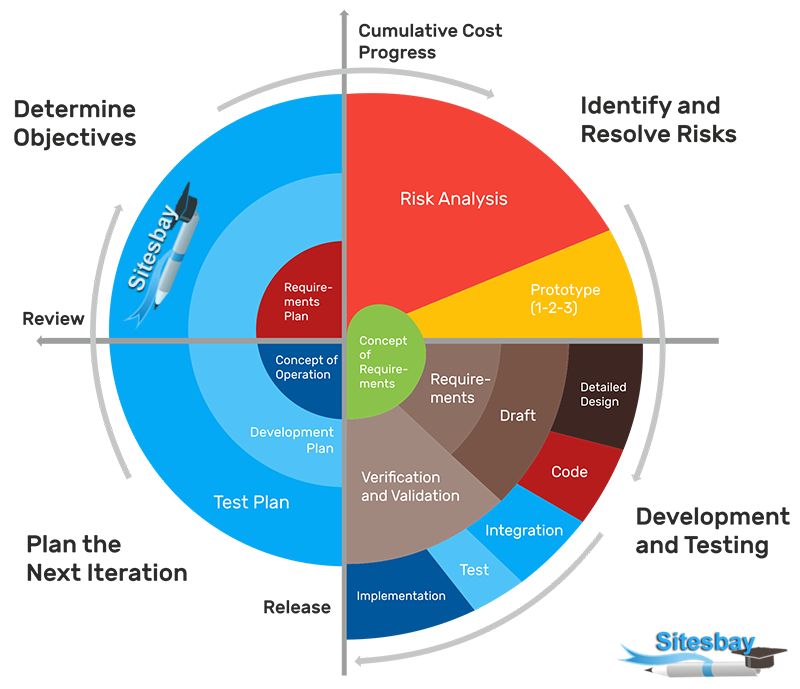


Figure 3: Spiral Model (sitesbay., n.d.)

#### Phases of Spiral Model

* **Determining objectives**

During this phase, the focus is on gathering and analyzing requirements. Objectives are established based on these requirements, and various alternative solutions are proposed.

* **Identify and reduce risk.**

In this phase, all proposed solutions are scrutinized, and any potential risks are identified, analyzed, and addressed.

* **Development and testing**

This phase involves implementing the different features of the software. Once implemented, thorough testing is conducted to verify these features.

* **Planning for the next iteration**

During this phase, the software is evaluated by the customer. Risk identification and monitoring, such as cost overruns or schedule delays, are also carried out. Subsequently, planning for the next phase commences.

(Rana, 2021)

#### Advantages of Spiral Model

* Risk handling ability.
* Implementation of change request
* Large and complex
* Suitable for large complex project

#### Disadvantages of Spiral Model

* Not suitable for a simpler and smaller project
* Difficult in time and cost estimation
* More documentation

### Prototyping Model

The Prototyping Model is a software development approach where a prototype is constructed, evaluated and refined until a satisfactory version is attained. This iterative process serves as a foundation for creating the final software or system. It's particularly effective when project requirements are unclear, involving continuous feedback and adjustments between developers and clients. (Martin, 2024)

A diagram of a software development process

Description automatically generated

Figure 4: Prototyping Model (naimishsahu08, 2023)

#### Phases of Prototyping Model

* **Requirement gathering and Analysis.**

Requirement analysis the initial step developing a prototyping model. In this phase, the specific needs of the system are carefully outlined. It involves interviewing system users to understand their expectations and requirements for the system.

* **Quick Design**

The second phase, there could be a preliminary or quick design step. Here, the fundamental structure of the system is established, though not in its entirety. This design offers users a swift glimpse into the system and facilitates prototype development.

* **Build a Prototype**

this phase, a actual prototype is created to reinforce the insights gathered during the quick design phase. This prototype serves as a small-scale, functional model of the desired system.

* **Initial user evaluation**

During this phase, the client is presented with the proposed system for preliminary testing. It's essential to assess the strengths and weaknesses of the performance model. Feedback and suggestions from the client are collected and relayed to the developer.

* **Refining prototype**

If the client is not satisfied with the current model, there's a need to enhance it based on user feedback and suggestions. Once the user is content with the updated model, a final system is developed based on the approved prototype.

* **Implement Product and Maintain**

After thorough testing, the final system is deployed for production use. Regular maintenance checks are performed to minimize downtime and prevent major issues.

(naimishsahu08, 2023)

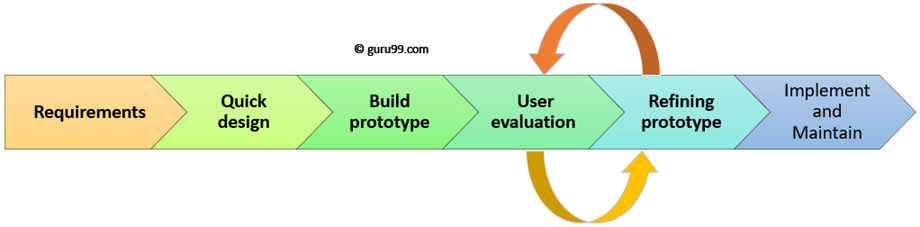


Figure 5: Prototyping model phases (Martin, 2024)

#### Advantages of Prototyping model

* Increased user satisfaction
* Reduced risk
* Enhanced communication
* Early feedback
* Improved requirement definition

#### Disadvantages of Prototyping model

* Focus on functionality over design.
* Scope creep
* Incomplete documentation
* Cost
* Unrealistic expectation

## Strengths and Weaknesses of Traditional Methodology

### Strengths of Traditional Methodology

* **Structured Approach**

Traditional methodologies, such as Waterfall, follow a clear sequence of steps (such as planning, execution, etc.), making it straightforward for teams, particularly on smaller projects.

* **Efficiency** **with Defined Requirements**

When project requirements are well-established upfront, traditional methods enable efficient project completion with a strong emphasis on quality control.

* **Effective Project Management**

A designated project manager oversees the entire process, ensuring adherence to plans and addressing risks. Centralized control is beneficial for complex projects requiring strict scope management.

* **Thorough Documentation**

Traditional methodologies promote comprehensive documentation at each stage, providing a clear record of decisions and progress for future reference and training purposes.

* **Predictable Timelines**

By following a structured approach, traditional methods offer more predictable project timelines, which is vital for projects with tight deadlines.

### Weaknesses of Traditional Methodology

* **Limited Adaptability**

Traditional methods struggle to accommodate changes in requirements. Revisiting completed phases can be cumbersome and time-consuming.

* **Slow Response to Changes**

Limited customer involvement in the initial stages makes it challenging to incorporate late-stage feedback, often resulting in delays and rework.

* **Innovation Constraints**

The rigid structure of traditional methodologies can inhibit creativity and hinder the exploration of new ideas during the development process.

* **Potential for Micromanagement**

Emphasizing control may lead to micromanagement, dampening team morale and impeding their ability to work autonomously.

* **Unrealistic Expectations**

Assuming all requirements are perfectly defined at the outset can be unrealistic for many projects, potentially leading to unforeseen challenges during implementation.

## Agile Methodology

The agile approach is a straightforward method for transforming ideas and diverse requirements into functional software solutions. It's an iterative and incremental approach to software development, focusing on continuous planning, understanding, improvement, team collaboration, development and delivery. Agile breaks down into individual models for teams, offering flexibility to adapt to changes.

agile process, customers define the end uses and problems the final product aims to solve. It's driven by the principles of delivering value and involving stakeholders, helping project teams understand and address customer expectations and requirements. (It Service, 2022)

## Types of Agile Methodology

### Scrum

Scrum isn't a methodology as commonly thought it's a simple framework employing empirical methods akin to the scientific approach. It aids individuals, software teams, and organizations in adapting and creatively solving complex problems efficiently. This adaptability facilitates the delivery of high-value products like software applications with timely delivery and cutting-edge features.

(Corrales, 2022)

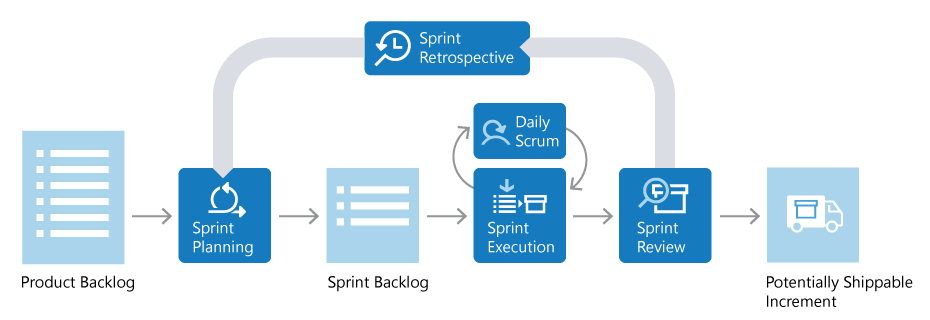


Figure 6: Scrum Framework (mijacobs, et al., 2022)

**Scrum Team Roles**

* **Product Owner**

The product owner is accountable for determining what the team constructs and the reasons behind it. They are also responsible for maintaining an updated and prioritized backlog of tasks.

* **Scrum Master**

The Scrum master ensures that the team adheres to the Scrum process, constantly seeking opportunities for improvement, and resolving any obstacles or issues encountered during the sprint. They play a role akin to a coach, team member, and motivator.

* **Development Team**

The development team members are tasked with physically constructing the product. They take ownership of the product's engineering and ensure its quality.

(mijacobs, et al., 2022)

**Product Backlog**

The product backlog is essentially a prioritized to-do list managed by the product owner. They're responsible for updating it with new tasks, changing priorities, and ensuring that the top items are ready for the team to tackle.

**Sprint Planning**

sprint planning, the team selects items from the backlog to work on during the upcoming sprint, considering both priority and their capacity. These chosen tasks form the sprint backlog, which the team aims to complete by the end of the sprint. Sometimes, each item in the sprint backlog is further broken down into smaller tasks. Once the team agrees that the sprint backlog is achievable, they kick off the sprint.

**Execute the Sprint**

Once the sprint begins, the team gets to work on the items in the sprint backlog. Scrum doesn't prescribe a specific way for the team to work; they decide how to manage their own tasks.

There's also a daily Scrum, or daily standup, where team members meet for a brief 15-minute update. Each member shares what they've accomplished since the last meeting, what they plan to do next, and any obstacles they're facing. This quick check-in helps keep everyone on track and aware of each other's progress.

#### Advantages of Scrum

* Adaptability and Flexibility
* Faster Time to Market
* Reduced Costs
* Improved Quality and Customer Satisfaction
* Increased Transparency and Team Collaboration
* Boosted Team Morale and Motivation

#### Disadvantages of Scrum

* Requires Experienced Team
* Limited Upfront Planning:
* Reliance on Ongoing Training
* Potential for Scope Creep
* Not Ideal for Large Projects

### Extreme programming (XP)

Extreme Programming (XP) stands out as a key Agile software development framework, aimed at enhancing software quality and responsiveness to customer needs. It emphasizes pushing proven best practices from previous programming projects to their limits.

(PAL, 2024)

A diagram of a flowchart

Description automatically generated

Figure 7: Extreme programming methodology (lucidchart, n.d.)

#### Advantages of Extreme Programming

Faster Development and Lower cost

Improved Communication and collaboration

High-quality software

adaptability to changing requirements.

#### Disadvantages of Extreme Programming

Limited focus on Design

May not suitable for all project

Not ideal for geographically dispersed teams

Requires high level of disciplines.

### Kanban

Kanban, a popular software development approach alongside Scrum, originated in the 1940s at Toyota for manufacturing, but was adapted for software development after the Agile Manifesto in 2001. It revolves around a Kanban Board, essentially a visual representation of the project's workflow, divided into columns representing stages like To-Do, In-Progress, Validation, and Completed.

