

Name/ID : ABDELKAREEM YOUSEF MAMDOH SOUBAR/19110022

Subject : **Software Development Lifecycles**

**-** 0040201220

Assignment Title: **Building a Software Utilizing a Software Development Lifecycle & Model**

Submission Date : Feb. 1, 2023

Table of Contents

Contents

[**Introduction** 4](#_Toc126182365)

[**Steps for SDLC** 4](#_Toc126182366)

[**Benefits of SDLC** 5](#_Toc126182367)

[**Lifecyle Models** 5](#_Toc126182368)

[**Iterative Lifecyle Model** 5](#_Toc126182369)

[**Agile** 6](#_Toc126182370)

[**Scrum** 7](#_Toc126182371)

[**Sequential Lifecycle Model** 9](#_Toc126182372)

[**Waterfall Model** 9](#_Toc126182373)

[**V Model** 11](#_Toc126182374)

[**Risk Management** 14](#_Toc126182375)

[**Risk Assessment** 14](#_Toc126182376)

[**Treating Risks** 15](#_Toc126182377)

[**The Waterfall Model in Practice** 16](#_Toc126182378)

[**Using the Waterfall Model in Large Projects** 17](#_Toc126182379)

[**A Comparison between Two Technical Solutions** 18](#_Toc126182380)

[**Monopoly** 18](#_Toc126182381)

[**Risk** 19](#_Toc126182382)

[**Conclusion** 20](#_Toc126182383)

[**The Feasibility Report** 21](#_Toc126182384)

[**Feasibility Report Components** 21](#_Toc126182385)

[**Feasibility Report Purpose** 21](#_Toc126182386)

[**Different Feasibility Criteria** 22](#_Toc126182387)

[**Software Behavioral Design Techniques** 23](#_Toc126182388)

[**Flowcharts** 24](#_Toc126182389)

[**Entity Relationship Diagrams** 27](#_Toc126182390)

[**Software Behavioral Design Tools** 31](#_Toc126182391)

[**Draw.io** 32](#_Toc126182392)

[**Microsoft Visio** 33](#_Toc126182393)

[**FSM vs EFSM** 34](#_Toc126182394)

[**FSM** 34](#_Toc126182395)

[**EFSM** 34](#_Toc126182396)

[**Data-driven Software** 35](#_Toc126182397)

[**Game Description** 36](#_Toc126182398)

[**Requirements** 36](#_Toc126182399)

[**User Interface** 37](#_Toc126182400)

[**Player Name** 38](#_Toc126182401)

[**Purchase a Country** 38](#_Toc126182402)

[**Buying a Product** 39](#_Toc126182403)

[**Design Documentation** 40](#_Toc126182404)

[**Class Diagram** 40](#_Toc126182405)

[**Sequence Diagram** 41](#_Toc126182406)

[**Stock exchange** 42](#_Toc126182407)

[**Eliminate Player** 43](#_Toc126182408)

[**Board** 44](#_Toc126182409)

[**Products Pricing** 45](#_Toc126182410)

[**Use Case** 46](#_Toc126182411)

[**Test Results** 47](#_Toc126182412)

[**Game test** 47](#_Toc126182413)

[**Square Test** 47](#_Toc126182414)

[**Stock Exchange** 48](#_Toc126182415)

[**Product Bidding Test** 48](#_Toc126182416)

[**Start of Quarter Test** 49](#_Toc126182417)

[**Product Test** 49](#_Toc126182418)

[**Company Test** 49](#_Toc126182419)

[**Antitrust Test** 50](#_Toc126182420)

[**Players Test** 50](#_Toc126182421)

[**RTM (Requirements Traceability Matrix)** 51](#_Toc126182422)

[**References** 85](#_Toc126182423)

# **Introduction**

In this short section I would like to talk about what is SDLC and what are the benefits of it. SDLC stands for Software Development Life Cycle, its main task is to provide software by following certain steps that enable the software to function at its optimum state.

## **Steps for SDLC**

You may ask yourself what are these main steps that are involved in SDLC, I would like to answer your question in this paragraph. [1]

1. **Project Planning:** In this phase, we answer the number one question in creating software, which is “What do we want?”. Once this question is answered, we can start with further phases. It is essential that we know what we want, because if there is some uncertainty within the idea of the project, I can assure you that it will be hard to proceed with it further.
2. **Gathering Requirements and Analysis:** We meet with the customer to find out what are the functional requirements and what are the nonfunctional requirements. In brief, functional requirements answer the question “How must the system work?” and nonfunctional requirements would answer the question “How should the system perform?”. Functional requirements specify the capabilities, features, and behaviors of a system.

These requirements describe what the system should do, what inputs it should accept, and what outputs it should produce. Some examples of functional requirements include:

User authentication: the system must provide a mechanism for users to log in and access their accounts.

Data storage: the system must be able to store and retrieve data from a database.

Search functionality: the system must provide a search function that allows users to search for specific items in a database.

Reporting: the system must generate reports based on specific criteria.

On the other hand, non-functional requirements define the quality attributes that a system must possess. These requirements describe how well the system performs its functional requirements and specify the constraints under which it operates. Some examples of non-functional requirements include:

Performance: the system must respond to user requests within a certain time frame, such as 2 seconds.

Security: the system must ensure that sensitive data is protected from unauthorized access and manipulation.

Scalability: the system must be able to handle an increasing number of users and transactions without degrading performance.

Usability: the system must be easy to use and navigate for users with different levels of technical proficiency.

It is important to consider both functional and non-functional requirements when designing and developing a system. Functional requirements define what the system does, while non-functional requirements define how well the system does it. Together, they provide a comprehensive picture of the desired system and help ensure that the final product meets the needs of the users and stakeholders.

[2]

1. **Design:** In the design phase we transform data that was gathered from the previous two phases into a prototype if you can call it that way. A design can include wireframes, UML diagrams, flowcharts, etc.…[3]
2. **Coding/Implementation:** After getting set with all of the information that we have, coding takes place. Coding is considered to be one of the longest phases in SDLC, this is because we must ensure the integrity of our software and follow certain design guidelines that are within the professional industry standards as well as the company’s standards.
3. **Testing:** No software is complete without having it tested. Tests can be automated or manual, in which a whole quality assurance team would be hired to test the software. This phase is very important, in which it can show some flaws that were not thought of as a programmer’s point-of-view.
4. **Deployment:** This phase is all about deploying the produced software to the customer. It is really necessary to get feedback from the customer and follow up with him/her.

Of course, the software that is deployed successfully met all of the requirements with the tests passed.

1. **Maintenance:** If any bug has been discovered by the customer, he/she will inform us as soon as possible. A certain bug could show from a customer’s point-of-view not a programmer’s point-of-view, since a programmer didn’t really think that much regarding this bug or how it could be produced.

## **Benefits of SDLC**

There are many benefits of using SDLC, I would like to point out a few! [4]

* It makes all parties engaged in the development process more aware of every stage of the life cycle.
* It facilitates project management and control.
* It helps with project estimating, planning, and scheduling.
* Project hazards are decreased.

# **Lifecyle Models**

In this part of this report, I would like to discuss two main categories of software lifecycle models which are iterative and sequential models. I will talk about the properties of each one and also give examples of them. [5]

## **Iterative Lifecyle Model**

A procedure where development is carried out in brief, repeating cycles, or iterations, is known as an iterative software development lifecycle. The goal is to quickly provide a functional version of the program so that it may be enhanced in later iterations. Planning, design, development, testing, and deployment processes are all included in each iteration. Up till the desired program is created, the procedure is repeated. Throughout the process, input is gathered, and the software is continually enhanced in response to it. With this strategy, it is possible to be flexible and alter courses as necessary.

There are two main examples that are under the tree of iterative lifecycle model, agile and scrum.[5]

### **Agile**

Agile is an iterative development process that places an emphasis on adaptability, teamwork, and quick iteration. Agile teams complete a tiny piece of functionality's planning, design, development, testing, and deployment in brief sprints, which are generally 2-4 weeks long. The team then evaluates the sprint's work, collects feedback, and utilizes it to plan the following sprint. Until the full piece of software is created, this procedure is repeated. [5]

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Customer satisfaction is rapid. | It is not a document-based method for creating software. |
| The product is frequently delivered. | Since this model can adapt to changes, it is very unclear to predict what the outcome would be. |
| Regular tolerance to shifting conditions. | Estimating the amount of work and resources is challenging. |



### **Scrum**

Scrum is an iterative method that focuses on managing iterative development rather than specific agile practices. Because it is easy to use, adaptable, and produces results rapidly, it is especially well-liked in the software development industry. It uses diverse teams to quickly iterate through functioning software to build (sprints). [6,7]

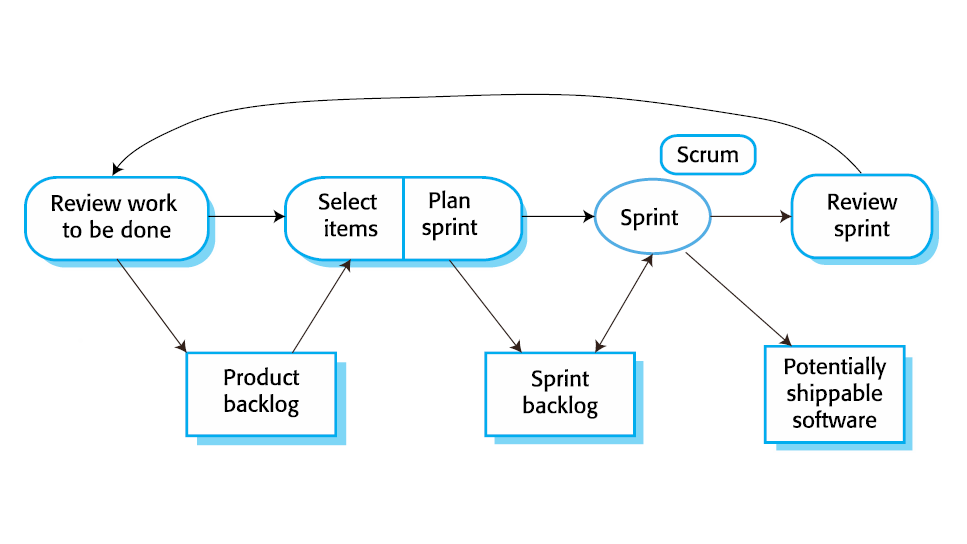
There are three main phases when using the scrum method in SDLC, I would like to give an outline to them.

* The initial phase is an outline planning phase where you establish the general objectives for the project and design the software architecture.
* After this step, it is followed by a series of sprint cycles in which each sprint could take two to four weeks to accomplish. Each cycle develops an “increment” of the system.
* The project closure phase wraps up the project, completes required documentation such as system help frames and user manuals and also assesses the lessons learned from the project.