(aadityasharma0115, 2023)

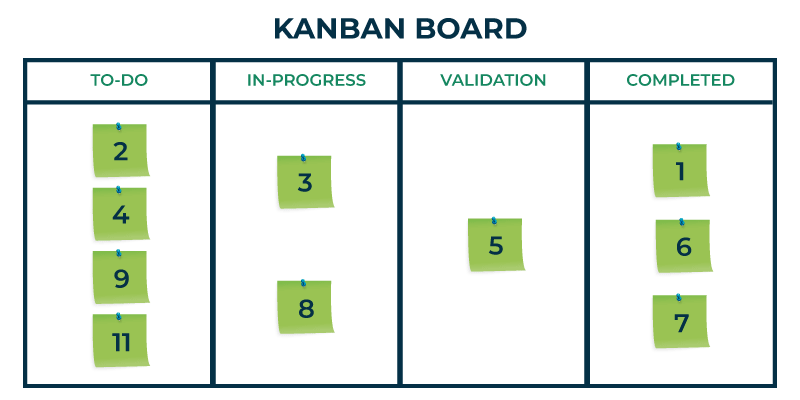


Figure 8: Kanban Board (aadityasharma0115, 2023)

#### Phases of Kanban

* **To-Do**

This is where everything begins with a clean slate. Tasks are listed in the first column, outlining user stories with essential details like deadlines, features, and task nature.

* **In-Progress**

Tasks in this column are actively being worked on by the team, signaling ongoing development efforts.

* **Validation**

Here, code is rigorously reviewed, and models undergo various tests such as unit, acceptance, and system testing. Any detected bugs or inconsistencies are addressed during this phase.

* **Done**

Once all functional and system tests are successfully completed, and quality assurance is confirmed, tasks move to the Done column. This column represents tasks that have been completed, deployed error-free, and are ready for use.

(aadityasharma0115, 2023)

#### Advantages of Kanban

* Continuous Improvement
* Reduced Waste
* Focus and efficiently.
* Visualization and transparency

#### Disadvantages of Kanban

* Unclear roles and responsibilities
* Limited structure
* Unpredictability
* Oversimplification

### Lean development

Lean software development focuses on maximizing efficiency and reducing waste in the software development process, drawing inspiration from the Lean manufacturing movement of the 1980s. It has evolved to become a fundamental aspect of Agile software development practices. (Ben Lutkevich, 2021)

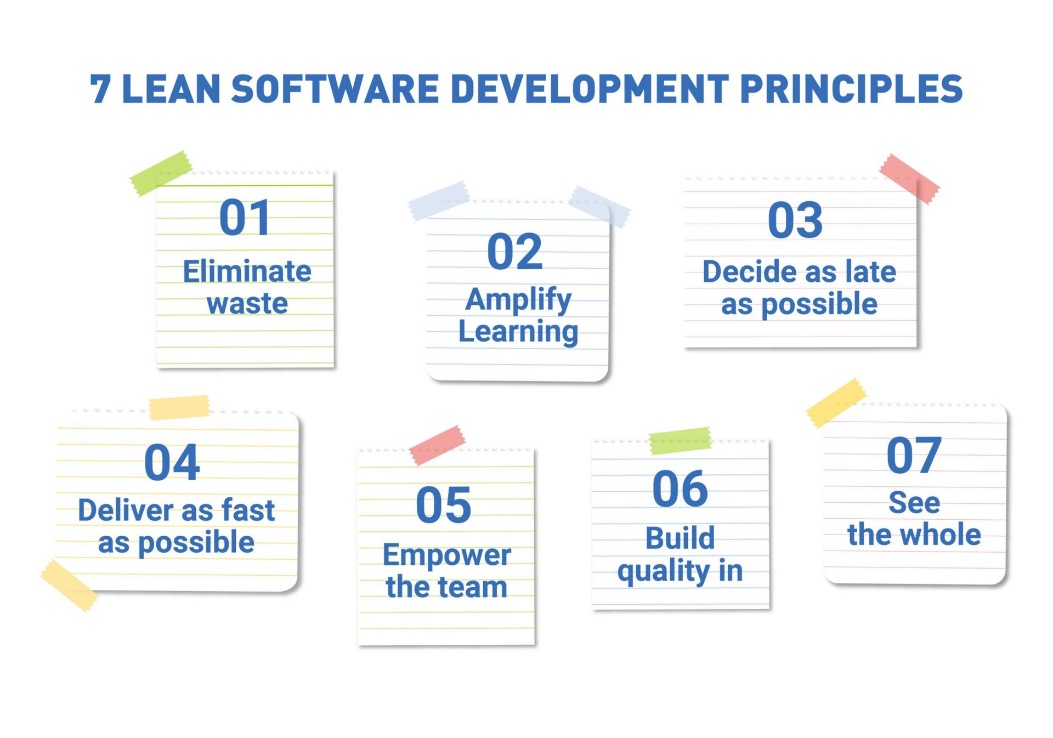


Figure 9: Lean Development principles (Staff, 2024)

#### Lean Development Principles

* **Eliminate Waste**

In lean software development, a core principle is to eliminate waste, which refers to anything that doesn't add value for the customer.

* **Amplify Learning**

In the context of software development, the idea of amplifying learning involves facilitating the software team's understanding of the system and requirements quickly and efficiently. Traditionally, this has involved heavy reliance on documentation and design processes.

* **Decide as late as possible.**

This principle suggests that there should be sufficient adaptability in both the system and the process to allow for decisions to be made later in the project. Early decisions shouldn't be constrained by the project's initial requirements.

* **Deliver as fast as possible.**

Customers and users generally prefer swift delivery of software. It can sometimes be challenging for them to comprehend the reasons behind delays in implementing changes. Nonetheless, expediting software delivery is advisable for various reasons, even if they understand the complexities involved.

* **Empower the team.**

In numerous organizations, managers typically make decisions and direct developers on what tasks to undertake, with developers following their instructions.

in a lean approach, the development team is granted more autonomy in decision-making. They have the freedom to perform their tasks, offer suggestions and recommendations to leaders and managers, and assess situations based on their own expertise and experience.

* **Build quality in**

Customers value the integrity of a system, which is demonstrated through two methods: conceptual integrity and perceived integrity.

Conceptual integrity entails ensuring that all components of the system harmonize effectively to create a cohesive whole. It should possess qualities like flexibility, maintainability, efficiency, and responsiveness. Additionally, the design and usability should remain consistent throughout and align with the requirements and business challenges it addresses.

Perceived integrity, on the other hand, refers to how users perceive and observe the system. This encompasses its functionality, delivery process, maintenance, pricing, and relevance to solving business issues.

* **See the whole.**

This principle of lean software development emphasizes the importance of development teams understanding the overall "big picture" of the system. Nowadays, software is rarely developed in isolation; it often interacts with other systems directly or through data feeds. It may even be a component of a larger system, such as Intranet sub-sites.

When designing and developing these systems, it's crucial to take into account these interactions. Sufficient testing during the release process is necessary to minimize any adverse impacts on other systems.

#### Advantages of Lean Development

Improved Quality

Faster Delivery

Empowered Teams

Reduced Waste

Increased Customer Statistician

#### Disadvantages Lean Development

Limited Scope

Focus on speed.

Requires disciplines.

Cultural shift

## Strengths and Weaknesses of Agile Methodology

### Strengths of Agile Methodology

* **Flexibility**

Agile adapts well to changing requirements, allowing for course correction as needed.

* **Speed to Market**

It delivers usable increments quickly, getting a basic product to users faster for early feedback and iteration.

* **Customer Satisfaction**

Collaboration with stakeholders ensures the final product meets their needs, enhancing satisfaction.

* **Team Communication**

Regular interaction fosters a collaborative environment, improving communication among team members.

* **Focus on Functionality**

Prioritizing working software over extensive documentation ensures the product delivers value.

### Weaknesses of Agile Methodology

* **Cost Uncertainty**

Evolving requirements make it challenging to estimate project costs accurately upfront.

* **Team Discipline Requirement**

Agile relies on disciplined, self-organized teams for effectiveness; without it, projects can become chaotic.

* **Documentation Challenges**

Emphasis on working software may lead to inadequate documentation, posing challenges for onboarding new team members or future maintenance.

* **Not Universally Applicable**

Agile may not suit projects with rigid requirements or extensive regulations requiring upfront planning.

* **Scope Creep Potential**

Constant iteration may lead to feature creep, potentially bloating the project with unnecessary additions.

## Comparison of Strengths and Weaknesses of Traditional vs. Agile System Analysis Methodologies

Table 1: Comparison of Strengths and Weaknesses of Traditional vs. Agile System Analysis Methodologies

|  |  |  |
| --- | --- | --- |
|  | Traditional Methodology | Agile Methodology |
| Strengths |  |  |
| Planning | Thorough upfront planning reduces risks and ensures all requirements are considered. | Iterative planning allows for flexibility and adaptation to changing needs. |
| Documentation | Clear documentation aids in future maintenance and onboarding new team members. | Less emphasis on documentation fosters faster development and reduces overhead. |
| Predictability | Defined phases and milestones provide a clear roadmap for development, making it easier to estimate timelines and costs. | Fast feedback loops allow for continuous improvement and course correction. |
| Project Size | Suited for large, well-defined projects with complex requirements. | More adaptable for projects with evolving requirements, especially smaller projects. |
| Customer Focus | (Limited) Customers have some influence early on, but requirements are locked in later stages. | High Customers are actively involved throughout the process, ensuring the final product meets their needs |
| Weaknesses |  |  |
| Flexibility | Inflexible and difficult to adapt to changing requirements during development. | Unpredictable delivery timelines and costs due to focus on flexibility. |
| Time to Market | Slow time to market due to lengthy upfront planning phase. | Limited documentation can lead to knowledge gaps and hinder future maintenance. |
| Risk Management | Relies on upfront planning to mitigate risks. | Early and frequent testing helps identify and address issues quickly but may miss broader risks. |
| Team Structure | Requires a structured and hierarchical team environment. | Requires a highly collaborative and self-organizing team. |

## Similarities Between Traditional vs. Agile System Analysis Methodologies

Table 2: Similarities Between Traditional vs. Agile System Analysis Methodologies

|  |  |  |
| --- | --- | --- |
| Feature | Traditional Methodology | Agile Methodology |
| Project Goal | Understand business needs and deliver a high-quality system that meets all requirements. | Understand business needs and deliver a high-quality system that provides value to users, with the ability to adapt to changing priorities. |
| Stakeholder Focus | Identify stakeholders upfront and manage their expectations through formal communication channels (reports, presentations). | Actively involve stakeholders throughout the process, gathering continuous feedback through informal interactions and demonstrations. |
| Quality Assurance | Implemented through well-defined testing phases (unit testing, integration testing, system testing). | Integrated into each iteration with automated testing and manual testing practices. |
| Risk Management | Mitigated through upfront planning, risk identification workshops, and contingency plans. | Reduced through early and frequent testing, allowing for quick course correction and adaptation to identified issues. |
| Project Delivery | Delivers a complete system upon project closure, meeting all pre-defined requirements | Delivers working functionalities in increments (iterations), with the final system evolving based on continuous feedback. |
| Documentation | Detailed and comprehensive documentation outlining system requirements, design, and implementation. | Less emphasis on formal documentation, with a focus on maintaining user stories and backlog items that capture evolving requirements. |
| Project Management | Relies on a predefined project plan with clear milestones and timelines | Emphasizes flexibility with an adaptive project plan that adjusts based on learnings from each iteration. |
| Communication | Formal communication channels with periodic progress reports and meetings. | Informal and ongoing communication with daily stand-up meetings and user feedback sessions. |

## Comparison in the Context of “Auto Cars” Web System

Table 3:Comparison in the Context of “Auto Cars” Web System

|  |  |  |
| --- | --- | --- |
| Aspects | Agile Methodology for “Auto Cars” | Traditional Methodology For “Auto Cars” |
| Flexibility and Adaptability | High - Adapts to changing requirements during development. | Low - Difficult to change requirements after initial planning. |
| Customer Collaboration | High - Continuous customer involvement ensures alignment with needs. | Low - Limited customer involvement may lead to mismatches. |
| Iterative Development | Fast delivery of core functionalities with ongoing refinement. | Slower development due to sequential phases and limited iteration. |
| Risk Management | Requires ongoing risk identification and mitigation throughout the project. | Upfront risk assessment helps identify potential issues early. |
| Documentation | Focuses on functional software over extensive documentation. | Emphasizes comprehensive documentation for future reference. |
| User Satisfaction | High emphasis on user feedback for continuous improvement. | Risk of user needs not being fully met due to limited involvement. |
| Time-to-Market | Faster delivery of features through iterative development. | Potentially longer development cycles due to sequential phases. |

**Traditional Methodology (Waterfall)**

**Strengths**

* **Clear System Requirements:** Thorough initial planning ensures all functionalities like customer and driver registration, booking, SMS notifications, driver selection, and location tracking are well-defined.
* **Compliance & Regulations:** Well-suited for meeting data privacy regulations and security standards relevant to Sri Lanka.
* **Large System Structure:** Beneficial for integrating with existing phone networks for SMS notifications or mapping services due to its structured approach.
* **Extensive Documentation:** Detailed documentation aids in future maintenance and training for operators and potential developers.