In the following table, I would like to discuss some key regarding the scrum method in software lifecycle. [7]

|  |  |
| --- | --- |
| **Scrum Term** | **Definition** |
| **Scrum** | It is a daily meeting that reviews the progress and prioritizes work to be done that day |
| **Scrum Master** | The Scrum Master is responsible for ensuring that the scrum process is followed and guides the team in the effective use of scrum |
| **Product Backlog** | It is a list of ‘to do’ items that the scrum team must handle and solve. |
| **Product Owner** | An induvial or a small group whose job is to identify product features or requirements. |

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| The scrum methodology ensures the integrity of using time and money resources. | To be in the scrum framework, intensive experience and training must be involved, making it sometimes hard and expensive. |
| It is a good option for projects with interchanging variables and conditions. | When implementing this style, project deadlines must not be strict. |
| Due to having scrum meetings, progress is well tracked. | Scrum meetings can sometimes affect people working on the project in a negative aspect. |



## **Sequential Lifecycle Model**

A sequential lifecycle is the traditional way of software development, in which they are divided into certain phases. You can classify a sequential lifecycle as a linear model, this is because you cannot go back to a previous phase. If you want to go to the next phase, you need to complete the phase that is prerequisite to it. This type of model is very useful when the project requirements are well-defined and there are no variables to be changed. However, if there are any changing variables, it is very hard to change them, since this model cannot tolerate any flexibility.[6, 7]

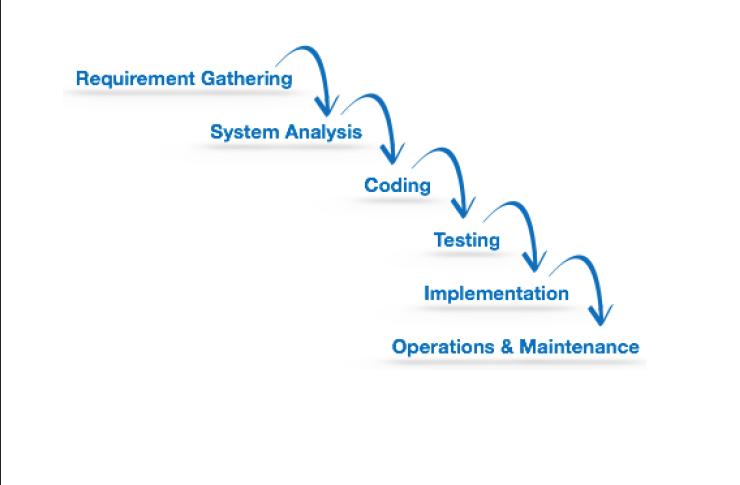
I would like to address two main examples that are under the sequential lifecycle categories. (Waterfall and V Models)

### **Waterfall Model**

In the early stages of the software engineering era, the waterfall was the most commonly used method, but nowadays it is not that commonly used. However, this model is the basis of every other model that exists. This model is quite easy and convenient to use and understand. There are several reasons that make the waterfall model easy to use and understand.[7]

* **Linear and Sequential:** Since the waterfall model is linear and sequential, it would provide a comprehensible and logical progression of steps. This is because each phase of the development process must be completed before moving on to the next phase.
* **Document-Driven:** Unlike scrum, this model relies heavily on documentation. Documentation really helps with making things clear and keeping track of every step that is produced.
* **Suitable for Well-defined Projects:** In an environment where each phase is clear and well-defined, where variables are not prone to any variations, it is more preferred to use the waterfall model.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Strict planning of the project development structure ensures that the number of obstacles that can be faced is decreased | If any of the project requirements need to be changed, it is very hard to go back and modify it. |
| Due to heavy documentation, any information related to the project is documented. This makes it easy for any new members to get track of what is going on. | Unlike agile and scrum methodologies, no prototypes can be provided to the customer during the phases. So, the customer’s feedback is not well thought off. |
| The paradigm is strict, making management easy because each phase has distinct deliverables and a review process. | It can provide a high risk of uncertainty, this is because if a mistake is made during one of the phases, it cannot be resolved in the next phase! |



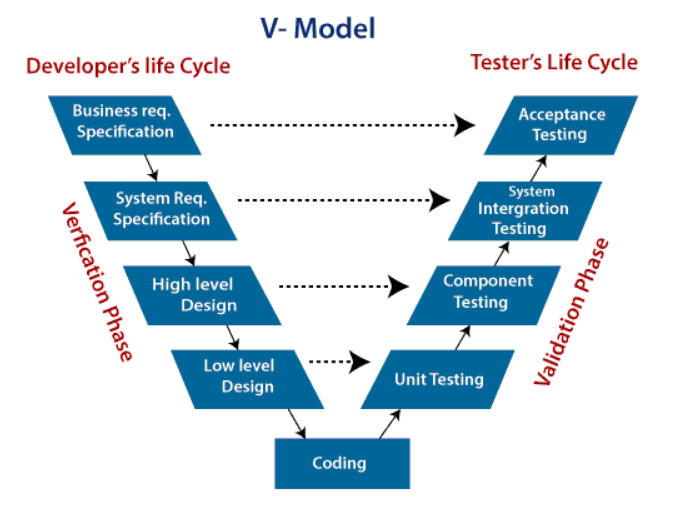
### **V Model**

V-Model is one of the famous types of sequential lifecycle models. It is considered as a strict methodology when creating certain software because of its sequential and highly structured characteristics. You may ask yourself, why is it called a V-Model? The answer to that question is that each phase of the development process corresponds to a specific stage in the product’s lifecycle. The V-model is employed to make sure that every step required to create a high-quality software product that satisfies the needs of the end-user is taken.[6]

There are two main phases in the V-Model, in which they’re the verification and the validation phase. They are next to each other, forming the letter V. Verification involves static analysis. Static analysis is a method of debugging without actually running the program. The verification definition is that the process of evaluating the product development process is done to see whether the requirements are met. On the other hand, validation involves a dynamic analysis method, in which it means testing and evaluating is done while the software is running. Validation is the step in which it can verify that the software has met the customer’s expectations and requirements.[7]

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| The model is easy to manage since each phase is well-defined and clear. | Since changes in the requirements is hard, it makes it unsuitable to make any other changes, making it inflexible and strict. |
| It is faster than the waterfall model, because the V-Model consumes less time to develop and come out with outcomes in the development cycle. | Due to the lack of providing prototypes to the customer, it sometimes can be challenging and hard to meet the exact customer’s expectations. This can be caused due to any miscommunication by either parties. |
| Advanced error tracking is within the basis of the V-Model, which helps the people that are responsible in this project provide a bug-free product while consuming the least time possible. | If any changes are needed within the software, every document that is involved within the project must be rewritten. |

As the V-Model is a little bit more complicated than the typical waterfall model, I would like to explain its diagram in brief.



As the graphic above illustrates, we have two phases which are verification and a validation phase.

The verification phase has the following steps:[8]

* **Business Requirement Specification:** In this step, product requirements are discussed with the customer, clear communication must be a part within meetings.
* **System Requirement Specification:** System engineers examine and study the proposed system by carefully reading the user requirement document.
* **High Level Design:** This phase can be also called “Architecture Design”, within this stage, it is clear how would the data transfer and communication between internal modules and other systems be processed.
* **Low Level Design:** This phase can be also called “Module Design”, in which the system is divided into small “modules”. Detailed internal design for all system modules is specified, it is very crucial that the design is appropriate to be used with other modules and external entities.
* **Coding Phase:** After getting done with the High Level Design and Low Level Design phases, we start with the coding phase. A certain programming language is determined with guidelines set.

After we are done with the verification phase, I would like to talk about the validation phase.Keep in mind that every step in the validation phase is parallel to a corresponding verification phase. [9]

* **Unit Testing:** Unit testing is present within the Low Level Design phase (Module Design Phase). Tests are provided in the code level, in which it can help eliminate bugs that are present. But one thing to keep in mind is that not all bugs are eliminated in this stage.
* **Component Testing:** Component testing can be also named as “Integration Testing”, is present within the High Level Design phase (Architecture Design Phase). The main function of component testing is to test out the communication between internal modules and to verify that every is up to standard.
* **System Integration Testing:** As illustrated in the figure above, this phase corresponds directly to the System Requirement Speciation Phase. This type of testing would examine the entire system functionality as well as communication with other external systems. This phase is tested by the customer’s business team.
* **Acceptance Testing:** Finally, this phase corresponds to the Business Requirement Phase. The test is run by the point-of-view of the user’s atmosphere. Non-functional requirements-related issues can also be shown, such as the performance of the system. [10]

# **Risk Management**

To begin with, we should define what are risks? Risks are anything that can negatively impact the flow of our project that can have a harmful effect. Risk management is a very important topic to discuss. It is not only important in SDLC, but it is also important in other businesses. In this part of the report, I would like to talk about how risks are assessed and handled. [11]

Risk management has some tasks that are followed, with these steps and knowing each step what it does, I can assure you a successful risk management plan. [11]

* Identify risks and their triggers.
* Classify and prioritize all risks.
* Create a plan that links each risk to mitigation.
* Monitor for risk triggers.
* Implement the mitigating action.
* Communicate risk status

Risks can be defined as the following, here is a few: [12]

* **New Unproven Technologies**: In this era, technology has evolved in an inflated manner. And let us be honest, we all like to implement new technologies. But if there was not enough knowledge and enough documentation within a new technology, this can lead to the failure of the project.
* **Application and System Architecture:** If we chose an inadequate system architecture regarding a certain project, this could decrease the functionality of the software, making it a risk that should be avoided.
* **Organizational:** Sometimes the atmosphere when working on a certain project can tension up. This can lead to a decrease in the workflow of the project, thus taking it longer to complete it. Also, another point to consider is that the quality of the written code can also decrease, which would cost more effort, time, and money to fix it and the reputation of the endorsement could be affected also.

## **Risk Assessment**

Risk assessment is the process to identify potential hazards and analyze what could happen if a hazard occurs. We have several risk assessment techniques that can be used to identify risks in which we can decrease the severity of their effect. [12]

* **Threat Modelling:** This entails detecting possible dangers to the software project and assessing their likelihood and effect.
* **Risk Analysis:** In this stage, we would analyze the likelihood and the impact of what a risk could negatively affect our project process.
* **Risk Response Planning:** This stage is all about how we would control and try to solve upcoming risks using the optimum techniques that would be provided.
* **Risk Monitoring and Control:** The monitoring process of any upcoming risks are included in this stage.

## **Treating Risks**

Risks can be treated in many several ways, this plays a factor on the severity of the risk’s impact, the budget of the customer, and the likelihood of risk’s existence. I would like to talk about some ways and methods that we can mitigate risks. [14]

* **Risk Avoidance:** Sometimes if the likelihood of such a risk is apparent, we could change some properties of our project that can decrease the likelihood of that risk.
* **Risk Transfer:** If a certain organization does not have the certain capabilities to handle a certain type of risk, risk transfer can be used. This means that if a risk happens, we can get assistance from another party that could help resolve this type of risk.
* **Risk Control:** This type of risk treatment would mean taking several actions that would help prevent the escalation of that risk and its consequences.
* **Risk Acceptance:** By its name, we would accept that plan and try to develop a certain plan that would help mitigate the risk. We wouldn’t change any properties of the project, but a plan would be made to assist us in resolving the risk.

# **The Waterfall Model in Practice**

Although we have several software modelling practices, I chose to represent the waterfall model when we want to design a car. The reason on why I chose to model, is that that when you want to design and produce a car, all the steps are sequential, as well as we use the waterfall to reduce the probability of errors and rework. What I mean by all the steps are sequential is that the we need to complete the step that we are in to progress to the next phase. For example, you cannot assemble and create a prototype without actually designing the car. Of course, prototypes change, but there must be a general idea of how the car would look like. [14]

I would like to talk about the steps when using the waterfall model when creating the car.

* **Analysis:** In the analysis phase, we “analyze” what is the body type of it, what is the targeted market, identify safety and emissions regulations. Once these questions are asked we can proceed to the next stage.
* **Design:** After we finished from the analysis phase, we would find ourselves in the design phase. At this phase, we determine the exterior shape, how would the inside of the car be, what are the offered engine types.
* **Implementation:** After we have completed the analysis and the design phases, we would actually work on creating the car.
* **Testing:** After choosing the optimum prototype that fits well with the required criteria, we can test it. The tests include crash tests, durability testing, and the general performance of the car. If we noticed any flowbacks, we would modify the prototype and test it again.
* **Maintenance:** We would provide maintenance to the car, in which it would not have any major problems that would defect the usage of the car.

# **Using the Waterfall Model in Large Projects**

Although we discussed a lot of different ways and methodologies that enable us to perform software development in an efficient manner, the waterfall model stays one of the most used in large software development projects, why is that? In this chapter I would like to talk about the usage of the waterfall model in large scales. [25, 18]

Based on what was mentioned in a few chapters, let us have a recap on the properties of the waterfall model. As we know, the waterfall model is a sequential type of model, it means that we cannot go back to a previous step, nor we can get to the next step unless we finished the step we are in.[19]

The waterfall model is used in large projects for the following reasons:

* **Clear Structure:**  By this statement, I mean that the steps that we would follow in this model are very clear. We have predefined steps that we would follow and that’s it, the steps are as followed: (Requirement Gathering, System Analysis, Coding, Testing, Implementation, Operations and Maintenance).
* **Well Documented:**  Unlike the agile methodology, the waterfall model relies heavily on documentation. This is a good method of keeping track of what is going on through our project and following what technologies and design patterns are we using. Also, if we face a problem throughout the testing process, we would definitely know how to solve it, this is because our project processes have been well-documented.
* **No Confusion:** What I would mean by the no confusion statement, is that there would be any uncertainty within the phases that we are in. For example, we would be in the coding phase, it wouldn’t be suitable that we would jump to the next phase (testing phase) while we still didn’t finish coding for the project.
* **Easily Managed:** When we have a project on a large scale, we would involve a lot of employees from different expertise to construct this project. The waterfall model enables us to follow these phases in a clear manner, making it possible to be followed by nearly everyone in the foundation.

# **A Comparison between Two Technical Solutions**

I would like to compare the feasibility studies of two similar games, Monopoly and Risk. In this part of the report, I would like to talk about each one and what are the feasibility criteria of each. The idea of both games consists of the idea of getting a part of land and buying it. Of course, there are some differences, but generally they are quite similar.

## **Monopoly**

A feasibility study on a Monopoly game application, with a proposed idea and based on IEEE 830 (or a similar standard), would involve the following steps:

* **Market Demand Analysis:** Evaluate the demand for mobile gaming and board games, specifically Monopoly, to determine if there is a market for the game application.
* **Technical Requirements:** Define the technical requirements for the development of the Monopoly game application, including platform compatibility, graphics, and user interface, in accordance with IEEE 830.
* **Software Requirement Specification (SRS):** Develop an SRS that outlines the functional and non-functional requirements for the Monopoly game application, based on market demand and technical requirements, in accordance with IEEE 830.
* **Development Costs:** Estimate the costs of developing, testing, and launching the Monopoly game application, including the costs of hiring developers, designers, and testers, and the cost of any necessary software development tools.
* **Revenue Potential:** Forecast the potential revenue from sales and in-app purchases, based on market demand and the features offered in the application.
* **Competition:** Analyze the competition in the market, including other Monopoly game applications, to determine the potential for success.
* **Risk Analysis:** Identify and assess the risks associated with developing and launching the Monopoly game application, such as changes in consumer preferences, technical challenges, and market competition, in accordance with IEEE 830.

Based on the results of this analysis, a conclusion can be drawn regarding the feasibility of developing and launching the Monopoly game application, including recommendations for improving its viability, in accordance with IEEE 830.

## **Risk**

A feasibility study on a risk game application, with a proposed idea and based on IEEE 830 (or a similar standard), would involve evaluating the following aspects to determine its viability and potential for success:

* **Market Demand:** Evaluate the demand for mobile gaming and strategy games, specifically risk games, to determine if there is a market for the game application. Conduct market research to gather data and insights on the size and characteristics of the target audience, their preferences and behaviors, and the trends in the mobile gaming industry. This analysis should include a review of similar risk games and their popularity, as well as an assessment of any unmet needs or gaps in the market that the proposed game application can fill.
* **Technical Requirements:** Determine the technical requirements for the development of the risk game application, including the platform it will be compatible with, the graphics and user interface required, and any other technical specifications. These should be based on the SRS (Software Requirement Specification) as outlined in IEEE 830 or a similar standard. The technical analysis should consider the feasibility of implementing the proposed features, the cost and availability of necessary development tools and resources, and the compatibility with existing systems and technologies.
* **Development Costs:** Estimate the costs of developing, testing, and launching the risk game application, including the costs of hiring developers, designers, and testers, the cost of any necessary software development tools, and any other expenses related to the development process. This analysis should include a breakdown of costs by phase, as well as a realistic timeline for the development process. The budget should be adequate to support the development of the game application and provide a margin for contingencies.
* **Revenue Potential:** Forecast the potential revenue from sales and in-app purchases, based on market demand, the features offered in the application, and the pricing strategies adopted. Consider different pricing models, such as a one-time purchase or a subscription-based model, to maximize revenue potential. The revenue analysis should consider the costs of marketing and promoting the game application, as well as any potential barriers to adoption or competition in the market.
* **Competition:** Analyze the competition in the market, including other risk game applications, to determine the potential for success. Consider factors such as the features and functionality offered by competing games, their popularity, and user reviews. This analysis should identify areas where the proposed risk game application can differentiate itself from the competition and provide a competitive advantage.
* **Risk Analysis:** Identify and assess the risks associated with developing and launching the risk game application, such as changes in consumer preferences, technical challenges, and market competition. Develop mitigation strategies to address the identified risks and minimize the potential impact on the viability and success of the game application. The risk analysis should consider the potential consequences of not launching the game application, as well as the impact of external factors, such as economic conditions, regulatory environment, and technological advancements.
* **Conclusion and Recommendations:** Based on the results of the analysis, draw a conclusion regarding the feasibility of developing and launching the risk game application. This conclusion should summarize the findings and provide recommendations for improving the viability and success of the game application. The recommendations should address the strengths and weaknesses of the proposal and suggest alternative solutions or options if necessary.

The feasibility study should provide a comprehensive and detailed assessment of the proposed risk game application, allowing stakeholders to make informed decisions about its development and launch. The study should be regularly reviewed and updated to ensure that it remains relevant and reflects any changes in the market, technology, or other factors that may impact the viability of the game application.

## **Conclusion**

As was covered from the two previous parts in this chapter, I would like to summarize which is more probable to succeed in the market. Overall both games provide similar strategies and gameplay experience with Risk being a little bit more complicated. Monopoly has a higher fanbase, in which it was known from years ago and whenever someone brings the idea of a board game, it is most likely that someone would think of Monopoly. So in conclusion, I feel that the digital version of the game Monopoly would succeed more than the game Risk.

# **The Feasibility Report**

To start things off, we need to define what is a feasibility study. A feasibility study is done to determine whether the project that is to be started is feasible. What we mean by a project being feasible is that we can produce an outcome without having too many difficulties while producing it or is it beneficial to start working with it. [22]

## **Feasibility Report Components**

To write a well-documented feasibility report, we have a set of essential components that should be involved in order to write a successful feasibility report. [26]

* **Project Description:** This is a very important step to begin with, this would include the purpose of our project. It is very important to know the purpose of our project, this is because it helps us to get an idea of what our project is. Also when knowing what is the purpose of our project, it can give out a higher performance when working on our project.
* **Market Analysis:** A market analysis is performed to know the target audience and how they would react to our project. It would show how successful our project would be if implemented to the general public.
* **Technical Specifications:** What we mean by technical specifications is mentioning all of the technical details that we used. This can include the programming languages that we used, which architecture method approach was used, and different security measures.
* **Risk Analysis:** As mentioned in a previous section in this report, risk analysis is one of the most important things to do in a project. Risk analysis is used to inform the customers of certain risks that we can face throughout the process of creating certain software and how we could mitigate those apparent risks.
* **Conclusion:** We must write a summary of the purpose of the project.

## **Feasibility Report Purpose**

There are various advantages when creating a feasibility report. Here is a few: [28]

* It would show the advantages and disadvantages when creating a certain project. In which a feasibility report would show us details of a project in a summarized form rather than going deeply in analyzing and going through it.
* A feasibility report could show if the project is risky or not. If the project had a high-risk factor, this could help us not to start with it. If we started with this project and it failed we would have lost some precious time, effort, and money that we could have used on another more successful project.
* A good feasibility report would also be used to convince investors to invest in our product. This could mean that we would have more financial resources to help us improve our product.

## **Different Feasibility Criteria**

When conducting a feasibility study, we should include certain key aspects that would make our feasibility report a successful one. [31]

* **Economic Feasibility:** Every project has a budget. Can this project be made with this budget or not? How good would this software be? These are some questions to ask yourself we are conducting the economic feasibility. Economic Feasibility would have the following parts:
  + The economic benefits from the project.
  + The economic costs of the project.
  + The net economic benefit, in which it can show if we benefited from it or not.
* **Schedule Feasibility:** As we go by the saying “Time is Gold,” time is a particularly important factor to consider. Would we be able to deliver functioning software that meets the customer’s requirements by the deadline? Bear in mind that a project will fail if not delivered by the required deadline! A detailed scheduling process and session should be considered.
* **Operational Feasibility:** When we are accepting the customer’s requirements for the software, we should be certain that we can deliver those requirements without any difficulties. Also, how easily can we maintain this type of technology after deployment? These are some questions to be considered and studied. In other terms, operational feasibility explains how efficiently we could use resources to give out a working product according to the customer’s requirements.
* **Legal Feasibility:** When we are creating any software, we should know that the user’s private information is not violated as well as how our software confines with international cyberlaws. If we sensed that a certain project violates some laws, we would immediately terminate it, this is because software should not be used in an unethical manner. [32]
* **Technical Feasibility:** Technical feasibility is a study of whether our hardware and software resources are enough to develop and run such a project. This could also include how our staff would be capable of creating this project in a technical manner. In summary, we could describe technical feasibility as analyzing our resources to produce a project in a successful manner, these resources can include software, hardware, and our technical teams.

# **Software Behavioral Design Techniques**

In the process of designing a certain software, we can use certain diagrams and techniques that would allow us to thoroughly create this software. These diagrams are very good to have, in which they would guide us when creating the software and they would also reduce the percentage of uncertainty of not knowing how would the software react/work in the real world, under the hands of the general public. [33]

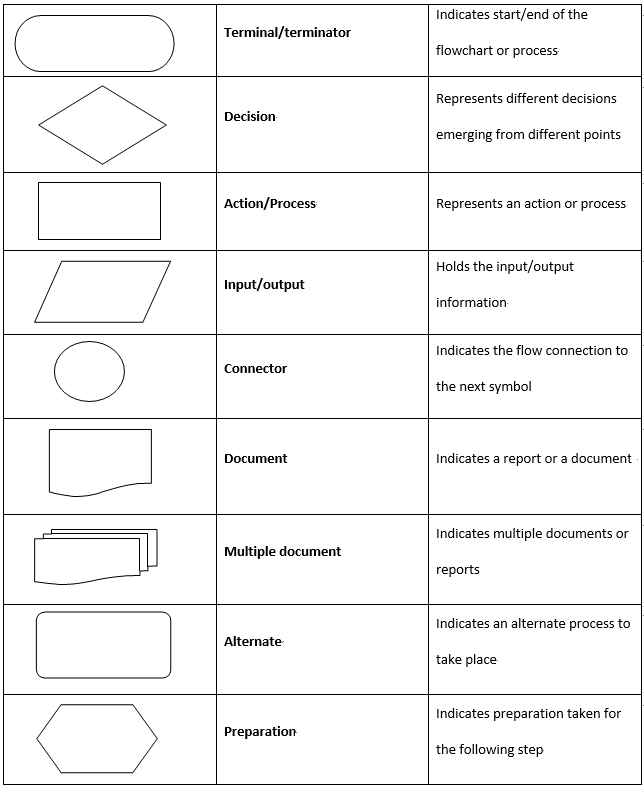
We have several diagrams and techniques that could be used, and I can assure you that every software that is created in this era has them all in their documentation. I would like to count a few: [36]

* **Flowcharts.**
* **Pseudocodes.**
* **Finite and Extended State Machine.**
* **Data Flow Diagrams.**
* **Entity Relation Diagrams.**

In this report, I would like to talk in detail about two of these techniques, which are (Flowcharts and Entity Relation Diagrams). The reason why I chose these two techniques is that I felt that they would describe how the system would react and how different data members are related to each other.[38]

## **Flowcharts**

To begin with, we should define what is flowchart and what is its purpose. A flowchart is a type of diagram that would show the sequential flow of steps for a certain algorithm, process, or workflow. Flowcharts contain a variety of shapes that would represent it, for example rectangles, ovals, diamonds which are then connected using arrows to show how they are connected to each other. Each shape has a different meaning, I would like to include an image that would illustrate those different shapes. [40]