**Weaknesses**

* + - * **Slow Time to Market:** Extensive upfront planning can delay system launch, potentially causing a loss in competitive advantage.
      * **Limited Customer Input:** Customers have minimal influence on early design, potentially overlooking valuable features like real-time ride tracking.
      * **Inflexible to Changes:** Adapting to new features or user feedback during development can be challenging and costly, potentially causing delays or increased budget.

**Agile Methodology:**

**Strengths:**

* + - * **Fast Feedback and Iteration**: Regular user testing allows for early feedback on functionalities such as the booking process and driver selection, enabling quick adjustments to enhance user experience.
      * **Focus on Core Functionality:** Prioritizes delivering a Minimum Viable Product (MVP) with essential booking and notification features first, facilitating faster launch and customer feedback.
      * **Adaptability to Market Trends:** Iterative approach enables the incorporation of features based on user feedback, such as estimated ride fares and in-app payment options, to stay aligned with evolving customer needs.
      * **Customer Centricity:** High level of customer involvement ensures the system remains user-friendly and addresses evolving needs effectively.

**Weaknesses:**

* + - * **Unpredictable Delivery Timelines:** Flexibility may make it challenging to accurately estimate timelines for complex features like driver location tracking, which could be crucial.
      * **Limited Upfront Planning:** Less emphasis on detailed planning may result in missing certain functionalities in the initial version, necessitating additional development cycles.
      * **Team Expertise:** Requires a highly collaborative team with expertise in web development and user experience design, which may need to be assembled specifically for the project.

## Recommended Approach

Auto Cars' web-based taxi reservation system can achieve maximum success by adopting a hybrid approach that combines elements of traditional and agile methodologies. This approach leverages the strengths of both worlds to deliver a secure, user-friendly, and adaptable platform.

In the initial phase, traditional methods take center stage. Here, Auto Cars prioritizes building a secure and compliant foundation for the system. They'll clearly define core functionalities like customer and driver registration, taxi booking, SMS notifications, and driver selection. Additionally, this phase ensures seamless integration with existing phone networks for SMS alerts and potentially mapping services for driver locations. Importantly, upfront planning allows for adherence to Sri Lanka's data security regulations and any other relevant industry standards. This focus on security and compliance ensures a strong foundation while minimizing technical risks. Following this initial build-out, the system transitions to an agile approach, allowing Auto Cars to continuously improve based on user feedback. This user-centric approach is crucial for long-term success. Through surveys, user testing, and customer support interactions, Auto Cars can gather valuable insights into user preferences and needs. Based on this feedback, they can prioritize and add features like real-time tracking or in-app payments, ensuring the system remains user-friendly and caters to evolving customer demands. Ultimately, this hybrid approach positions Auto Cars for success by providing a secure and adaptable system that keeps users happy and fosters a competitive edge in the market.

# Activity 02

Feasibility Study Report

[Auto Car web-based booking system]

“Auto Cars”

Taxi service in Sri Lanka

**Name: Mohamed Zaheer**

**E Number: E171743**

## 2.1. Feasibility Study

A feasibility study evaluates the chances of a planned project, like introducing a new product or system, succeeding. It looks into various aspects such as technical, economic, and legal factors to decide if the project is a sound investment. Additionally, it helps pinpoint any possible challenges or drawbacks that might come up during the project's implementation. (team, 2023)

Figure 10: Feasibility study.

### 2.1.1. Importance of Feasibility Study

* They help determine whether to proceed with or discard a project idea based on valid reasons.
* Feasibility studies assist in sharpening the focus of the project team.
* They offer valuable insights for guiding the next steps after the study is completed.
* Feasibility studies narrow down the range of potential business options.
* They evaluate both existing resources and the technology required for the project.
* Feasibility studies contribute to the project's success or failure by thoroughly assessing all relevant factors.
* They estimate the potential return on investment.

(team, 2023)

### 2.1.2. Types Of Feasibility Studies

#### Technical Feasibility

Determining technical feasibility involves assessing the availability of necessary technical resources and software within the organization. If these resources are present, it's crucial to ascertain if the technical team can adapt them to create new operational systems for the project. evaluating the hardware and software of existing equipment is essential to ensure compatibility with the proposed plan.

#### Economic Feasibility

Economic feasibility is the entail conducting the cost-benefit analyses to provide decision-makers with insights into potential economic advantages for the organization. Understanding the total expenses, including unforeseen costs, is vital for anticipating and addressing any financial challenges that may arise during the project.

#### Legal Feasibility

Legal feasibility involves verifying that all aspects of the proposed project comply with relevant legal regulations and requirements in the specific geographic area. This includes examining zoning laws, social media regulations, and other legal considerations to ensure full compliance before the project commences.

#### Schedule Feasibility

Schedule feasibility is a critical aspect of the study, estimating the time required for project completion. It's essential for all stakeholders to acknowledge and agree upon a realistic timeframe for successful project delivery.

#### Operational Feasibility

Operational feasibility evaluating the how well the proposed plan aligns with the current business environment and whether it will be adopted by end-users. Factors such as management support, user satisfaction with the current system, and the potential benefits to the organization are considered in this assessment.

(team, 2023)

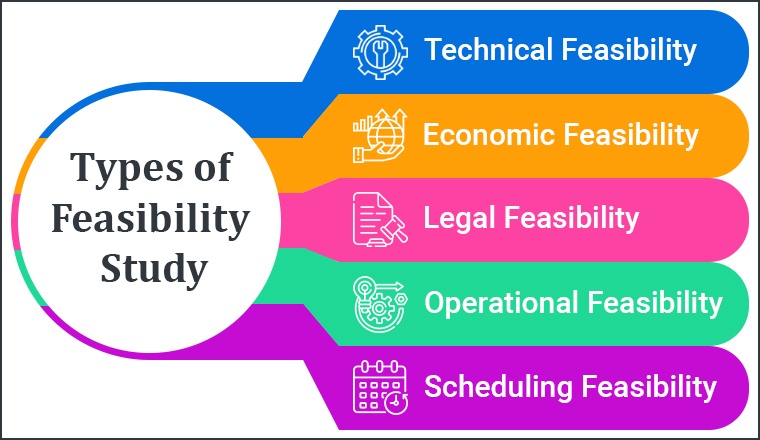


Figure 11: Types of Feasibility study.

## 2.2. Feasibility Study Report for the Web-Based System of "Auto Cars"

### Project Scope

The project is centered around developing a user-friendly web-based platform to simplify taxi reservations for both customers and drivers. It has three main objectives: firstly, to boost customer satisfaction by offering real-time booking confirmations, access to driver information, and vehicle tracking features. Secondly, it aims to improve operational efficiency by automating reservation processes to reduce wait times and errors. Lastly, the project seeks to gain a competitive edge by incorporating features like driver ratings and convenient payment methods. By achieving these goals, the platform endeavors to establish itself as a leading solution in the transportation industry, catering to the needs of modern consumers while surpassing the limitations of traditional reservation systems.

### Current Analysis

The current assessment reveals significant drawbacks in the existing taxi reservation system. It heavily relies on manual processes, leading to delays and a lack of real-time coordination between customers and available taxis. These issues result in a subpar user experience, as customers often encounter delays and uncertainties when attempting to book a taxi. Furthermore, the system struggles to expand efficiently to accommodate a growing customer base, highlighting its limited scalability. In today's tech-driven society, where efficiency and convenience are highly valued, the system fails to meet expectations. This highlights the pressing need for a more sophisticated and automated solution to overcome the shortcomings of the current system and better cater to the needs of both customers and drivers.

### Requirements

To ensure the success of the project, it's crucial to take a comprehensive approach, covering technology, human resources, and financial aspects. Regarding technology, we need a robust server infrastructure capable of handling high traffic, reliable database systems to maintain data integrity, an SMS gateway for instant communication, and email services for registration confirmations. Human resources are equally vital, requiring a skilled team comprising developers to build the system, designers to create a user-friendly interface, system analysts to ensure all requirements are met, and a project manager to oversee timelines and deliverables. Lastly, adequate financial resources are essential to cover initial setup costs, ongoing system maintenance, and effective marketing efforts to promote the new web-based platform. By addressing these critical areas, we can establish a solid foundation for development and ensure successful implementation.

### The Approach

During the development phase, utilizing the Agile methodology is crucial for the project's success. This method offers flexibility in both design and functionality, allowing the team to swiftly adapt to user feedback and changing market demands. By dividing the development process into iterative cycles, Agile ensures continuous improvement and guarantees that the final product meets or surpasses user expectations. Similarly, in the design stage, the main focus should be on crafting an accessible interface suitable for a diverse customer base, including people with disabilities. Prioritizing accessibility ensures that the platform is inclusive and easy to use for everyone, thereby enhancing the overall user experience. Additionally, it's essential to implement thorough testing protocols to ensure the system's security, reliability and ease of use before its launch. Rigorous testing helps detect and address any issues or glitches, reducing risks and ensuring a smooth rollout of the web-based platform. By adhering to these principles throughout development, design and testing, the project can deliver a top-notch product that fulfills the needs of both customers and drivers.

### Evaluation

Before proceeding with the project, it's essential to delve into market analysis to gauge the demand for online taxi services in the target market and assess competitors' strengths and weaknesses. This involves researching customer preferences, market trends, and competitor strategies to identify areas for differentiation and growth. Additionally, it's crucial to evaluate the technical feasibility of the project, determining if the current IT infrastructure can support the new system and identifying necessary upgrades or changes. This ensures the platform can operate effectively and meet user expectations without encountering technical limitations. Moreover, conducting a detailed cost-benefit analysis is vital to understand the financial implications and potential returns from the investment. This involves examining upfront costs, ongoing maintenance expenses, and projected revenue streams to make informed decisions about resource allocation and budget management. These analyses offer valuable insights into the project's viability and potential success, guiding strategic planning and decision-making processes.

### Review

Following a thorough feasibility study, it's clear that creating a web-based system for taxi reservations offers both promising advantages and inherent risks. The study highlights a strong demand for online taxi services in the target market, providing "Auto Cars" with an opportunity to leverage this trend and gain a competitive edge. By embracing advanced technology and agile methodologies, the company can enhance customer satisfaction, streamline operations, and boost overall efficiency. Nevertheless, there are challenges to address, including technical hurdles like ensuring compatibility with existing IT infrastructure and financial uncertainties regarding setup and maintenance costs. Moreover, market competition and evolving consumer preferences pose potential risks to the project's success.

Considering these findings, it's advisable to proceed with developing the web-based reservation system for taxis. The strategic objectives of "Auto Cars" align closely with the identified benefits from the feasibility study, suggesting that the project could positively impact the company's growth and competitiveness. With meticulous planning, effective risk management strategies, and ongoing monitoring, the project can navigate challenges and seize opportunities in the market. By prioritizing customer satisfaction, embracing technological advancements, and enhancing operational efficiency, "Auto Cars" has the potential to establish itself as an industry leader and achieve sustainable success.

## 2.3. Feasibility study for “Auto Car”

### ****2.3.1. Technical Feasibility****

The technical feasibility for Auto Cars’ web-based taxi reservation system focuses on assessing whether the existing technological resources are adequate to support the system’s requirements. This involves a detailed evaluation of the software and hardware components necessary for implementing functionalities such as online registration, SMS integration for communication, and real-time location tracking services.