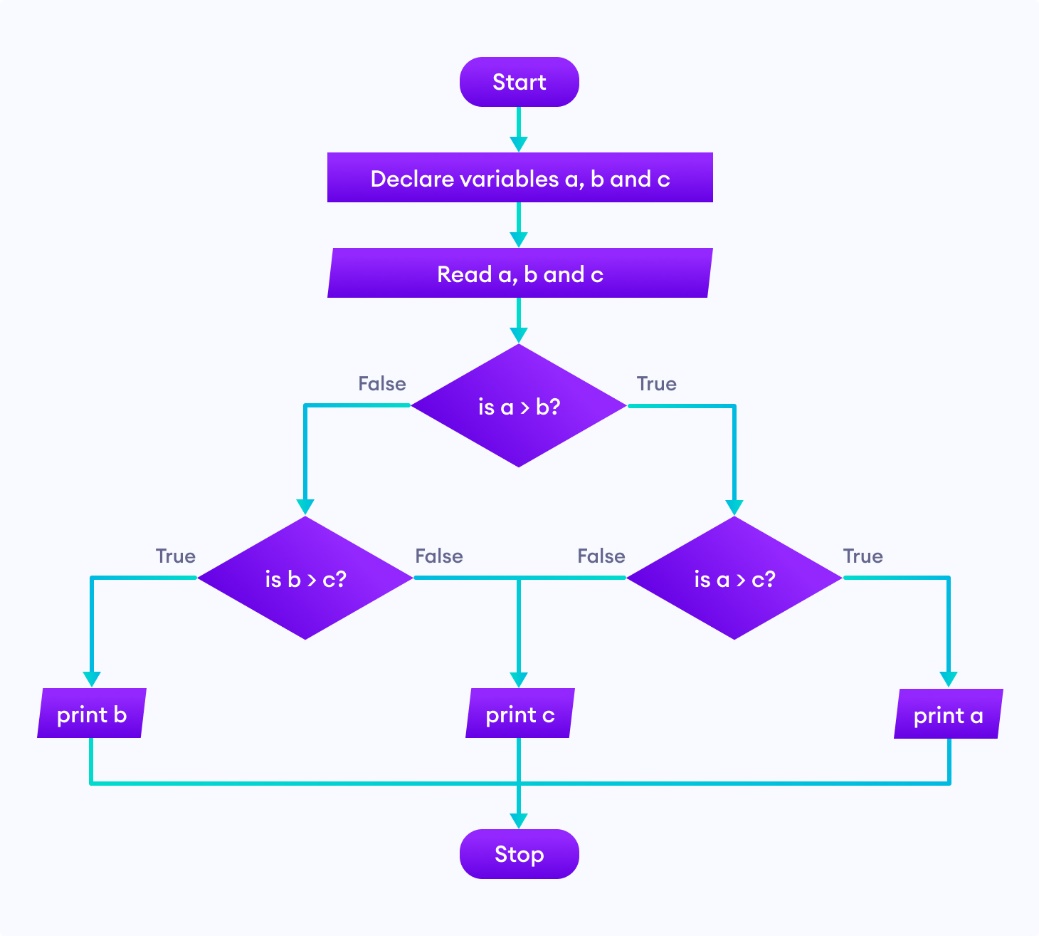
Flowcharts can be drawn to show a general outline of a process/algorithm or can be drawn in great detail that would show the tiniest steps that are involved. This of course depends on what the flowchart represent, is it a sophisticated process/algorithm?. I would like to talk about the general steps when creating a flowchart, which are: [40]

* **Define the Purpose and Scope:** When creating a certain flowchart, we must fully know why we created this flowchart. Is it for a process? Is it for a certain algorithm that is used in our software? In which detail should we draw it? This step is an especially important step, in which when we have definite tasks to perform, this would make our lives much easier.
* **Identify Steps in Chronological Order:** Since this type of diagram is considered as a sequential type-of diagram, we should fully know the steps in order. It would not really make sense if we were in step #5, and we magically jumped to step #8.
* **Draw the Flowchart:** After setting out the pervious steps in a clear manner, we would actually start drawing the chart. This could be done as a simple sketch by hand, or we would also use specific software that could be used to illustrate it.
* **Confirm the Flowchart:** After drawing out the actual flowchart, we would go through it to ensure the integrity of the shapes and steps drawn. If we sensed that there are missing steps, we would redraw it and go through it from the beginning to the end another time.

What are some rules that must be followed to create a successful flowchart? We have four main rules that are present. [40]

1. The flowchart opening statement must include a “start” keyword.
2. The flowchart ending statement must include an “end” keyword.
3. All symbols in the flowchart must be connected with an arrow line.
4. The decision symbol must not be associated with the arrow line.

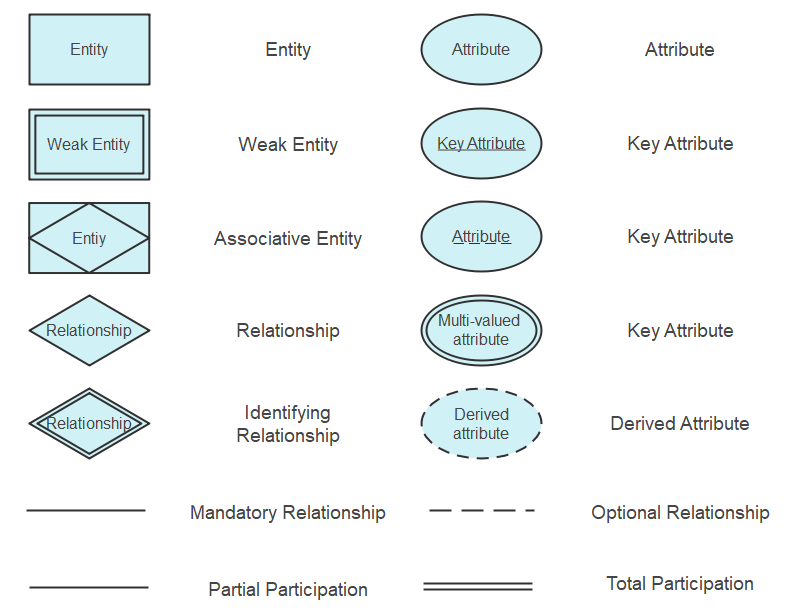
|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Easy to make and understand. | Although it is easy to make, it can be tricky when used to represent complex and large systems. |
| Flowcharts can provide a better understanding of the proposed system. | If any problem in it occurs, it would mean reproducing it from the start up. Which can be a time-consuming process. |
| Identifying mistakes can be easier. This can show any mistakes in the sense of logic in the proposed system. | When making flowcharts, there are no set of standards to determine the level of detail. |



In this graphic, it would show a flowchart of a program that would determine the largest number amongst three numbers.

## **Entity Relationship Diagrams**

An ERD (Entity Relationship Diagram) is a type of diagram that is used to represent and show how different tables in a database are connected to each other, the structure of the database, and the fields that are involved. Similar to flowcharts, ERD has a set of different symbols that help us define a database, I would like to provide a photo that justify this statement. [45]



When creating an ERD, we have a set of components that we should consider as well as certain rules that would help us create a strong database model. I would like to define some terms that are used in ERDs. [45]

* **Entity:** You can think about an entity as a thing, place, person, or object that is totally independent of each other, in which they would represent data components within a database.
* **Attributes:** An attribute is the set of properties that describe a certain entity. An attribute could include a variety of things, such as ID, gender, date-of-birth, first name, last name, etc..
* **Relationships:** Relationships would show how entities are related to each other as well as how they would communicate. In which it can be used to reduce redundant data and improve table structures. We have three types of relationships, which are:
  + **One-to-One:** In this kind of relationship, one entity in a table can relate only once to another table.
  + **One-to-Many:** In this kind of relationship, one record from a table would interact with one or more records from another table.
  + **Many-to-Many:** In this kind of relationship, many records in a table are associated with multiple records from another table.

**Diagram

Description automatically generated**

Any type of diagram would have certain rules that would be followed in order to create a successful and error-free diagram. In this case, we have some rules when creating an ERD. [45]

* **Provide Proper Naming for Entities:** When naming an entity, we should avoid using vague naming and we should always try to use nouns. However, we can sometimes use adjectives to differentiate like-nouns, for example part-time employee and full-time employee.
* **Avoid Unnecessary Relationships:** Unnecessary relationships could create uncertainty and duplication of data. These false data could result in a false creation of queries.
* **Keep the Diagram Clear:** If the diagram is not easy to read and there is some overlapping in between the tables, it would be tricky to understand the database. Thus, causing a possibility for errors and uncertainty.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Generally, it is a simple model to use. We should only know what are the relationships that are present. Then we could draw the diagram easily. | It is fairly complicated when we want to represent any data manipulation. |
| Can be easily transformed into any other data model. These other data models can include hierarchical data model and network data model. | Like flowcharts, there is no standards to follow when creating the ERD. This could cause certain confusion amongst programmers that would read it. |
| Can provide better visual representation. This can mean a better representation of the relationships amongst entities. | ERD models provide a limited representation of relationships. |

Diagram

Description automatically generated

In this graphic, we would see how an ERD of a student enrollment system would be represented.

# **Software Behavioral Design Tools**

Different software tools are used to draw and illustrate those diagrams mentioned above. They are not only limited to these mentioned above, but endless types of diagrams can also be drawn. In this part of the report, I would like to talk about two famous tools that enterprises/companies use when they would like to create those important diagrams. I would like to talk about Draw.io and Microsoft Visio. [47]

## **Draw.io**

Draw.io is an open-source tool that is used to create diagrams that are related to modelling and illustrating diagrams that are related to software creation. It is considered a budget friendly option, since it offers its services for free, of course there are some limitations and disadvantages when using such a tool. But to start things off as a mediocre developer/company that has limited budget, it is a great option. [48]

Each program has its advantages and disadvantages, in the following table I would like to discuss some of them.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| It is cross-platform compatible. Meaning that it can be used from any machine using any operating system (Windows, MacOs, or Linux). | Compared to other more advanced tools, Draw.io is not a suitable option for complex and bigger projects, since it has only a limited number of shapes. |
| Real-time collaboration is one of the nice features that are included in Draw.io. Real-time collaboration is a feature that enables multiple user to access and work on a single project (for example an ERD). | After reading some reviews on the functionality of Draw.io, it has shown that some people had some problems uploading their project files from another diagraming software to Draw.io. This could cause a problem since there could be an integration gap. |
| Multiple integration technologies. Draw.io does not only use their technologies and restrict the user to it, however they can provide some flexibility. Some of these technologies include Google Drive, Confluence, and Jira. | Since it is a free tool, of course it cannot inhibit all of the advanced features that are covered by more sophisticated systems. These advanced features could include automation, data linking, and reporting. |

## **Microsoft Visio**

Microsoft Visio is one of the most popular diagramming software that is used on a more advanced level. By its name, it is endorsed by the company Microsoft, in which it would require a subscription. With such a complex system, Microsoft Visio requires a hefty subscription, some might question the offered price, but I am sure that it is a nice investment to have for complex software design. [47]

In the following table, I would like to talk about the advantages and disadvantages of Microsoft Visio.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Microsoft Visio is used in a high-grade standard level; hence it would produce high-quality diagrams. | Since it is only Microsoft endorsed software, this means that we need Windows-running machines to be able to run the program. |
| Since it is from a Microsoft endorsement, integration with other existing Microsoft programs works flawlessly. | Compared to Draw.io, it has limited collaboration options. Thus, making it obligatory to stick with the available collaboration options, which are probably Microsoft products. |
| Since Microsoft Visio is used in more complex software diagrams, it has the ability to provide automated diagraming. Automated diagramming is important when we have dynamic variables. | Since it is a big and complex diagraming program, it is very hard to learn it all at once. This means that it has a steep learning curve and needs some training to use it. This training could cost the company time and money, which is not sometimes feasible in addition to the price of the license of the software. |

# **FSM vs EFSM**

We have two main state machines types, which are FSM (Finite State Machines) and EFSM (Extended Finite State Machines). Both are used to represent certain states that a software can cover, but there are some differences I would like to cover further on. We need to define what is a state machine diagram. It is a type of diagram that shows the sequence of events that an object goes through, in which it goes through a set of inputs and changes states depending on those given inputs. In other words, state machine diagrams model dynamic behavior.[49]

## **FSM**

An FSM stands for a Finite State Machine is a type of model that is used to design and analyze how would a system react within a set of given inputs, in which it performs a “transition” from one state to another state. In FSM each transition has input Boolean conditions and output Boolean functions linked to it. In FSM only one single state in the system can be active. [49]

## **EFSM**

On the other hand, we have something called an EFSM. EFSM stands for as an Extended Finite State Machine. Unlike FSM, EFSM can provide a more detailed representation of states. Since FSM uses Boolean conditions as an input, EFSM uses an "if statement" with a number of trigger conditions that can be used to express the transition.[49]

# **Data-driven Software**

A software design approach known as data-driven software places a greater emphasis on data input than on pre-set rules created by programmers or by human interaction. Data-driven software approach differs from the traditional software design approaches in several aspects. One of these aspects is that there is always room for improvement, what I mean by this sentence is that it always gathers and processes data to be used for improving the software. Other software may only collect data at certain stages of their development. Also, one way that data-driven software is different than typical software development approaches is that they automatically adapt and reshape itself based on varying user behaviors, while other software design approaches need to be updated at some different intervals to be suitable for various user behavior. [43]

Data-driven software is present in several different fields such as streaming companies, ride hailing applications, and prediction analytics systems. All of these industries need dynamic data to be fed into the system and as a result of that, it would provide a better and tailored user experience (Netflix for example). [43]

Why do we use data-driven software in certain fields rather than the different methods that are available?

* It provides a better user experience. That is because it collects data from user interactions and behavior with the system, and therefore the system adjusts itself so that it is “tailored” for the user.
* A better performance is usually apparent in such systems. This means that these systems know how to allocate and use resources in a more linear and optimized way.
* Improved accuracy is also one of the benefits of using the data-driven methodology. Since the system relies on data, the chance of having errors, mistakes, and bugs is lower than the typical software methodologies.

# **Game Description**

In the game, all players start on the "Start of Quarter" square at the bottom right of the board, similar to "GO" in Monopoly. Players take turns rolling the dice and move the number they roll. They can land on squares for products or events. When a player lands on an unowned product square, they can buy it. If they land on a "Product Bidding" square, their products are put up for auction and other players can bid to buy them. If someone else lands on a player's owned square, they have to pay to use the product. Players can build warehouses to increase the price others have to pay to use the product. When a player has warehouses on all squares of a field, they can select a region to expand their business, but it will cost them. When a player passes the "Start of Quarter" square 4 times, the income event happens and if their income is too low, they are eliminated. There are "Stock Exchange" squares where players can take a gamble to either double or lose money. There's also a "Tax" square which charges the player 10% of their value and a "Bank Holiday" square which does nothing. Each product has a unique value that changes when upgraded. The last player left is the winner.

# **Requirements**

This project is about making a virtual board game, similar to the popular game Monopoly. The focus is on the steps of figuring out what the game needs, creating a plan, making the game, and testing it to make sure it works properly. The plan will be shown using a type of picture language called UML. The game will have a unique theme based on the world of computers and information technology, instead of buying properties and charging rent, players will control different areas of technology and allocate resources. The game will be played using a simple text interface through a program like Eclipse, rather than having a fancy graphical interface. The game will communicate with players using English text

A use case is a set of steps that an actor takes to achieve a desired outcome. Each use case is represented by an ellipse in a diagram. The ellipse contains the name of the use case and the corresponding description explains the normal flow of actions and alternative flows under certain circumstances. Several use cases can appear in a single use case diagram and each use case must have a description. The relationships between use cases in the diagram can be <<extend>> or <<include>> and they show how one use case is affected by another as actions take place. The use case description also mentions the preconditions needed for a use case to occur. A game like Monopoly app starts with a set of use cases and unfolds as the actor takes actions to achieve the desired outcomes.

Monopoly app is a board game that involves up to 8 players who take turns rolling virtual dice and landing on squares. The players have to enter their names. They are then told what to do or what their obligations are when they land on a square. If a player's resources change, the system shows the reason and announces their new balance. There is a start square where players pick up resources and other squares where things happen. The game has fields, which are like color groups in Monopoly, and they are made up of different areas. Players must own a whole field before they can develop an area within it. To develop an area, players build houses, and when they have three houses, they can build a hotel. If a player lands on an area they don't own, they have to give up resources. If a player runs out of resources or stops playing, the game ends, and the final state of play is shown. The outcome of the game depends on how it is set up.

# **User Interface**

The player options menu is easy to use and makes the game experience better. The menu box is clearly written so that everyone can understand it. Players choose what they want by entering a number. Some options may not be available if the player hasn't met certain requirements. For example, you can only buy a warehouse if you own all the products of a company. This makes sure players can't cheat. The same goes for requesting a warehouse.

Text

Description automatically generated

## **Player Name**

The player names section allows players to easily enter their names. It's simple and straightforward, making it easy for players to understand. The names are checked to make sure no two players have the same name to avoid confusion. If a name is already taken, the player will have to enter a different name to move on with the game.

Graphical user interface, text

Description automatically generated

## **Purchase a Country**

The "Purchase a country" menu is simple and easy for players to use. They just pick a number for the option they want and the game moves forward. Only products the player owns will be shown at the top of the menu. This is to make sure the player doesn't accidentally break the game by choosing the wrong product. There's also a validation check to make sure the player can't pick the same country twice within the same company. If they already bought a country, like "France", that option will be removed from the list for other products in the company. This is to prevent the player from using the same country again.

Text

Description automatically generated

## **Buying a Product**

In simple words: When buying products, the layout is simple and easy to understand. The system shows the price of the product and lets the player choose if they want to buy it or not. If the player doesn't have enough money, they get an error message. This is because of validation, which helps the game work properly and stops the player from making mistakes that could end the game. To make the game more challenging, the details of the product are hidden.

Text

Description automatically generated

# **Design Documentation**

## **Class Diagram**

The diagram shows the different classes in the system and how they connect with each other. The classes have both things that get information (getters) and things that set information (setters), but they are not shown in the diagram. The main way the player interacts with the game is through the Game class and the Menu class makes it easy to create a menu for the player. The Menu class was taken from a class the team took in level 1. The diagram changed from the first report because the "Company" section is now its own class, which makes it easier to check if the player owns all the products in that company. The diagram also includes new information and changes that have been added during the project. The relationships between the classes have not changed.

Diagram, schematic

Description automatically generated

## **Sequence Diagram**

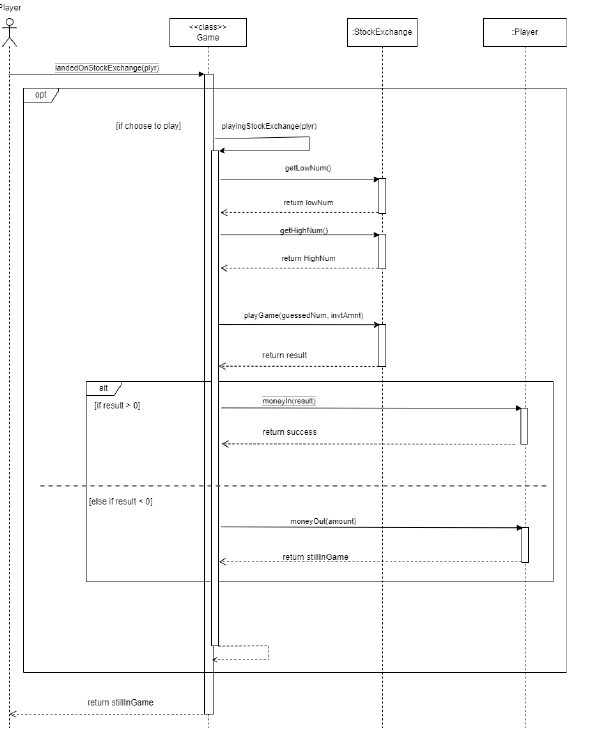
In simple words, this is a diagram that shows what happens when a player lands on a product in the game that is owned by another player. The game checks if the player has enough money to pay for using the product. If the player does, the payment is made and the player who owns the product gets the money. If the player does not have enough money, it returns false.

Diagram

Description automatically generated

## **Stock exchange**

This diagram shows how the game "Choice to Play Game," "Stock Exchange Game," "Money In," "Lost Stock Exchange," and "Money Out" work. It was in the first report, but has been updated. The "Player Eliminated" use case was removed because it got complicated. A new method called "playingStockExchange()" was added to make the code easier to read and maintain. The conditions in the if and else if statements changed and the calls to "moneyIn()" and "moneyOut()" were moved to "playingStockExchange()".



## **Eliminate Player**

This diagram shows how the game "Choice to Play Game," "Stock Exchange Game," "Money In," "Lost Stock Exchange," and "Money Out" work. It was in the first report, but has been updated. The "Player Eliminated" use case was removed because it got complicated. A new method called "playingStockExchange()" was added to make the code easier to read and maintain. The conditions in the if and else if statements changed and the calls to "moneyIn()" and "moneyOut()" were moved to "playingStockExchange()".