For online registration, the system needs a reliable database server that can handle concurrent user requests and store sensitive customer and driver information securely. The software must be robust enough to manage user data and provide seamless registration experiences.

SMS integration is crucial for notifying customers about their booking details and driver information. This requires a service that can send SMS messages reliably and at scale, which may involve partnering with a telecommunications provider or using a cloud-based messaging service.

Real-time location services are essential for assigning the nearest driver to a customer. This requires GPS technology and mapping software that can accurately track and update driver locations. The hardware, likely in the form of smartphones or GPS devices within the vehicles, must be compatible with the software to ensure precise location tracking.

Overall, the technical feasibility study will determine if the current technology stack of Auto Cars can be upgraded or if new technology needs to be acquired to meet the system’s demands, ensuring a smooth and efficient operation.

### ****2.3.2. Economic Feasibility****

The economic feasibility of Auto Cars’ proposed web-based taxi reservation system is a critical consideration that involves conducting a cost-benefit analysis to weigh the financial implications of the project. This analysis will compare the estimated costs associated with developing, deploying, and maintaining the system against the anticipated benefits, such as improved operational efficiency, enhanced customer satisfaction, and potential expansion of the customer base.

**Costs** might include initial development expenses, such as hiring software developers, purchasing hardware, and investing in marketing to promote the new system. Ongoing costs could involve server maintenance, technical support, software updates, and SMS service fees.

**Benefits** are expected to manifest as increased bookings due to the convenience of online reservations, higher customer retention through improved service quality, and reduced manual labor costs due to automation. Additionally, the system’s data analytics capabilities could provide valuable insights into customer behavior, leading to more targeted marketing strategies and further revenue growth.

Ultimately, the economic feasibility will be affirmed if the long-term financial gains and competitive advantages of the system substantially surpass the investment and operational costs, ensuring a favorable return on investment for Auto Cars.

#### Cost For Auto Cars

Table 4: Cost for Auto Cars

|  |  |  |  |
| --- | --- | --- | --- |
| Year | Description | Costs (in LKR) | Cash Discounts (in  LKR) |
| Year 1 | Initial development  costs | 750,000 | -75,000 |
|  | Operational  expenses | 85,000 | -197,000 |
|  | Delivering expenses | 90,000 | -700,000 |
|  | Web server and IT-  related expenses | 905,000 |  |
|  | System implementation  expenses | 55,000 |  |
|  | Training expenses | 55,000 |  |
| Total | Year 1 Total Costs | 1,945,000 | -972,000 |
| Year 2 | Reduced  development and operational expenses | 600,000 | -197,000 |
| Total | Year 2 Total Costs | 403,000 | -197,000 |
| Year 3 | Further cost  reduction with cash discounts | 950,000 | -700,000 |
| Total | Year 3 Total Costs | 250,000 | -700,000 |

### ****2.3.3. Legal Feasibility****

The legal feasibility of the web-based taxi reservation system for Auto Cars is an essential aspect that involves ensuring the system’s compliance with all applicable laws and regulations. This encompasses a broad range of legal considerations, particularly those related to data protection and privacy laws. Given that the system will handle sensitive personal information of both customers and drivers, it is imperative to adhere to regulations such as the General Data Protection Regulation (GDPR) if applicable, or any local data protection laws in Sri Lanka.

The system must be designed to safeguard personal data against unauthorized access, use, or disclosure. This includes implementing robust security measures, such as encryption and secure data storage, as well as establishing clear policies for data retention and deletion. Additionally, the system should provide users with the ability to access, correct, and delete their personal information, in line with their legal rights.

Moreover, the legal feasibility study would also consider the implications of SMS communication, ensuring that it complies with telecommunications regulations and consumer protection laws, which may require obtaining explicit consent from users for such communications.

In summary, the legal feasibility ensures that the system not only protects the privacy and rights of individuals but also positions Auto Cars as a responsible and trustworthy service provider in the eyes of its customers and the law.

### ****2.3.4. Operational Feasibility****

Operational feasibility for Auto Cars’ web-based taxi reservation system is a measure of how well the proposed system will integrate into the company’s existing operational structures and workflows. It involves a comprehensive evaluation of the staff’s ability to adapt to and embrace the new technology, as well as the system’s potential effects on current business processes.

A key component of operational feasibility is staff training and support. The company must ensure that employees, especially the phone operators and drivers, are adequately trained to use the new system effectively. This includes understanding how to navigate the online platform, manage bookings, and communicate with customers through the system.

Another consideration is the impact on existing processes. The introduction of a web-based system may streamline certain operations, such as booking and dispatching taxis, but it could also require changes to other business practices. For instance, manual reservation methods may need to be updated or replaced, and data management practices might need to be revised to accommodate the digital system.

Ultimately, operational feasibility is confirmed when the new system can be integrated smoothly into the company’s operations without causing significant disruption, while also enhancing overall efficiency and service quality.

### ****2.3.5. Scheduling Feasibility****

Scheduling feasibility is a crucial aspect of project planning for Auto Cars’ web-based taxi reservation system. It involves establishing a realistic and achievable timeline for the project’s milestones and final completion. This timeline must account for all phases of the project, from initial design and development to testing, deployment, and post-launch support.

Creating a scheduling feasibility plan requires a detailed understanding of the project scope, the tasks involved, and the resources available. It also necessitates contingency planning for potential obstacles or delays that could impact the project timeline. For Auto Cars, this means setting clear deadlines for each development stage, allocating sufficient time for quality assurance, and ensuring that the launch aligns with the company’s strategic objectives.

The goal is to ensure that the system is developed efficiently and launched promptly, without compromising on quality or functionality. By adhering to a well-structured schedule, Auto Cars can coordinate its efforts effectively, manage expectations, and achieve a successful rollout of the new system within the desired timeframe.

#### Project Timeline For “Auto Cars”

Table 5: Project Timeline for Auto Cars

|  |  |
| --- | --- |
| Activity | Time (in weeks) |
| Specifying and gathering requirements | 4 weeks |
| Analyzing the requirements | 4 weeks |
| Designing the system | 3 weeks |
| Developing the system | 3 weeks |
| Testing the trial version | 2 weeks |
| Implementing the system | 4 weeks |
| Total | 20 weeks |

## 2.4. Importance of Feasibility Criteria in Evaluating the Proposed System for "Auto Cars"

### Financial Feasibility

Financial feasibility is a critical factor in assessing whether the proposed "Auto Cars" system makes economic sense. This assessment involves carefully analyzing the costs associated with developing, implementing, and maintaining the system, and comparing them with the potential benefits it could bring. In the context of "Auto Cars," this analysis considers several important elements. Firstly, it looks at the expenses related to acquiring the necessary autonomous vehicle technology, including sensors, software, and other essential components for autonomous driving. Additionally, it takes into consideration the costs of upgrading infrastructure to support autonomous vehicles, such as road markings, signage, and communication systems. Ongoing maintenance expenses, such as repairs, upgrades, and software updates, are also factored in. Furthermore, the analysis evaluates potential revenue streams that could arise from deploying "Auto Cars," such as improved efficiency, fewer accidents, and the introduction of new business models like ride-sharing services. By conducting a comprehensive financial feasibility analysis, decision-makers can gain valuable insights into the economic viability of implementing autonomous vehicles and make well-informed choices regarding resource allocation and investment in the "Auto Cars" project.

### Technical Feasibility

Assessing the technical feasibility of the "Auto Cars" system is crucial to determine if it can be effectively developed using existing technology. This evaluation involves closely examining the capabilities of autonomous driving technology, sensor systems, communication infrastructure, and software algorithms needed for autonomous vehicle operation. Specifically, for "Auto Cars," it requires thoroughly evaluating the current state of these technologies and systems. This assessment includes checking the reliability, accuracy, and efficiency of autonomous driving technology in safely guiding vehicles without human intervention. Additionally, it looks at the functionality and compatibility of sensor systems that detect obstacles, pedestrians, and other vehicles on the road. It also considers the communication infrastructure needed for real-time data exchange between autonomous vehicles and infrastructure like traffic lights. Lastly, it examines the strength and adaptability of software algorithms responsible for decision-making, route planning, and avoiding collisions in various driving situations. Through a comprehensive technical feasibility analysis, stakeholders can determine if the proposed "Auto Cars" system can be implemented using existing technology and pinpoint any technological challenges that may need to be addressed for successful deployment.

### Operational Feasibility

considering the success of the proposed "Auto Cars" system, operational feasibility becomes crucial. This assessment focuses on whether introducing autonomous vehicles seamlessly fits into the organization's current operations, processes, and procedures. Specifically for "Auto Cars," it involves a detailed examination of how these vehicles become part of existing transportation systems, follow traffic regulations, meet safety standards, and gain public acceptance. This evaluation covers various aspects, like how autonomous vehicles interact with traditional human-driven ones on the road, navigate through different infrastructure types, and handle unexpected situations. Additionally, it delves into how autonomous vehicles affect existing transportation workflows, including maintenance, fleet management, and customer service. Furthermore, it looks at whether regulatory frameworks and infrastructure are prepared to support the integration of autonomous vehicles and considers public perceptions and concerns regarding safety, privacy, and trust in autonomous technology. Through a thorough operational feasibility assessment, stakeholders can determine if the proposed "Auto Cars" system aligns with current operations and identify any necessary adjustments to ensure successful integration and acceptance within the organization and the wider community.

### Legal and Regulatory Feasibility

Legal and regulatory feasibility is a key factor to consider when assessing the potential implementation of the "Auto Cars" system. This evaluation focuses on ensuring that the proposed system complies with relevant laws and regulations governing autonomous driving, liability, data privacy, and safety standards. Specifically tailored to "Auto Cars," legal and regulatory feasibility involves examining various legal aspects, such as regulations concerning the testing and operation of autonomous vehicles on public roads, liability frameworks in case of accidents involving autonomous vehicles, data privacy laws related to the handling of sensitive information by these vehicles, and compliance with safety standards to ensure their safe operation. Additionally, this assessment includes keeping up-to-date with evolving legal and regulatory requirements surrounding autonomous driving technology and ensuring continuous compliance throughout the development and deployment phases of the "Auto Cars" system. Through a comprehensive legal and regulatory feasibility assessment, stakeholders can identify any legal obstacles or compliance requirements that need to be addressed to ensure the successful and lawful implementation of the "Auto Cars" system.

### Schedule Feasibility

schedule feasibility evaluates whether the proposed "Auto Cars" system can be developed and deployed within a reasonable timeframe. This assessment involves considering various stages of the system development lifecycle, including research, development, testing, regulatory approval processes, and deployment of autonomous vehicle technology. Specifically tailored to "Auto Cars," schedule feasibility involves analyzing the time required for research and development efforts to advance autonomous driving technology, testing procedures to validate system functionality and safety, regulatory approval processes to obtain necessary permits and certifications for autonomous vehicle deployment, and the deployment timeline to integrate autonomous vehicles into existing transportation networks. Additionally, schedule feasibility entails identifying potential bottlenecks or delays in each phase of the project and implementing strategies to mitigate them to ensure timely delivery of the "Auto Cars" system. Through a thorough schedule feasibility analysis, stakeholders can gain insights into the realistic timeline for the development and deployment of the "Auto Cars" system and identify any necessary adjustments or optimizations to meet project deadlines and milestones.

## 2.5. Strength and weakness of Traditional and Agile Methodology

### 2.5.1. Traditional Methodologies Strengths

* **Clear Structure and Planning**

Traditional methodologies offer a structured approach akin to following a well-marked hiking trail. This roadmap outlines stages and milestones, aiding project managers in resource allocation, progress tracking, and anticipating challenges.

* **Detailed Documentation**

Traditional methods prioritize thorough documentation, comparable to a detailed recipe book for a complex dish. This comprehensive record serves as a valuable resource for future reference, team training, and issue troubleshooting during maintenance.

* **Predictable Timelines and Costs**

Like budgeting for a vacation, traditional methodologies enable upfront estimation of project duration and expenses. This transparency is vital for stakeholders in resource allocation and expectation management, facilitated by defined phases and tasks.

* **Focus on Quality**

Traditional methodologies resemble baking a cake with a precise recipe. Well-defined requirements and testing procedures ensure high product quality, minimizing rework and ensuring project compliance.

* **Suited for Large and Complex Projects**

Traditional methodologies excel in managing large-scale endeavors like building a skyscraper. Clear phases and roles facilitate coordination among large teams and management of intricate deliverables across departments.

### 2.5.2. Traditional Methodologies Weaknesses

* **Limited Flexibility**

Traditional methodologies can be rigid, akin to adhering to a strict train schedule. Incorporating changes mid-project may be challenging and time-consuming, potentially causing delays and rework.

* **Low Customer Involvement**

Traditional methodologies may involve limited customer participation, similar to designing a house without consulting the homeowner. This can result in a final product that doesn't fully meet customer needs.

* **Slower Time to Market**

Traditional methodologies, like a slow cooker recipe, involve lengthy development phases, leading to longer development cycles and delays in product launch. This can be a disadvantage in fast-paced markets where rapid innovation is critical.

* **Over-reliance on Initial Planning**

Traditional methodologies may overly rely on the initial project plan, akin to building a house on an inflexible foundation. Unforeseen challenges can disrupt projects if plans are too rigid to allow adjustments.

* **Potential for Communication Silos**

Defined roles in traditional methodologies may create communication barriers between project teams, resembling isolated departments in a company. This can hinder collaboration and result in information gaps.

### 2.5.3. Agile Methodologies Strengths

* **Adaptability and Flexibility**

Agile methodologies are like building a ship equipped to adjust sails based on wind changes. They excel at adapting to evolving requirements, employing an iterative approach for continuous course correction. Agile allows for incorporating new features or modifications during ongoing cycles, making it perfect for projects with changing needs.

* **High Customer Involvement**

Unlike a chef surprising with a meal, Agile methodologies actively engage customers throughout development. Customers participate in regular feedback sessions and demos, ensuring the product meets their needs. This involvement increases the likelihood of delivering a satisfying product.

* **Faster Time to Market**

Agile methodologies are akin to a race car swiftly adapting to different tracks. They enable quicker delivery of working features through iterative cycles, breaking down the project into smaller chunks ("Sprints"). This approach allows for faster feedback loops and shorter time to market compared to traditional methods.

* **Focus on Collaboration**

Agile methodologies promote a collaborative and self-organizing team environment, akin to a jazz band improvising together. Team members work closely, fostering innovation and problem-solving. This collaborative approach encourages creative solutions and an engaged workforce.

* **Reduced Risk**

Agile methodologies continuously test and provide quick adjustments, akin to testing a recipe while cooking. Frequent testing and feedback loops identify and address issues early, minimizing the risk of major problems later. This iterative approach mitigates risks and ensures quality throughout development.

### 2.5.4. Agile Methodologies Weakness

* **Unpredictable Timelines and Costs**

Unlike a fixed train schedule, Agile methodologies can have unpredictable timelines and costs due to their focus on flexibility. While estimation techniques are used, evolving requirements can lead to fluctuations in both time and budget.

* **Limited Documentation**

Agile methodologies prioritize working software over extensive documentation, similar to a cookbook with only pictures. This focus may result in knowledge gaps for future team members or challenges in understanding system architecture later in the project.

* **Requires Committed Teams**

Agile success heavily relies on a motivated and collaborative team environment. Team members need to be adaptable, disciplined, and effective communicators. Without strong team commitment, Agile projects may struggle to achieve their goals.

* **Potential for Scope Creep**

Agile's focus on adaptation can lead to scope creep if not managed effectively, akin to a project continuously adding new features without control. Balancing new features with project priorities and timelines is essential to avoid overload and delays.

* **Not Ideal for Large Projects**

Agile methodologies may become challenging to manage in very large projects, resembling managing a complex orchestra. Coordinating dependencies and large teams can be overwhelming, where traditional methods might be better suited for their well-defined structure and roles.

## 2.6. Strength and weakness of Feasibility study

Table 6:Strength and weakness of Feasibility study

|  |  |  |  |
| --- | --- | --- | --- |
| Strength | Description | Weakness | Description |
| Informed Decision Making | Provides a data-driven analysis to objectively evaluate project viability. | **Time and Cost** | Can be resource-intensive, requiring time for research, analysis, and report writing. |
| Identification of Potential Issues | Uncovers potential problems early on, allowing for proactive solutions. | **Limited Predictability** | Future uncertainties like market shifts or technological advancements can impact the study's accuracy. |
| Improved Resource Allocation | Helps determine and optimize resource needs (personnel, finances, equipment) for efficient project execution. | **Focus on Negatives** | May lead to a pessimistic view by highlighting potential problems more than potential benefits. |
| Increased Project Success Rate | Improves project design and implementation by identifying weaknesses, leading to a higher chance of achieving goals. | **Over-Reliance on the Study** | Shouldn't be the sole decision-making factor. Integrate it with business judgement and ongoing market research. |
| Stronger Risk Management | Identifies and assesses potential risks, allowing for development of mitigation strategies and contingency plans. | **Potential Bias** | The study team's unconscious biases might influence findings. Consider using an objective team or external consultants. |

## 2.7. Critical Evaluation

The feasibility study is crucial for addressing specific business challenges within the Auto Cars system. In the automotive industry, such studies are vital for evaluating the technical and economic feasibility of potential projects, like introducing new vehicle components or developing autonomous vehicle technologies. Conducting a feasibility study enables Auto Cars to thoroughly assess the possibilities, costs, and benefits of implementing new systems or technologies. This evaluation helps the company minimize risks and make informed decisions by gauging the likelihood of a project's success and its profitability. Specifically, for the Auto Cars system, a feasibility study is essential for evaluating the technical and commercial viability of introducing new features, such as a web-based reservation system. It offers a comprehensive analysis of the technical and economic aspects, enabling the company to validate its plans and make informed decisions before moving forward. Additionally, the feasibility study helps identify potential risks and tasks associated with implementing new technologies, assisting Auto Cars in determining the value and potential risks of investing in specific business solutions. Therefore, the feasibility study serves as a critical tool for Auto Cars in assessing the feasibility and potential impact of integrating new technologies or systems, ultimately aiding in more effective decision-making and risk management.

# Activity 03

## 3.1. User requirements

User requirements encompass the needs and desires of the customer in natural language, accompanied by visual aids such as diagrams or tables. They detail the services offered by the system and any constraints it must adhere to in operation. User requirements are centred on what actions or functionalities the user wants or needs from the system, outlining the tasks they can perform. These requirements include both functional and non-functional aspects, presented in a way that is understandable to users without technical expertise. They are documented in a User Requirements Document using descriptive text rather than technical jargon or formal notation. (KOVOOR, 2023)

Figure 12: User Requirements

### 3.1.1. User Requirements for “Auto Cars” Web system.

**Registration**: Users must be able to register and receive login credentials via email.

**Taxi Booking**: Users must be able to book taxis and receive driver details via SMS.

**Driver Rating**: Users must have the option to rate drivers after their trip.

**Manual Reservation**: Phone operators should be able to reserve taxis for users without an online account.

**Driver Assignment**: The system must prioritize the driver closest to the user’s location for assignment.

**Location Entry**: Both drivers and users must be able to manually enter or update their locations. or

## 3.2. Organizational Requirements

Organizational requirements refer to the rules and guidelines that users must follow when performing their tasks within an organization. These requirements are essential for ensuring effective operations, collaboration and alignment with organizational goals.

### 3.2.1. Organizational Requirements for “Auto Cars” Web system.

**Competitive Advantage:** The system should provide a unique selling proposition compared to competitors.

**Operational Efficiency:** The system should streamline the booking and dispatch process.

**Customer Retention:** Features like driver ratings should help improve service quality and customer loyalty.

**Data Analytics:** The system should be able to analyze customer behavior for business insights.

**Compliance:** The system must comply with local regulations and data protection laws.

## 3.3. System Requirements

System requirements are a structured document that outlines the functions, services, and operational limitations of a system. It serves as a guide for developers to construct and execute the system design. This document specifies the functionalities necessary for the system to meet user requirements. System requirements can be communicated in natural language or more structured formats with graphical elements. However, since it targets developers, it provides a detailed description of the project services. It is written in natural language and documented as a System Requirement Specification (SRS). (KOVOOR, 2023)

Figure 13: System Requirements

### 3.3.1. System Requirements for “Auto Cars” Web system.

**Database:** A secure and scalable database to store user and trip data.

**SMS Gateway:** Integration with an SMS gateway for sending booking confirmations.

**GPS Functionality:** Real-time GPS tracking for driver location updates.

**User Interface:** An intuitive and user-friendly web interface for booking and managing trips.

**Backup and Recovery:** Reliable backup and recovery solutions to protect data integrity.

## 3.4. Functional Requirements

A functional requirement essentially describes how a system is supposed to act, outlining what tasks it needs to perform to fulfil the user's requirements or desires. These requirements represent the visible features or functionalities that users interact with. They are distinct from non-functional requirements, which dictate how the system operates internally, such as its performance or security measures. (janardan333, 2024)

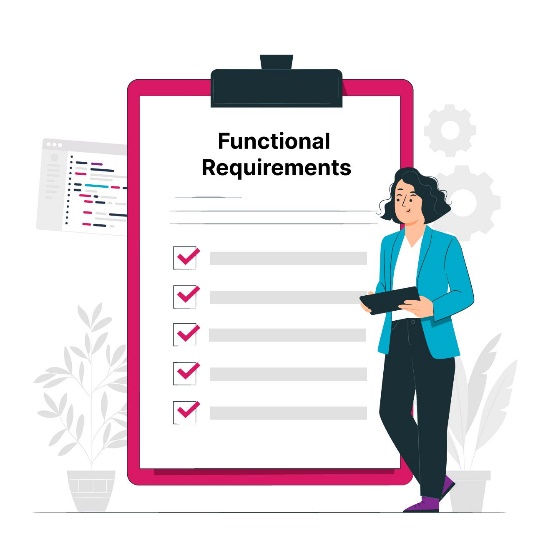


Figure 14: Functional Requirements

### 3.4.1. Functional Requirements for “Auto Cars” Web system.

**User Authentication**: Secure login process for users and drivers.

**Booking System**: A real-time booking system with SMS confirmation.

**Rating System**: A feature for users to rate drivers and provide feedback.

**Operator Interfac**e: A dedicated interface for phone operators to manage reservations.

**Location Services**: Accurate and up-to-date location services for driver and user location entry.

## 3.5. Non-Functional Requirements

Non-functional requirements are essentially the limitations or conditions placed upon a system, detailing the quality characteristics of the software. They cover aspects like how well the software can adapt to growth, how easy it is to maintain, its speed, flexibility, security, dependability, and other similar factors. These requirements are crucial for ensuring the overall quality of software systems. (AshmeetSaggu, 2024)

### 3.5.1. Non-Functional Requirements for “Auto Cars” Web system.

**Performance**: The system should be fast and responsive, with minimal downtime.

**Scalability**: The system should be able to handle a growing number of users and bookings.

**Security**: Strong security measures to protect user data and prevent unauthorized access.

**Usability**: The system should be easy to use, with clear navigation and instructions.

**Reliability**: The system should function correctly, providing consistent and accurate information.

## 3.6. Analysis of the Most Suitable Methodology for “Auto Cars”

For “Auto Cars” to develop a web-based taxi reservation system that meets the outlined requirements, an Agile methodology would be most suitable. Agile methodologies are known for their flexibility, iterative development, and emphasis on customer collaboration, which aligns well with the dynamic needs of such a service.

why Agile methodologies most suitable

* **Customer Collaboration**

Agile methodologies prioritize customer involvement throughout the development process. This is crucial for “Auto Cars” as the system needs to be user-friendly for both customers and drivers.

* **Iterative Development**

Agile allows for iterative releases, enabling “Auto Cars” to launch a Minimum Viable Product (MVP) and then incrementally add features like driver rating, SMS notifications, and manual operator reservations.