Diagram

Description automatically generated

# **Board**

Chart, waterfall chart

Description automatically generated

# **Products Pricing**

Table

Description automatically generated

# **Use Case**

Diagram

Description automatically generated

# **Test Results**

## **Game test**

Graphical user interface, text, application, chat or text message

Description automatically generated

## **Square Test**

Graphical user interface, text, application, chat or text message

Description automatically generated

**Stock Exchange** Graphical user interface, text, application, chat or text message

Description automatically generated

## **Product Bidding Test**

Graphical user interface, text, application, chat or text message

Description automatically generated

## **Start of Quarter Test**

Graphical user interface, text, application, chat or text message

Description automatically generated

## **Product Test**

Graphical user interface, text, application, chat or text message

Description automatically generated

## **Company Test**

Graphical user interface, text, application, chat or text message

Description automatically generated

## **Antitrust Test**

Graphical user interface, text, application, chat or text message

Description automatically generated

## **Players Test**

Graphical user interface, text, application, chat or text message

Description automatically generated

# **RTM (Requirements Traceability Matrix)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Use Case | Description of Test | Test Initialisation | Test Inputs | Test Procedure | Expected Results | Passed? | Result of Test | Reason for failure | JUnit Test | Tester |
| 1 | Money In | Testing if a positive value is added to the user's balance and if the function will return true | The user's balance is 1000 | 100 | Pass '100' into the method 'moneyIn (amount)' | The user's balance should be 1100 and the 'moneyIn' method should return true |  | The amount was successfully added to the user's balance and the method returned 'True' | N/A | moneyInValidAmnt() | Adam Logan [A.L] |
| 2 | Money In | Testing if a negative value is not added to the player's balance and if the function will return false | The user's balance is 1000 | -100 | Pass '-100' into the method 'moneyIn(amount)' | The 'moneyIn' method should return false and the user's balance should remain unchanged |  | The user's balance did not change and the method returned 'False' | N/A | moneyInNegativeInput() | Adam Logan [A.L] |
| 3 | Money In | Testing if nothing will be added to the user's balance if 0 is the value entered into 'moneyIn' | The user's balance is 1000 | 0 | Pass '0' into the method 'moneyIn (amount)' | Nothing should be added to the player's balance |  | The user's balance did not change | N/A | moneyInAddNothing() | Adam Logan [A.L] |
| 4 | Money Out | Testing if a positive value is subtracted from the user's balance and if the function will return true | The user's balance is 1000 | 100 | Pass '100' into the method 'moneyOut (amount)' | The user's balance should be 900 and the 'moneyOut' method should return true |  | The user's balance was 900 and the method returned 'True' | N/A | moneyOutInBalance() | Adam Logan [A.L] |
| 5 | Money Out | Testing if a negative value is not subtracted from the player's balance and if the function will return false | The user's balance is 1000 | -100 | Pass '-100' into the method 'moneyOut (amount)' | The 'moneyOut' method should return true and the user's balance should be 1000 |  | The user's balance was 1100 and not 1000 | The original check in the method simply checked if the balance was greater or equal to the amount and therefore the negative number was subtracted (which added the value to the balance as a '-, - = +') from the balance (successful test ID = 6) | moneyOutNegativeInput() | Adam Logan [A.L] |
| 6 | Money Out | Testing if a negative value is not subtracted from the player's balance and if the function will return false | The user's balance is 1000 | -100 | Pass '-100' into the method 'moneyOut (amount)' | The 'moneyOut' method should return false |  | The user's balance was 1000 | N/A | moneyOutNegativeInput() | Adam Logan [A.L] |
| 7 | Money Out | Testing if nothing will be subtracted to the user's balance if 0 is the value entered into 'moneyOut' | The user's balance is 1000 | 0 | Pass '0' into the method 'moneyOut (amount)' | Nothing should be subtracted from the player's balance |  | The user's balance did not change | N/A | moneyOutRemoveNothing() | Adam Logan [A.L] |
| 8 | Money Out | Testing if the user can lose all the money in their balance and if the function will return true | The user's balance is 1000 | 1000 | Pass '1000' into the method 'moneyOut (amount)' | The user's balance should be 0 and the 'moneyOut' method should return true |  | The user's balance was 0 and the method returned 'True' | N/A | moneyOutRemoveMax() | Adam Logan [A.L] |
| 9 | Money Out | Testing if the 'moneyOut' function will return false if the user loses more money than there is in their account | The user's balance is 1000 | 1001 | Pass '1001' into the method 'moneyOut (amount)' | The user's balance should be -1 and the 'moneyOut' method should return false |  | The user's balance is -1 and the method returned 'False' | N/A | moneyOutRemoveOverBala nce() | Adam Logan [A.L] |
| 10 | Anti-Trust | Testing if the correct persentage is removed from the user's balance when landing on the Anti- Trust square | The user's balance is 1000 and their current square is an Anti-Trust Square | Anti-Trust Square set with a value of 10% | A Player class is created and is passed into the 'fine' method of the AntiTrust class | The user's new balance should be 900 |  | The user's balance is 900 | N/A | fineTest() | Adam Logan [A.L] |
| 11 | Choice to Play Game | Testing if the player can choose not to play the stock exchange game | The player has landed on the stock exchange square | 2 (this is the option to skip the game) | Enter the option 2 | Message displayed informing the player that they have chosen not to play the game |  |  | N/A | N/A | Adam Logan [A.L] |
| 12 | Choice to Play Game | Testing if the player can choose to play the stock exchange game | The player has landed on the stock exchange square | 1 (this is the option to play the game) | Enter the option 1 | The Stock Exchange Game will be played |  |  | N/A | N/A | Adam Logan [A.L] |
| 13 | Stock Exchange Game | Testing if the user can enter a valid amount of money in the stock exchange game | The player has landed on the stock exchange square, the player has choosen to play and they have more than £20 in their balance | 20 | Enter option 1 | The Stock Exchange Game will continue and will prompt the player for a random number |  |  | N/A | N/A | Adam Logan [A.L] |
| then enter 20 |
| 14 | Stock Exchange Game | Testing if the user can enter a valid amount of money in the stock exchange game | The player has landed on the stock exchange square, the player has choosen to play and they have exactly £1000 in their balance | 1000 | Enter option 1 | The Stock Exchange Game will continue and will prompt the player for a random number |  |  | N/A | N/A | Adam Logan [A.L] |
| then enter 1000 |
| 15 | Stock Exchange Game | Testing if the user can enter an invalid amount of money in the stock exchange game and for an error message to be displayed | The player has landed on the stock exchange square, the player has choosen to play and they have exactly £1000 in their balance | 2000 | Enter option 1 | An error message will be displayed and the player will be prompted to enter another amount |  |  | N/A | N/A | Adam Logan [A.L] |
| then enter 2000 |
| 16 | Stock Exchange Game | Testing if the user can enter an invalid amount of money in the stock exchange game and for an error message to be displayed | The player has landed on the stock exchange square, the player has choosen to play and they have exactly £1000 in their balance | 1001 | Enter option 1 | An error message will be displayed and the player will be prompted to enter another amount |  |  | N/A | N/A | Adam Logan [A.L] |
| then enter 1001 |
| 17 | Stock Exchange Game | Testing if the user can enter any character in the stock exchange game and for an error message to be displayed | The player has landed on the stock exchange square, the player has choosen to play | win! | Enter option 1 then enter win! | An error message will be displayed and the player will be prompted to enter an integer value |  |  | N/A | N/A | Adam Logan [A.L] |
| 18 | Stock Exchange Game | Testing if the user can enter any character and some numbers in the stock exchange game and for an error message to be displayed | The player has landed on the stock exchange square, the player has choosen to play | win12! | Enter option 1 then enter win12! | An error message will be displayed and the player will be prompted to enter an integer value |  |  | N/A | N/A | Adam Logan [A.L] |
| 19 | Stock Exchange Game | Testing if the user can enter a valid amount of money in the stock exchange game | The player has landed on the stock exchange square, the player has choosen to play and they have more than £20 in their balance | 20 | Enter option 1 | The Stock Exchange Game will continue and will prompt the player for a random number |  |  | Test failed due to when a string is being converted into an integer it does not recognise the number as an integer even though they have the same value (successful test ID = 20) | N/A | Adam Logan [A.L] |
| then enter 20.0 |
| 20 | Stock Exchange Game | Testing if the user can enter a valid amount of money in the stock exchange game | The player has landed on the stock exchange square, the player has choosen to play and they have more than £20 in their balance | 20.1 | Enter option 1 | The Stock Exchange Game will continue and will prompt the player for a random number |  |  | N/A | N/A | Adam Logan [A.L] |
| then enter 20.1 |
| 21 | Stock Exchange Game | Testing if the user can enter a float into the stock exchange game | The player has landed on the stock exchange square, the player has choosen to play and they have more than £20 in their balance | 20.3 | Enter option 1 | An error message will be displayed and the player will be prompted to enter an integer value |  |  | N/A | N/A | Adam Logan [A.L] |
| then enter 20.3 |
| 22 | Stock Exchange Game | Testing if the user can enter a value within the range of the stock exchange's defined numbers | The player has landed on a stock exchange with the range set to 1 - 10 | 5 | Enter option 1, enter 20 and then enter 5 | The apprpraite message will be displayed (this depends if the value entered is the number generated by the system) |  |  | N/A | N/A | Adam Logan [A.L] |
| 23 | Stock Exchange Game | Testing if the user can enter a value within the range of the stock exchange's defined numbers | The player has landed on a stock exchange with the range set to 1 - 10 | 10 | Enter option 1, enter 20 and then enter 10 | The apprpraite message will be displayed (this depends if the value entered is the number generated by the system) |  |  | N/A | N/A | Adam Logan [A.L] |
| 24 | Stock Exchange Game | Testing if the user can enter a value that is not within the pre- defined range | The player has landed on a stock exchange with the range set to 1 - 10 | 11 | Enter option 1, enter 20 and then enter 11 | An error message will be displayed and the player will be prompted to enter another value |  |  | N/A | N/A | Adam Logan [A.L] |
| 25 | Stock Exchange Game | Testing if the user can enter characters when asked to choose a number and for an error message to be displayed | The player has landed on a stock exchange with the range set to 1 - 10 | win! | Enter option 1, enter 20 and then enter win! | An error message will be displayed and the player will be prompted to enter an integer value |  |  | N/A | N/A | Adam Logan [A.L] |
| 26 | Stock Exchange Game | Testing if the user can enter any characters and some numbers in the when asked to choose a number and for an error message to be displayed | The player has landed on a stock exchange with the range set to 1 - 10 | win12! | Enter option 1, enter 20 and then enter win12! | An error message will be displayed and the player will be prompted to enter an integer value |  |  | N/A | N/A | Adam Logan [A.L] |
| 27 | Stock Exchange Game | Testing if the user can enter a float into the stock exchange game when asked to guess a random number | The player has landed on a stock exchange with the range set to 1 - 10 | 5 | Enter option 1, enter 20 and then enter 5.0 | The apprpraite message will be displayed (this depends if the value entered is the number generated by the system) |  |  | Test failed due to when a string is being converted into an integer it does not recognise the number as an integer even though they have the same value (successful test ID = 28) | N/A | Adam Logan [A.L] |
| 28 | Stock Exchange Game | Testing if the user can enter a float into the stock exchange game when asked to guess a random number | The player has landed on a stock exchange with the range set to 1 - 11 | 5.1 | Enter option 1, enter 20 and then enter 5.1 | The apprpraite message will be displayed (this depends if the value entered is the number generated by the system) |  |  | N/A | N/A | Adam Logan [A.L] |
| 29 | Stock Exchange Game | Testing if the user can enter a float into the stock exchange game when asked to guess a random number | The player has landed on a stock exchange with the range set to 1 - 10 | 5.3 | Enter option 1, enter 20 and then enter 5.3 | An error message will be displayed and the player will be prompted to enter an integer value |  |  | N/A | N/A | Adam Logan [A.L] |
| 30 | Stock Exchange Game | Testing if the correct number is returned when the user correctly guesses the number | The stock exchange object has been set with the range 1 - 10 and a risk of | 5 | pass 5 as they guessedNum, 100 as the invtAmnt and 5 as the correctGuess into the 'playGame()' method | The returned value should be 125 |  | Returned 125 | N/A | winningGmaeTest | Adam Logan [A.L] |
| 0.25. The correct guess is 5 and the investement is 100. |
| 31 | Lost Stock Exchange | Testing if the correct amount is returned when the user incorreclty guesses the number | The stock exchange object has been set with the range 1 - 10 and a risk of | 1 | pass 1 as the guessedNum, 100 as the invtAmnt and 5 as the correctGuess into the 'playGame()' method | The returned value should be |  | Returned -40 | N/A | loseBy4() | Adam Logan [A.L] |
| 0.25. The correct guess is 5 and the investement is 100. | -40 |
| 32 | Lost Stock Exchange | Testing if the correct amount is returned when the user incorreclty guesses the number | The stock exchange object has been set with the range 1 - 10 and a risk of | 2 | pass 2 as the guessedNum, 100 as the invtAmnt and 5 as the correctGuess into the 'playGame()' method | The returned value should be |  | Returned -30 | N/A | loseBy3() | Adam Logan [A.L] |
| 0.25. The correct guess is 5 and the investement is 100. | -30 |
| 33 | Lost Stock Exchange | Testing if the correct amount is returned when the user incorreclty guesses the number | The stock exchange object has been set with the range 1 - 10 and a risk of | 3 | pass 3 as the guessedNum, 100 as the invtAmnt and 5 as the correctGuess into the 'playGame()' method | The returned value should be |  | Returned -20 | N/A | loseBy2() | Adam Logan [A.L] |
| 0.25. The correct guess is 5 and the investement is 100. | -20 |
| 34 | Lost Stock Exchange | Testing if the correct amount is returned when the user incorreclty guesses the number | The stock exchange object has been set with the range 1 - 10 and a risk of | 4 | pass 4 as the guessedNum, 100 as the invtAmnt and 5 as the correctGuess into the 'playGame()' method | The returned value should be |  | Returned -10 | N/A | loseBy1() | Adam Logan [A.L] |
| 0.25. The correct guess is 5 and the investement is 100. | -10 |
| 35 | Lost Stock Exchange | Testing if the correct amount is returned when the user incorreclty guesses the number | The stock exchange object has been set with the range 1 - 10 and a risk of | 6 | pass 6 as the guessedNum, 100 as the invtAmnt and 1 as the correctGuess into the 'playGame()' method | The returned value should be |  | Returned -50 | N/A | loseBy5() | Adam Logan [A.L] |
| 0.25. The correct guess is 5 and the investement is 100. | -50 |
| 36 | Lost Stock Exchange | Testing if the correct amount is returned when the user incorreclty guesses the number | The stock exchange object has been set with the range 1 - 10 and a risk of | 7 | pass 7 as the guessedNum, 100 as the invtAmnt and 1 as the correctGuess into the 'playGame()' method | The returned value should be |  | Returned -60 | N/A | loseBy6() | Adam Logan [A.L] |
| 0.25. The correct guess is 5 and the investement is 100. | -60 |
| 37 | Lost Stock Exchange | Testing if the correct amount is returned when the user incorreclty guesses the number | The stock exchange object has been set with the range 1 - 10 and a risk of | 8 | pass 8 as the guessedNum, 100 as the invtAmnt and 1 as the correctGuess into the 'playGame()' method | The returned value should be |  | Returned -70 | N/A | loseBy7() | Adam Logan [A.L] |
| 0.25. The correct guess is 5 and the investement is 100. | -70 |
| 38 | Lost Stock Exchange | Testing if the correct amount is returned when the user incorreclty guesses the number | The stock exchange object has been set with the range 1 - 10 and a risk of | 9 | pass 9 as the guessedNum, 100 as the invtAmnt and 1 as the correctGuess into the 'playGame()' method | The returned value should be |  | Returned -80 | N/A | loseBy8() | Adam Logan [A.L] |
| 0.25. The correct guess is 5 and the investement is 100. | -80 |
| 39 | Lost Stock Exchange | Testing if the correct amount is returned when the user incorreclty guesses the number | The stock exchange object has been set with the range 1 - 10 and a risk of | 10 | pass 10 as the guessedNum, 100 as the invtAmnt and 1 as the correctGuess into the 'playGame()' method | The returned value should be |  | Returned -90 | N/A | loseBy9() | Adam Logan [A.L] |
| 0.25. The correct guess is 5 and the investement is 100. | -90 |
| 40 | Auction starts | Testing if the system can choose a field to put up for auction | The player has landed on the Company Bidding field, and the system will select a field from the resulting player | N/A | System will randomly select a field from the resulting player to auction to the other players | The system will pick a field at random and the auction will begin |  | One of the products was chosen | N/A | getValidPrdctTest() | XXXXX [X.X] |
| 41 | No Products Owned | Testing if the System will display an appropriate error message when the player who landed on the product bidding square has no products | Player lands on the product bidding square, but player does not own any products to bid | N/A | System tries to randomly select a field from resulting player | The Sytem will display an error message, stating the resulting player does not own any products. game will procced as normal |  |  | N/A | N/A | XXXXX [X.X] |
| 42 | Auction starts | Testing if the current player can place their bid | Player 2 will enter their bid for Player 1s feild | 500 | When prompted, enter 500 | Player 2 will be able to successfully enter their bid for player 1s field |  |  | N/A | N/A | XXXXX [X.X] |
| 43 | Auction starts | Testing what happens when the current player, enters a lower bidding amount than the previous player | The current player will enter thier bid for the field. | 250 | When prompted, enter 250 | The System will display an error message stating the current players bis is lower than the previous players amount, and ask them to re-enter their bid |  |  | N/A | N/A | XXXXX [X.X] |
| 44 | Auction starts | Testing to see if the system will correctly select the player with the highest bid | The system will look at all the bids and select the highest | 500 | System will select 500 (as it is the largest) | The system will select the correct player and this player will recieve Player 1s field |  |  | N/A | N/A | XXXXX [X.X] |
| 45 | Create Player | Testing if the user is able to enter valid charactors | The system will ask "what is your name", the Player will enter a name | "Jake" | Player will enter "Jake" when prompted by System | The system will allow this as it contains valid charactors. the Game will move onto the next players name |  |  | N/A | N/A | XXXXX [X.X] |
| 46 | Create Player | Testing if the user can enter a non-valid charactor | System will ask player to enter their name | "Jak£" | Player will enter "Jak£" when prompted by System | The system will display an error message back to the player and say that they are not allowed to enter non-letter charactors |  |  | There is no check for special characters or numbers (successful test ID = 47) | N/A | XXXXX [X.X] |