* **Flexibility**

Given the need for real-time updates of driver locations and availability, Agile’s flexibility would allow “Auto Cars” to adapt the system quickly in response to user feedback or changing requirements.

* **Continuous Testing and Integration**

Agile methodologies often include continuous testing, which would help ensure that the registration, booking, and SMS notification features work seamlessly together.

* **Scalability**

As “Auto Cars” grows, the Agile approach would facilitate scaling the system to accommodate more users, drivers, and potentially more services.

#### Agile Implementation Steps For Auto Cars

For "Auto Cars," employing the Agile methodology would be ideal for constructing their web-based reservation system. Agile's iterative approach allows for flexibility and adaptability, which is vital for a project with evolving requirements like this one.

* **Project Initiation:** Initiate with a meeting to outline the project's vision and scope, involving stakeholders from "Auto Cars" and the development team.
* **Creating Product Backlog:** Compile a list of all required features (a-g) for the web-based system, prioritizing them based on business value and customer needs.
* **Sprint Planning:** Divide the project into sprints, each aimed at delivering a potentially shippable increment of the product, like the registration system or booking feature.
* **4.Development and Testing:** Within each sprint, the team works on designing, developing, and testing the features to ensure they meet the acceptance criteria and are ready for deployment.
* **Customer Feedback:** Following each sprint, present the increment to "Auto Cars" for feedback, which can then be integrated into the subsequent sprint.
* **Continuous Improvement:** Conduct retrospective meetings at the end of each sprint to review successes and areas for improvement, adjusting the process accordingly.

## 3.7. Reason for choosing scrum methodology for “Auto Cars “Web based System.

Scrum methodology is particularly well-suited for the "Auto Cars" web-based system scenario due to several reasons:

* **Flexibility and Adaptability:** Scrum allows for rapid adaptation to changes, which is essential in a dynamic business environment like the taxi service industry.
* **Iterative Development:** The iterative nature of Scrum enables the development team to build, test, and release smaller sections of the project (like the booking system or driver rating feature) more frequently, which aligns with the need for continuous improvement.
* **Customer Collaboration:** Scrum emphasizes close collaboration with the customer, ensuring that the product being developed meets their needs and expectations, which is crucial for user satisfaction in service-oriented platforms.
* **Transparency:** Scrum provides a transparent process where progress can be easily tracked and communicated to all stakeholders, including non-technical members like taxi operators.
* **Risk Management:** Regular sprints and reviews help in identifying and addressing issues early, reducing the risk of major problems at later stages of the project.
* **Quality Control:** With regular sprints, the development team can focus on high-quality development, testing, and collaboration, leading to a better end product.
* **Efficiency:** Scrum can lead to more efficient use of time and resources by focusing on the development of high-priority features first, which is important for staying competitive in the market.

In summary, Scrum provides a structured yet flexible framework that can accommodate the evolving requirements of the "Auto Cars" web-based system, while also ensuring stakeholder engagement and high-quality output.

## 3.8. Advantages and Disadvantages of Scrum Methodology

Table 7: Advantages and Disadvantages of Scrum Methodology

|  |  |  |
| --- | --- | --- |
| Feature | Advantage | Disadvantage |
| Adaptability and Flexibility | Great for changing requirements | Requires experienced team |
| Increased Efficiency | Breaks large projects into sprints | Not ideal for large projects |
| Improved Quality | Regular reviews and feedback loops | Potential for scope creep |
| Boosted Team Morale | Self-organizing teams foster ownership | Frequent meetings can feel repetitive |
| Enhanced Customer Satisfaction | Early and frequent customer feedback | Heavy reliance on self-organization |

## 3.9. Justification of the Chosen Methodology for the Auto Cars

The choice of Scrum as the analysis methodology for the “Auto Cars” web-based system is justified by its alignment with the project’s goals and the dynamic nature of the service industry. Here are the key reasons:

* **Iterative and Incremental Approach**: Scrum’s iterative cycles, known as sprints, allow for the development of the web-based system in manageable increments. This approach enables the team to focus on delivering specific features, such as the customer registration or taxi booking system, and then refine them over successive sprints based on feedback and testing.
* **Customer and Stakeholder Engagement**: Scrum involves customers and other stakeholders in the development process, ensuring that their needs and expectations are understood and met. For “Auto Cars,” this means that both drivers and customers can provide input that influences the system’s design and functionality, leading to a more user-centric product.
* **Flexibility to Change**: The taxi service industry is subject to frequent changes due to varying customer demands, regulatory changes, and technological advancements. Scrum’s flexible framework allows “Auto Cars” to adapt to these changes quickly, ensuring that the system remains relevant and effective.
* **Transparency and Communication**: Scrum promotes transparency throughout the development process. Regular meetings and reviews provide visibility into the project’s progress, facilitating clear communication between the development team, management, and other stakeholders.
* **Risk Reduction**: By delivering working features at the end of each sprint, Scrum helps to identify and address issues early, reducing the risk of project delays or failures.
* **Quality Focus**: Scrum’s emphasis on continuous improvement and regular reviews ensures that the web-based system is developed to a high standard, with quality being a priority throughout the development lifecycle.

In summary, Scrum is an effective methodology for the “Auto Cars” project as it supports the company’s need for a responsive, high-quality, and customer-focused solution, while also providing the structure and discipline required to manage the project efficiently. This makes it an ideal choice for developing a complex, interactive web-based system in the service industry.

## 3.10. Effectiveness and suitability of the chosen methodology

Scrum is an effective and suitable methodology for the "Auto Cars" web-based system due to its emphasis on iterative development, flexibility, and customer feedback.

* **Iterative Development:** Scrum's sprint-based approach allows for frequent reassessment and refinement of the project, ensuring that each feature is developed to meet the evolving needs of the business and its customers.
* **Flexibility:** Scrum accommodates changes in project requirements, which is crucial for adapting to new market trends or customer feedback. This flexibility ensures that the final product remains relevant and competitive.
* **Customer Feedback:** Regular reviews and retrospectives with stakeholders ensure that the product aligns with customer expectations and business objectives, leading to higher customer satisfaction.
* **Risk Management:** By breaking down the project into smaller sprints, risks are identified and addressed early, reducing the potential for significant issues later in the development process.
* **Quality Focus:** Scrum's emphasis on continuous testing and integration helps maintain high quality throughout the development process, resulting in a reliable and user-friendly system.
* **Efficiency:** Prioritizing work based on value and urgency ensures that the most critical features are developed first, optimizing the use of resources and time.

In summary, Scrum's structured yet flexible framework is well-suited for developing the "Auto Cars" web-based system, as it supports the company's need for a responsive, high-quality, and customer-focused solution.

# Activity 04

## 4.1. Effectiveness of the design and the methodology used with reference to how it meets the user requirements.

* **Automatically generate documents.**

A requirements workbench enables the automatic generation of documents by leveraging data from the requirements model. This streamlines the process, allowing more time for requirements analysis and improvement rather than mundane document creation and maintenance. With minimal effort, fully formed documents can be obtained, enhancing efficiency and focus.

* **Easily create simulations.**

modern requirements platforms facilitate the creation of simulations from requirements models. These simulations, which utilize various elements such as use cases, data, visualizations, and textual requirements, offer an animated and real-time depiction of the application's future functionality. Reviews benefit significantly from simulations, enhancing understanding and communication.

Traceability is a critical aspect of the review process, ensuring that the analysis is linked back to the initial business necessity. Contemporary requirements software provides traceability not only during requirements writing but also during simulation analysis. It captures comments, criticism, and conversations generated during simulations, ensuring they are recorded and easily accessible for review and decision-making purposes.

* **Automatically generate tests.**

a requirements workbench automatically generates tests from the specified requirements model. This functionality is immensely valuable for Quality Assurance (QA) specialists working on software projects, as it covers various usage scenarios and links each test to the relevant requirement(s).

* **Integrate with development & test tools**

Integration with development and test tools is essential, as requirements serve as the foundation for developers and testers. Modern requirements software seamlessly integrates with popular UML design and QA testing toolkits, delivering high-quality requirements content where it is most needed, thereby facilitating the development and testing processes.

## System Design Specification for “Auto Cars” Web-Based Taxi Reservation System

* **User Registration and Authentication**

**Functionality**: Allow customers and drivers to register via an online form.

**Process:** After registration, an automated email is sent with login credentials.

* **Taxi Booking and Notification System**

**Booking Interface**: Customers book taxis using a web interface.

**SMS Integration:** On booking, an SMS is sent to the customer with driver and vehicle details.

* **Rating and Feedback Mechanism**

**Rating System:** Customers can rate drivers’ post-trip on the website.

* **Operator-Assisted Booking**

**Manual Entry:** Operators can book taxis for customers without an online account.

**SMS Notification:** The customer receives an SMS with booking details entered by the operator.

* **Driver Assignment Logic**

**Proximity Algorithm:** The system suggests the nearest driver to the customer’s location.

**Driver Selection:** Customers can choose from a list of available drivers.

* **Location Tracking**

**Location Input:** Drivers and customers manually enter or update their locations in the system.

* **Backend and Database**

**Database Management**: Store user data, booking records, driver ratings and location data.

**Server-Side Logic:** Handle authentication, booking logic, SMS notifications and driver assignment.

* **Frontend Development**

**User Interface:** Design a responsive web interface for customers and operators.

**Interactive Elements:** Include forms for registration, booking and rating.

* **Security Measures**

**Data Encryption:** Protect user data and credentials.

**Secure Communication:** Implement HTTPS for secure data transfer.

* **SMS Gateway Integration**

**SMS Service:** Integrate with an SMS gateway provider for sending booking details and confirmations.

* **Geolocation Services (Optional)**

**Mapping API:** Integrate a mapping API for real-time location tracking of drivers (if required).

## 4.2. System Specification Requirements (SRS)

A software requirements specification (SRS) is like a detailed blueprint that outlines exactly what the software is supposed to accomplish and the conditions it will operate in. It covers everything from the software's purpose to how it should function. (Rosencrance, 2019)

## 4.2.1 System Specification for Auto Cars

### Introduction

The Auto Cars system aims to revolutionize taxi reservations through a web-based platform, enhancing user experiences and streamlining booking processes. Users can access the system across various devices, such as smartphones, computers, and tablets.

### Objectives

**General Objective**

Develop an efficient web-based taxi reservation system for seamless customer-driver interaction.

**Specific Objectives**

* Create a user-friendly platform for convenient taxi reservations.
* Implement a dashboard for operators to manage reservations efficiently.
* Deploy and rigorously test the system for reliability.

### Project Scope

The system is a web-based application accessible on multiple devices, allowing for efficient taxi reservations locally and globally, eliminating manual searches and long-distance travel.

### Overall Description

**Product Perspective**

Web-based system employing client-server technology, offering cross-platform support, user accounts, and efficient local search capabilities.

### Product Function

Display available taxis to users.

Enable user account creation and login.

Provide an interface for administrators to manage transactions, add drivers, and define regions.

Allow drivers to manage customer orders through their accounts.

### User Characteristics

Users require basic internet knowledge and device proficiency, while operators and drivers may need additional technical skills.

### Constraints

Optimize the database for efficient data traffic.

Ensure HTML5 support for better user experience.

Depend on internet access for device connectivity.

### Assumptions and Dependencies

The system relies on browsers supporting HTML5, necessitating users with outdated browsers to consider upgrading.

### Specific Requirements

**External Interface Requirements**

* The system must be web-based.
* Enhancements should improve system features.

**User Interface Requirements**

* Implement a user-friendly graphical interface with easy navigation.
* Enable dynamic interface changes and include search functionality.
* Use attractive coloration and visuals.

**Hardware Interface Requirements**

* Mobile devices should have a processor speed of 0.5 GHz or higher.
* Desktop computers and gadgets should have a processor speed of 1.5 GHz or higher.
* All devices should have a minimum of 500 MB of RAM and more than 100 MB of free storage memory.