| 47 | Create Player | Testing if the user can enter a non-valid charactor | System will ask player to enter their name | "Jak£" | Player will enter "Jak£" when prompted by System | The system will display an error message back to the player and say that they are not allowed to enter non-letter charactors |  |  | N/A | N/A | XXXXX [X.X] |
| 48 | Create Player | Testing if the game will proced once everyone has entered thier names | The system will ask each player to enter their names, in turns | "Jake", "Adam", "Joe", "ShaunChathal" | Players will enter thier own names, in turns. system will finish once everyone has entered their names. | Once the players have entered therir names, the system willl proced onto the game as normal, starting with the first players' turn |  |  | N/A | N/A | XXXXX [X.X] |
| 49 | Create Player | Testing to see if an Error message will appear when 2 players enter the same name as each other | The system will ask each player to enter their names, in turns | "Adam" "Adam" | Players will enter thier own names, in turns. system will finish once everyone has entered their names. | when a player enters the same name as another player, the system will display an error message to the user stating they cant have the same name as another player |  |  | N/A | N/A | XXXXX [X.X] |
| 50 | Income Event | Testing to the player is kept in the game, if they have more money than the threshold | when the player passes go, the threshold is £500, and the player has £1000 in their balance | N/A | player lands / passes go with | When the player passes go for the 4th time in a row, the player will be kept in the game as they have more funds in their balance, than the threshold |  |  | N/A | N/A | XXXXX [X.X] |
| £1000 in their balance, and is above the threshold |
| 51 | Income event | Testing to the player is kept in the game, if they have more money than the threshold | when the player passes go, the threshold is £500, and the player has £450 in their balance | N/A | player lands / passes go with | When the player passes go for the 4th time in a row, the player will be kept eliminated from the game |  |  | N/A | N/A | XXXXX [X.X] |
| £450 in their balance, and is below the threshold |
| 52 | Request to buy country | Testing to see if the player is able to purchase an country for a product they own | The player will select the 'request to buy country' option and will own all the products the company 'AMD' and 4 warehouses are dedicated to each of the products. | Selects to buy | The player will relocate the product SeaMicro to the United States. | The option to request a country is shown and the amount to charge for the product SeMicro has increased by 150. The player's balance will decrease by 45. |  | The amount to charge of SeaMicro was increased by 150 and the player was charged 45. | N/A | N/A | XXXXX [X.X] |
| 53 | Request to buy country | Testing to see if the option to purchase a country is not shown, if they dont own all the products within the company | The player will select the 'request to buy country' option and will own 'ryzen' with all 4 warehouses, within the company 'AMD' | Selects to buy | the player wont get the option to relocate their product as they dont own all products within the company | The option to request a country will not be shown as the player does not own all the products within 'AMD' |  |  | N/A | N/A | XXXXX [X.X] |
| 54 | Request to buy country | Testing to see if the player can see the option to request to purchase a country, if they dont have all 4 warehouses on the products | The player will select the 'request to buy country' option and will own all then products within the company 'AMD', but doesnt own all 4 warehouses on all the products | Selects to buy | The player wont be given the option to purchase a country when they dont own all the warehouses on the products | The option to request a country will not be shown as the player does not own all the warehouses, on their products, within 'AMD' |  |  | N/A | N/A | XXXXX [X.X] |
| 55 | Request to buy country | Testing to see if, when the player tries to purchase another country, the same option will not appear twice | The player will select 'request to purchase a country' and wont be given the option 'United States', as the player already has a product with this county | Selects to buy | the player will select the 'request to buy office' option. The system will check to see if the player already owns any other countries on other products. | Since the player already owns 'United States' on another product, the system wont display this option again to the user, making them choose a different option. the relocation fee has also increased by £25 |  |  | N/A | N/A | XXXXX [X.X] |
| 56 | Charged Product | Testing to see if player is charged money when they land on owned product | Both users balance start with 1000 | 1 (Roll dice) | Enter 1 (Land on an owned product) | The player landing on the owned product will have their balance reduced, the player owning product will receive their balance |  |  | N/A | N/A | XXXXX [X.X] |
| 57 | Charged Product | Testing if player will be charged more if product has warehouse | Player with product must also have a warehouse | 1 (Roll dice) | Enter 1 (Land on an owned product) | Since player 1 has added a warehouse to their product, it should charge the next player who lands on it |  |  | N/A | N/A | XXXXX [X.X] |
| 58 | Charged Product | Testing if product with a country will charge more | Player with product must own a country | 1 (Roll dice) | Enter 1 (Land on an owned product) | When player that owns a product upgrades and gets a country, it should charge the other players landing on it |  |  | N/A | N/A | XXXXX [X.X] |
| 59 | Request Warehouse | Testing if warehouse will be built on owned product | Player must own a product | 7 (Purchase Warehouse) | Choose option 'Purchase Warehouse' | When choosing to purchase a in the menu it should subtract the money from users balance and build warehouse |  | 17 was subtracted from the player's balance and the new amount to charge a player for landing on the product has been updated | N/A | N/A | XXXXX [X.X] |
| 60 | Request Warehouse | When choosing a product to purchase a warehouse for enter string instead of int | Player must own a product | f | Choose option 'Purchase Warehouse' then enter "f" for answer | The system should reject the input and return an error message and allow them to re enter a number. |  |  | N/A | N/A | XXXXX [X.X] |
| 61 | Request Warehouse | Continue entering incorrect values when purchasing warehouse | Must be purchasing a warehouse | 5 | Choose the option to purchase the warehouse enter integer out of range, letter and larger integers. | The system should display an error message |  |  | N/A | N/A | XXXXX [X.X] |
| 62 | Request Warehouse | Continue entering incorrect values when purchasing warehouse | Must be purchasing a warehouse | 55 | Choose the option to purchase the warehouse enter integer out of range, letter and larger integers. | The system should display an error message |  |  | N/A | N/A | XXXXX [X.X] |
| 63 | Request Warehouse | Continue entering incorrect values when purchasing warehouse | Must be purchasing a warehouse | 1000000 | Choose the option to purchase the warehouse enter integer out of range, letter and larger integers. | The system should display an error message |  |  | N/A | N/A | XXXXX [X.X] |
| 64 | Request Warehouse | Continue entering incorrect values when purchasing warehouse | Must be purchasing a warehouse | 1E+23 | Choose the option to purchase the warehouse enter integer out of range, letter and larger integers. | The system should display an error message |  |  | N/A | N/A | XXXXX [X.X] |
| 0 |
| 65 | Request Warehouse | When buying a new product will it be added to the list of available products to build warehouses for | Purchase an available product | Enter 7 on menu | Enter 1 (Land on an available product), purchase product. Purchase warehouse on menu. | The system should add the purchased product to the list of options for a warehouse |  |  | N/A | N/A | XXXXX [X.X] |
| 66 | Request Warehouse | Try to build a warehouse when not owning any products | N/A | Choose to buy warehouse on menu | Choose the option to purchase the warehouse | The system should hide the ability to do anything if they player cannot do it, so the option to build warehouse should be hidden |  |  | N/A | N/A | XXXXX [X.X] |
| 67 | Request Warehouse | Building multiple warehouses an the one product | N/A | Choose to buy warehouse on menu | Choose the option to purchase the warehouse | The system should increase the price when landing on product |  | The resource use when landing on the product will increase, the money will also be took from the players balance | N/A | N/A | XXXXX [X.X] |
| 68 | Request Warehouse | Test system to see if it stops player from purchasing more than 4 warehouses for 1 product | Purchase 4 warehouses for a single product | 7 on menu multiple times | Purchase 4 warehouses for one product, after 4 request to buy warehouse | Any product that reaches the limit of 4 warehouses, will be removed from the list of available products |  |  | N/A | N/A | XXXXX [X.X] |
| 69 | Player's Turn | Testing if the turn switches between four players. | It is the first player's turn. | Enter | Press 'Enter' to proceed through the game to the next player's turn. | The turn will change between the four players. |  | The turn switches between all four players. | N/A | N/A | XXXXX [X.X] |
| 70 | Player's Turn | Testing if the player can choose other game options, i.e. purchasing a product, before ending their turn. | It is the first player's turn. | 5 | Pass '5' into the menu to check the player's inventory. | The player will be able to choose view their inventory before ending their turn. |  | The player is able to choose to view their inventory before ending their turn. | N/A | N/A | XXXXX [X.X] |
| 71 | Player's Turn | Testing if the system lets eliminated players have a turn. | A player must be eliminated. | Enter | Press 'Enter' to proceed through the game to the next player's turn until it gets back to the turn of the player it started on. | The game will cycle through the turns, skipping eliminated players. |  | The game cycles through the turns, skipping eliminated players. | N/A | N/A | XXXXX [X.X] |
| 72 | Player's Turn | Testing if the turn changes in a consistent order. | It is the first player's turn. | Enter | Press 'Enter' to proceed through the game to the next person's turn 12 times. | The turn will change in a consistent order between all players. |  | The turn switches in a consistent order. | N/A | N/A | XXXXX [X.X] |
| 73 | Passing Go | Testing if the player receives money when they land on/pass the starting position. | The player is one move away from landing on or passing the starting position. | 0 | Pass '0' into the menu to roll the dice. | The player will receive money when they land on or pass the starting position. |  | The player receives money when they land on or pass the starting position. | N/A | N/A | XXXXX [X.X] |
| 74 | Passing Go | Testing if the player receives the correct amount of money when they land on/pass the starting position. | The player's balance is 1000. | 0 | Pass '0' into the menu to roll the dice. | The player will have recieved |  | The player receives £150. | N/A | N/A | XXXXX [X.X] |
| £150 when passing the start of quarter. |
| 75 | Buying Product | Testing if the system offers the player the option to buy a product. | The product must not be owned by any players. | N/A | Roll the dice until the player lands on a product which is known not to be owned by another player | The player will be prompted with a message asking if they want to purchase the product |  | The player is presented with the option to buy the product. | N/A | N/A | XXXXX [X.X] |
| 76 | Buying Product | Testing if the player can purchase a product with sufficient funds. | The player must have sufficient funds and the product must not be owned by a player. | N/A | Roll the dice until the player lands on a product which is known not to be owned by another player | The player will be able to purchase the product. |  | The player is able to purchase the product. | N/A | N/A | XXXXX [X.X] |
| 77 | Buying Product | Testing if the player can purchase a product without having sufficient funds. | The player must not have sufficient funds to purchase the product. | N/A | Roll the dice until the player lands on a product which is known not to be owned by another player | The player won't be able to purchase the product. |  | The player is not able to purchase the product. | N/A | N/A | XXXXX [X.X] |
| 78 | Buying Product | Testing if a player can purchase a procuct which is owned by someone else. | The product must be owned by someone else. | N/A | Roll the dice until the player lands on a product which is known to be owned by another player | The player won't be able to purchase the product. |  | The player is not able to purchase the product. | N/A | N/A | XXXXX [X.X] |
| 79 | Buying Product | Testing if a player can purhcase a product that they already own. | The player must already own the product. | N/A | Roll the dice until the player lands on a product which is known to be owned by the current player | The player won't be able to purchase the product again. |  | The player is not able to purchase the product again. | N/A | N/A | XXXXX [X.X] |
| 80 | Buying Product | Testing if the system removes the correct amount from the player's balance for the purchase. | The player must land on Galaxy Tab with a balance of 1000. | N/A | Roll the dice until the player lands on a product which is known to be owned by the current player | 220 will be deducted from the player's balance leaving them with a balance of 780. |  | The correct amount is deducted from the player's balance. | N/A | N/A | XXXXX [X.X] |
| 81 | Buying Product | Testing if the system changes the players turn once they select refuse to buy the product. | The player must be given the option to buy a product. | N/A | Pass 'N' into the landedOnProduct method. | The turn will move onto the next player. |  | The turn moves onto the next player. | N/A | N/A | XXXXX [X.X] |
| 82 | Buying Product | Testing if when a player buys a product, it adds the product to their inventory. | The first player must be given the option to buy a product. | N/A | Check the player's balance | The product will be added to the player's inventory and will be listed when the player views their inventory. |  | The product is added to the player's inventory. | N/A | N/A | XXXXX [X.X] |
| 83 | Eliminate Player | Testing if the correct player is eliminated. | One player must be called "a". | a | Pass "a" into the eliminatePlayers method. | The specified player will be eliminated. |  | The correct player is eliminated. | N/A | N/A | XXXXX [X.X] |
| 84 | Eliminate Player | Testing if the player is eliminated once their balance reaches 0. | Player's balance must be at 5 when they land on FitBit. | Y | Pass 'Y' into the landedOnProduct method. | The player will be eliminated after they purchase the product. |  | The player is eliminated after they purchase the product. | N/A | N/A | XXXXX [X.X] |
| 85 | Eliminate Player | Testing if the products owned by the player are returned to the system once the player is eliminated. | Player must own at least one product when they are eliminated | 2 | Pass '2' into the menu to eliminate the player. | The products owned by the player will be returned to the system. |  | The products owned by the player are returned to the system. | N/A | N/A | XXXXX [X.X] |
| 86 | Winning | Testing if the win message displays correctly when a player wins. | There must be two players left. | 2 | Pass '2' into the menu to eliminate a player. | The win message will display correctly. |  | The win message displays correctly. | N/A | N/A | XXXXX [X.X] |
| 87 | Winning | Testing if a player wins once they are the last player left. | There must be two players left. | 2 | Pass '2' into the menu to eliminate a player. | The player will win when they are the last one left. |  | The player wins when they're the last one left alive. | N/A | N/A | XXXXX [X.X] |
| 88 | Winning | Testing if the game ends once a player wins. | A player must have won. | Enter | When the win message is displayed, input 'Enter'. | The game will end and the application will close. |  | The game ends when a player wins. | N/A | N/A | XXXXX [X.X] |
| 89 | Save Game | Testing if the correct details of the players are saved into a csv file correclty | The game is started with 3 players Adam, Joe and Jake | Adam (balance = 980, present square = Anti Trust), Joe (balance = 1000, present square = Sotheby's Auction) and Jake (balance = 711, present square = Kindle). | To save and close the game on Jake's turn | A csv file will be created with the correct inputs |  |  | N/A | N/A | Adam Logan [A.L] |
| None of the players have passed the start. Adam ows ipad, ipod and iphone while Jake owns Galaxy Phone. |
| 90 | Save Game | Testing if the correct deails of the products are saved into a csv file correclty | The products Galaxy Phone, iPad, iPod and iPhone should be owned | Galxy Phone (amount to charge = 10, cost of relocation | To save and close the game on Jake's turn | A csv file will be created with the correct inputs |  |  | N/A | N/A | Adam Logan [A.L] |
| = 145), iPad (amount to charge = 9, cost of relocation = 25), iPod (amount to charge = 7, cost of relocation = 20), iPhone (amount to charge = 57, cost of relocation = 45). |
| The products iPad, iPodm and iPhone will have 4 warehouses bought and Galaxy Phone will not have any bought. The only product to have a countyr owned is iPhone and the country should be the United States. |
| 91 | Restore Game | Testing if the correct details of the players are loaded from the csv file correclty, into the game | For a Game to be saved with the details described in the 'Test Inputs' column | Adam (balance = 980, present square = Anti Trust), Joe (balance = 1000, present square = Sotheby's Auction) and Jake (balance = 711, present square = Kindle). | To load the saved game | For the correct details to be loaded into the game, and for Jake to be the first player to have their turn |  | All player's present square was the one which was saved in the csv file and Jake was the first player to have their turn | N/A | N/A | Adam Logan [A.L] |
| None of the players have passed the start. Adam ows ipad, ipod and iphone while Jake owns Galaxy Phone. |
| 92 | Restore Game | Testing if the correct details of the products are loaded from the csv file correclty, into the game | For a Game to be saved with the details described in the 'Test Inputs' column | Galxy Phone (amount to charge = 10, cost of relocation | To load the saved game | For the correct details to be loaded into the game |  |  | N/A | N/A | Adam Logan [A.L] |
| = 145), iPad (amount to charge = 9, cost of relocation = 25), iPod (amount to charge = 7, cost of relocation = 20), iPhone (amount to charge = 57, cost of relocation = 45). |
| The products iPad, iPodm and iPhone will have 4 warehouses bought and Galaxy Phone will not have any bought. The only product to have a countyr owned is iPhone and the country should be the United States. |
| 93 | Close Game | Testing if all players that have the same balance are returned | For an array of three players to be created | All players in the array to have 1000 in their balance | Pass an array of the players into the 'highestBalance()' method | For the returned arraylist to contain all three players |  | Returned an arraylist of all three players | N/A | allPlyrsSameBalance() | Adam Logan [A.L] |
| 94 | Close Game | Testing if the two players that have the highest balance are returned | For an array of three players to be created | For two players in the array to have 1001 in their balance | Pass an array of the players into the 'highestBalance()' method | For the returned arraylist to contain the two players with the highest balance |  | Returned an arraylist of all two players | N/A | twoPlyrsWithHighBalance() | Adam Logan [A.L] |
| 95 | Close Game | Testing if the player with the highest balance is returned | For an array of three players to be created | For one player in the array to have 1001 in their balance | Pass an array of the players into the 'highestBalance()' method | For the returned arraylist to contain the one player with the highest balance |  | Returned an arraylist with just the one player with the highest balance | N/A | onePlyrWithHighBalance() | Adam Logan [A.L] |