**Software Interface Requirements**

* Compatibility with various operating systems, including Windows, Android, Linux, Mac OS X, and Chrome.
* Supported browsers include Mozilla Firefox, Google Chrome, Opera Mini, UC Browser, or Internet Explorer.

**Communication Interface Requirements**

* Internet connectivity is essential.

### Application Maintenance

Table 8: Application Maintenance (Author Developed)

|  |  |
| --- | --- |
| Aspects | Description |
| Corrective Maintenance | Identifying and fixing bugs, defects, or errors that prevent the application from functioning as intended. This is often the first thing people think of when it comes to application maintenance. |
| Performance Optimization | Analyzing and improving the speed, responsiveness, and resource utilization of the application. This can involve tasks like code optimization, database tuning, and infrastructure upgrades. |
| Adaptive Maintenance | Ensuring the application remains compatible with new technologies and changing environments. This includes things like updating for new operating systems, hardware, and external APIs. |
| Preventive Maintenance | Regularly reviewing and updating the application code to identify and address potential problems before they occur. This can help to prevent costly downtime and data loss. |
| Security Maintenance | Keeping the application secure from vulnerabilities and cyberattacks. This involves patching security holes, implementing security best practices, and staying up-to-date on the latest threats. |
| Enhancement | Adding new features and functionalities to the application based on user feedback and evolving business needs. This helps to keep the application competitive and relevant. |

## 4.3. Data Flow Diagram (DFD)

A data flow diagram (DFD) illustrates the flow of information within a process or system using symbols like rectangles, circles, and arrows, along with concise text labels. These diagrams can vary from simple hand-drawn overviews to complex, multi-level representations that delve into how data is managed. They serve to analyze existing systems or model new ones effectively. DFDs have enduring popularity because they convey information visually, making it easier to understand complex concepts for both technical and non-technical audiences, from developers to CEOs. However, they may not be as suitable for visualizing interactive, real-time, or database-driven software or systems in today's context. (lucidchart, n.d.)

### Dataflow diagram for Auto cars

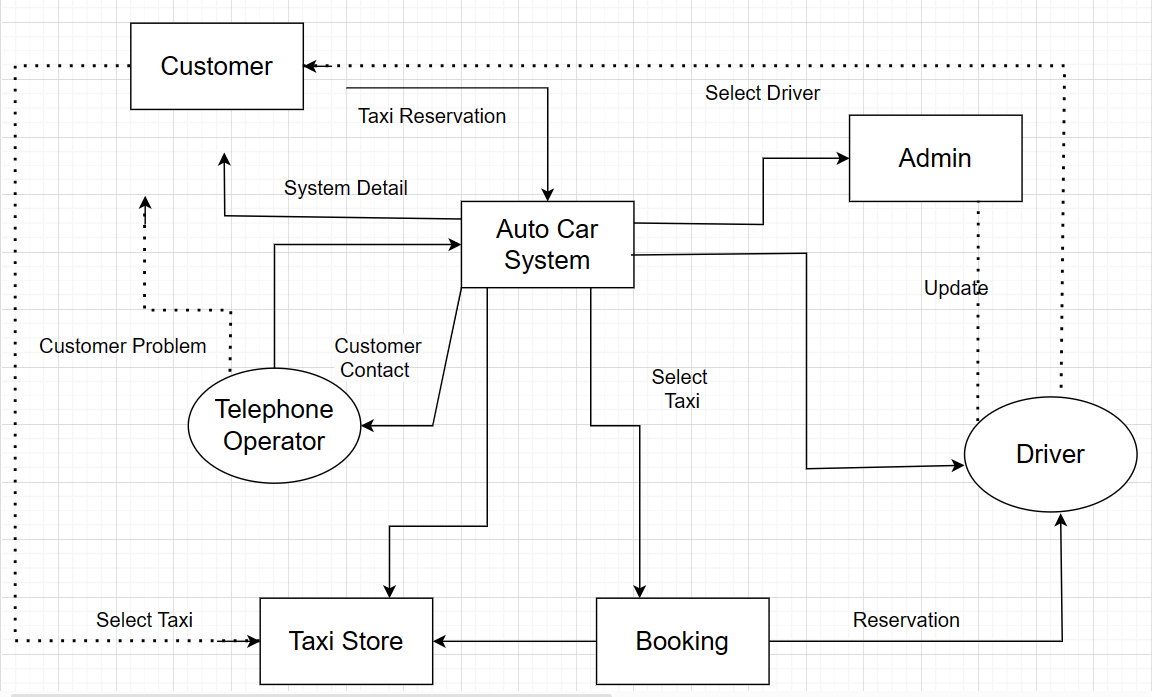


Figure 15: Data Flow Diagram Levelo 0

A diagram of a software company

Description automatically generated

Figure 16: Data Flow Diagram Level 1

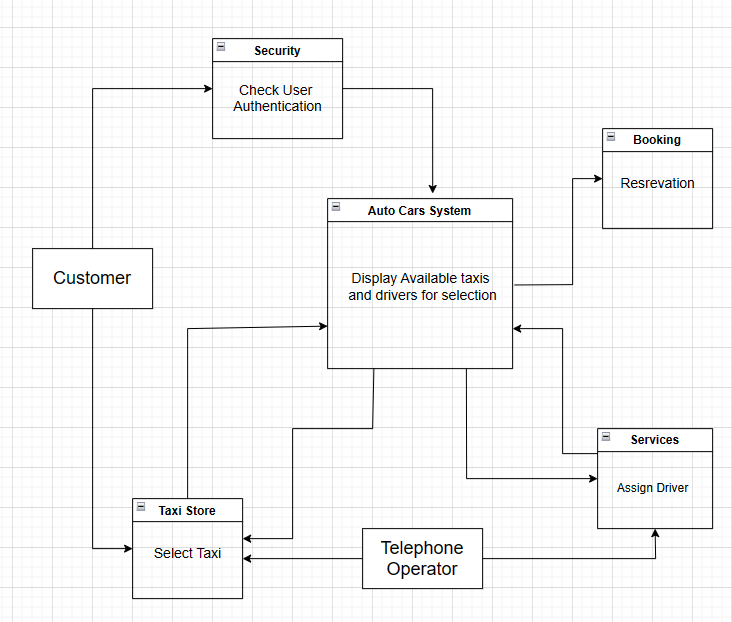


Figure 17: DFD Diagram Level 2

## 4.4. Use case Diagram.

A UML use case diagram visually represents how users or external systems interact with a system, showcasing its functionality from the user's viewpoint. These diagrams outline the system's functional requirements and illustrate how various actors achieve specific goals or tasks by interacting with the system. They offer a broad perspective on the system's capabilities, detailing how users or external entities engage with it. Use case diagrams facilitate communication among stakeholders, aiding in requirement clarification, boundary definition, and supporting development and testing efforts. (creately, 2022)

### Use case Diagram for Auto cars.

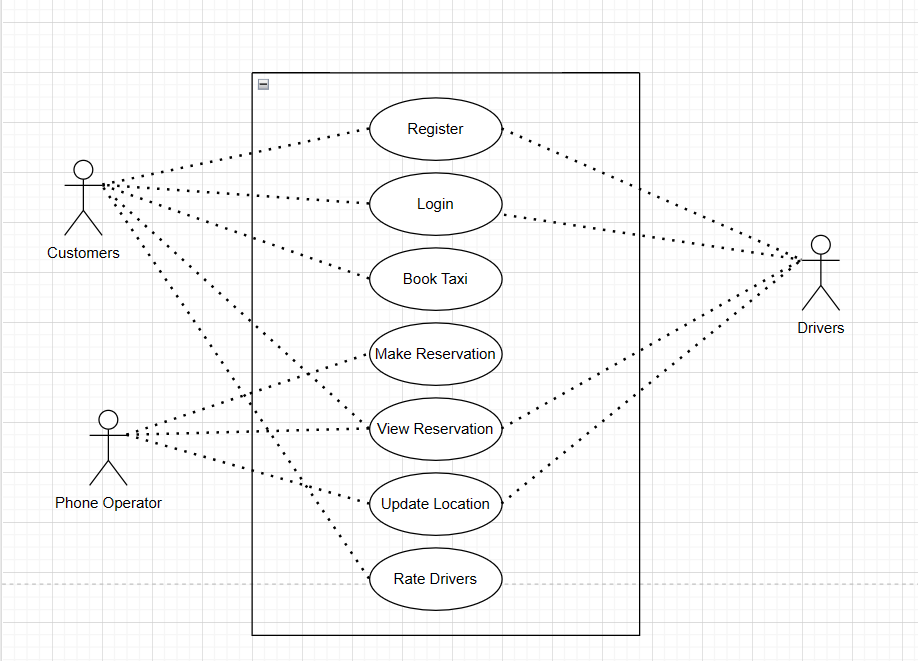


Figure 18: Use case Diagram for “Auto car” (Author Developed)

## 4.5. Flowchart Diagram

A flowchart is a visual representation of a process, system or computer algorithm, widely employed across various fields to document, analyze, plan, enhance and communicate intricate processes in a clear and understandable manner. Utilizing shapes like rectangles, ovals, diamonds and other symbols, along with connecting arrows, flowcharts illustrate the sequence and flow of steps. They can vary from basic, hand-drawn charts to intricate computer-generated diagrams showcasing multiple steps and pathways. Flowcharts are incredibly common globally, utilized by both technical and non-technical individuals in diverse industries. They may go by different names like Process Flowchart, Process Map, Functional Flowchart, Business Process Mapping, BPMN or Process Flow Diagram (PFD). They are closely associated with other popular diagrams such as Data Flow Diagrams (DFDs) and UML Activity Diagrams. (lucidchart., n.d.)

### Flowchart Diagram for Auto cars

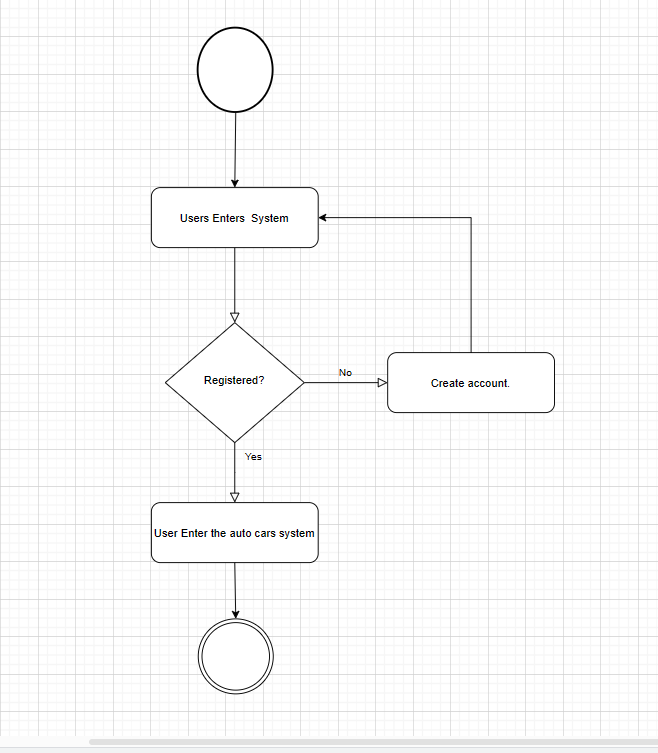


Figure 19: Flowchart Diagram (Author Developed)

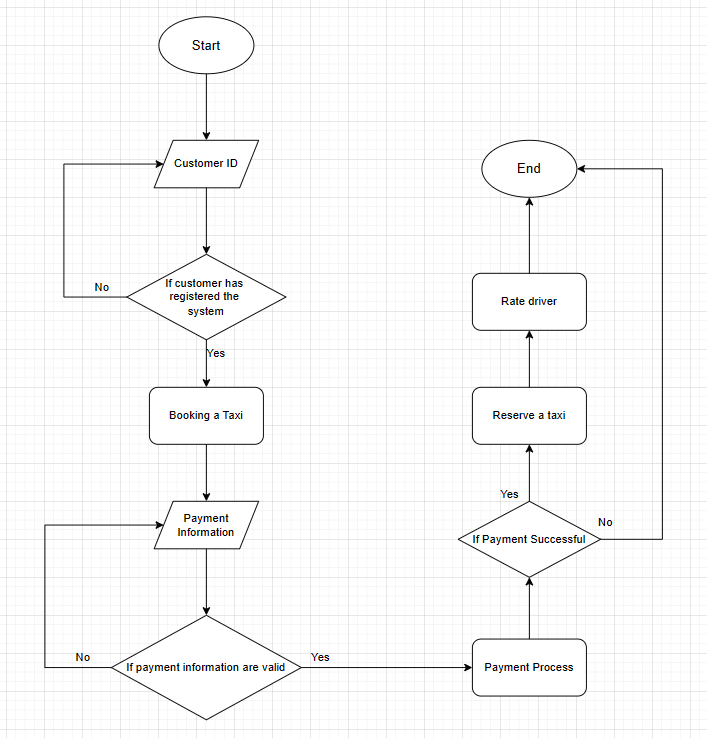


Figure 20: Flowchart diagram for auto cars (Author Developed)

## 4.6. Entity Relationship Diagram

An Entity Relationship (ER) Diagram is a graphical representation that shows the connections between "entities" like people, objects or ideas within a system. These diagrams also referred to as ERDs or ER Models are commonly used in software engineering, business information systems, education and research to design or troubleshoot relational databases. Using symbols such as rectangles, diamonds, ovals, and lines, ER Diagrams illustrate how entities, relationships and their characteristics are interconnected. They follow a structure similar to grammar, with entities representing nouns and relationships acting as verbs. (lucidchart, n.d.)

### Database Diagram For Auto cars

A diagram of a computer

Description automatically generated

Figure 21: Auto Cars For Database Diagram

### ER Diagram for Auto Cars

A diagram of a company

Description automatically generated

Figure 22: ER Diagram

## 4.7. Prototypes

### Login Page

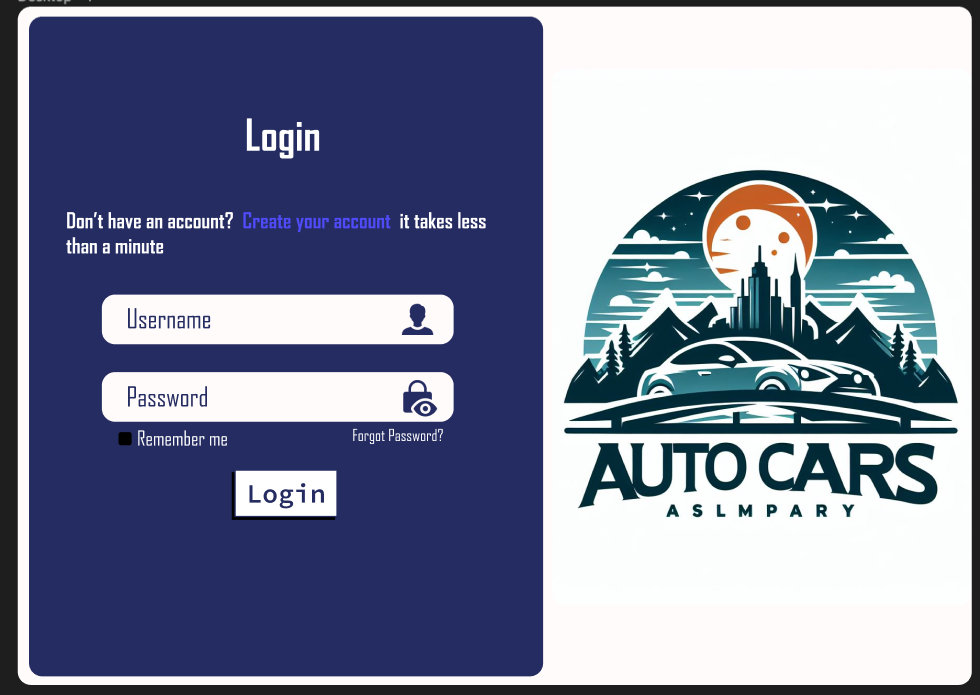


Figure 23: Login Page (Author Developed)

### Customer Registration



Figure 24: Customer Registration (Author Developed)

### Drivers Registration



Figure 25: Drivers Registration (Author Developed)

### Drivers Home Page

A screenshot of a car rental service

Description automatically generated

Figure 26: Drivers Home Page

### Customer Home Page

A blue and white sign with white text

Description automatically generated

Figure 27: Customer Home Page (Author Developed)

### Taxi Booking Page

A screenshot of a taxi application

Description automatically generated

Figure 28: Taxi booking page (Author Developed)

### Rating and Feedback Page

A screenshot of a computer

Description automatically generated

Figure 29: Rating and Feedback Page (Author Developed)

### Recent trip page

A blue and white card with white text and symbols

Description automatically generated

Figure 30: Recent Trip page (Author Developed)

### Settings Page

A screen shot of a blue and white screen

Description automatically generated

Figure 31: Settings Page (Author Developed)

## 4.8. Release Plans and Tools

### Release Plan Gant Chart

Table 9: Gant Chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **January** | | | | **February** | | | | **March** | | | | **April** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **System Design** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **UI Development** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Database Implementation** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Functionality Development** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Testing and QA** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Deployment and Training** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Post-Launch Monitoring** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### Tools

**System Design**

Tool: Trello (for task management and collaboration)

**UI Development**

Tool: Figma (for UI/UX design and prototyping)

**Database Implementation**

Tool: MySQL Workbench (for database design and management)

**Functionality Development**

Tools: HTML/CSS, JavaScript, PHP (for web development)

**Testing and QA**

Tool: Selenium (for automated testing)

**Deployment and Training**

Tools: Git (for version control), Jira (for issue tracking and project management)

**Post-Launch Monitoring**

Tool: Google Analytics (for website analytics)

## 4.9. Assessment of Effectiveness

Evaluating the effectiveness of the system design for "Auto Cars" involves looking at several key areas to make sure it meets both user and system needs. First and foremost, it's crucial that the design stays true to the initial requirements of the system. By using tools like data flow diagrams (DFDs) and entity relationship diagrams (ERD), we can make sure that these requirements are clearly represented and logically organized. DFDs show how data and processes move through the system, while ERDs help us understand the relationships between different parts of the system.

Flow charts are also important for giving everyone involved a clear picture of how the system works. They make complex processes easier to understand and help facilitate communication between stakeholders. Prototypes are another valuable tool, as they provide a hands-on way to see how users will interact with the system. By getting feedback on prototypes, we can continuously improve the design.

Having a well-defined release plan, laid out with a Gantt chart, ensures that the project progresses in a structured and realistic way, from design all the way to post-launch monitoring. And of course, choosing the right development and management tools is key. By using tools like Figma, MySQL Workbench, Git, and Jira, we can create a seamless workflow that meets the project's needs.

In summary, by focusing on requirements, using clear diagrams and visualizations, creating effective prototypes, following a realistic release plan, and choosing the right tools, we can ensure that the system design for "Auto Cars" is robust, user-friendly, and well-managed throughout its development and implementation.

## 4.10. Justification of Methodology in the Business Problem Context

The Choice to develop the agile method to develop the network of "Auto Cars" etc. The opportunity is warranted by the automotive industry’s strategy and allows for a corresponding uninterrupted transformation of the aggregate, which is the most important services in the industry and customer prioritization through its fastest movement and the changes that occur. which is of the customer reaction, of attachment And given priority, Agile ensures that "Auto Cars" remain finely tuned in order to meet the specific needs and expectations of their target audience, for customers overall satisfaction increases Furthermore, the Agile iterative approach to improvement, coupled with continuous testing, cultivates a culture of quality assurance and innovation, resulting in more polished reliability f final product This iterative quality also provides " Auto Car" ability to deliver functional versions of the system quickly, facilitate market entry quickly and provide flexibility to adapt real-time feedback that smoothly Ensures productivity and reduces disruption By fostering increased communication and cooperation among team members and between stakeholders so Agile encourages increased communication and alignment, which increases productivity and morale Furthermore, Agile's valuable feature of focus provides first materials used are better, so "Auto Cars" better allocate materials to produce and enable waste reduction Essentially, by adopting an Agile methodology, "Auto Cars" can overcome automotive industry challenges efficiently, ultimately delivering functional, user-friendly, and quality

# References

aadityasharma0115, 2023. *geeksforgeek.* [Online]   
Available at: https://www.geeksforgeeks.org/what-is-kanban/  
[Accessed 02 03 2024].

Anon., 2023. *geeksforgeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/software-engineering-spiral-model/  
[Accessed 21 02 2024].

Anon., n.d. *tryqa.* [Online]   
Available at: https://tryqa.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/  
[Accessed 21 02 2024].

AshmeetSaggu, 2024. *geeksforgeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/non-functional-requirements-in-software-engineering/  
[Accessed 29 03 2024].

Ben Lutkevich, 2021. *techtarget.* [Online]   
Available at: https://www.techtarget.com/searchsoftwarequality/definition/lean-programming  
[Accessed 14 03 2024].

Ben Lutkevich, 2022. *techtarget.* [Online]   
Available at: https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model  
[Accessed 21 2 2024].

Corrales, E., 2022. *developer.* [Online]   
Available at: https://www.developer.com/project-management/what-is-scrum/  
[Accessed 02 03 2024].

creately, 2022. *creately.* [Online]   
Available at: https://creately.com/guides/use-case-diagram-tutorial/  
[Accessed 26 03 2024].

It Service, S., 2022. *inthera.* [Online]   
Available at: https://www.inthera.ca/7-essential-types-of-agile-methodology/  
[Accessed 27 02 2024].

janardan333, 2024. *geeksforgeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/what-are-functional-requirements-in-system-design-examples-definition/  
[Accessed 29 03 2024].

Kashyap, S., 2024. *proofhub.* [Online]   
Available at: https://www.proofhub.com/articles/traditional-vs-agile-project-management  
[Accessed 10 2 2024].

KOVOOR, S., 2023. *guidingcode.* [Online]   
Available at: https://guidingcode.com/user-and-system-requirements-in-software-engineering/  
[Accessed 26 03 2024].

lucidchart., n.d. *lucidchart..* [Online]   
Available at: https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial  
[Accessed 26 03 2024].

lucidchart, n.d. *lucidchart.* [Online]   
Available at: https://www.lucidchart.com/blog/what-is-extreme-programming  
[Accessed 2 03 2024].

lucidchart, n.d. *lucidchart.* [Online]   
Available at: https://www.lucidchart.com/pages/data-flow-diagram  
[Accessed 26 03 2024].

lucidchart, n.d. *lucidchart.* [Online]   
Available at: https://www.lucidchart.com/pages/er-diagrams  
[Accessed 26 03 2024].

Martin, M., 2024. *guru99.* [Online]   
Available at: https://www.guru99.com/software-engineering-prototyping-model.html  
[Accessed 27 02 2024].

mijacobs, et al., 2022. *microsoft.* [Online]   
Available at: https://learn.microsoft.com/en-us/devops/plan/what-is-scrum  
[Accessed 14 03 2024].

naimishsahu08, 2023. *geeksforgeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/software-prototyping-model-and-phases/  
[Accessed 27 02 2024].

PAL, S. K., 2024. *geeksforgeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/software-engineering-extreme-programming-xp/  
[Accessed 02 03 2024].

Rana, K., 2021. *artoftesting..* [Online]   
Available at: https://artoftesting.com/spiral-model  
[Accessed 27 02 2024].

Rosencrance, L., 2019. *techtarget.* [Online]   
Available at: https://www.techtarget.com/searchsoftwarequality/definition/software-requirements-specification  
[Accessed 29 03 2024].

sitesbay., n.d. *sitesbay..* [Online]   
Available at: https://www.sitesbay.com/software-engineering/se-spiral-model  
[Accessed 27 02 2024].

Staff, E., 2024. *computercareers.* [Online]   
Available at: https://www.computercareers.org/lean-software-development-principles/  
[Accessed 14 03 2024].

team, i. e., 2023. *indeed.* [Online]   
Available at: https://www.indeed.com/career-advice/career-development/feasibility-studies  
[Accessed 14 03 2024].