# **References**

1. Thampy, R. (2019). *7 Stages of SDLC | 7 Phases of SDLC Software Development Life Cycle | Betsol*. [online] www.betsol.com. Available at: <https://www.betsol.com/blog/7-stages-of-sdlc-how-to-keep-development-teams-running/>.
2. Singla, C. (2020). *Functional vs Non Functional Requirements*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/.
3. Halwai, img src="https://www openxcell com/wp-content/uploads/2021/06/Siften\_Image jpg" class="blog-use-image" alt="author"/> S. (2021). *Design Phase in SDLC - A Complete Guide [2021]*. [online] OpenXcell. Available at: https://www.openxcell.com/blog/design-phase-in-sdlc/#:~:text=What%20is%20the%20Design%20Phase.
4. www.synopsys.com. (n.d.). *What Is the Software Development Life Cycle (SDLC) and How Does It Work? | Synopsys*. [online] Available at: https://www.synopsys.com/glossary/what-is-sdlc.html#:~:text=Definition.
5. JavaTpoint (2011). *Agile Model - javatpoint*. [online] www.javatpoint.com. Available at: https://www.javatpoint.com/software-engineering-agile-model.
6. Gurendo, D. (2015). *Scrum Methodology Phases which Help in Agile SDLC Process: 5 Key Steps*. [online] XB Software. Available at: https://xbsoftware.com/blog/software-development-life-cycle-sdlc-scrum-step-step/.
7. admin (2020). *What is Scrum?* [online] Agile Arena. Available at: https://www.agilearena.net/what-is-scrum/.
8. www.h2kinfosys.com. (2018). *SOFTWARE DEVELOPMENT LIFE CYCLE APPROACHES | SDLC MODELS*. [online] Available at: https://www.h2kinfosys.com/blog/sdlc-models/#:~:text=The%20popular%20SDLC%20models%20that.
9. www.h2kinfosys.com. (2021). *SEQUENTIAL SDLC MODEL | H2kinfosys Blog*. [online] Available at: https://www.h2kinfosys.com/blog/sequential-sdlc-model/.
10. Kumar Pal, S. (2019). *Software Engineering | Classical Waterfall Model - GeeksforGeeks*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/.
11. Martin, M. (2019). *What is Waterfall Model in SDLC? Advantages & Disadvantages*. [online] Guru99.com. Available at: https://www.guru99.com/what-is-sdlc-or-waterfall-model.html.
12. Inc, B. (n.d.). *What is Waterfall Model and Why It’s so Great - An Overview*. [online] blog.bydrec.com. Available at: https://blog.bydrec.com/the-time-tested-waterfall-model-and-why-its-so-great#:~:text=The%20model%20is%20simple%20and [Accessed 1 Feb. 2023].
13. JavaTpoint (2011). *Advantage of Agile Methodology | Disadvantage of Agile Methodology - Javatpoint*. [online] www.javatpoint.com. Available at: https://www.javatpoint.com/advantage-and-disadvantage-of-agile-methodology.
14. GeeksForGeeks (2020). *Agile Methodology Advantages and Disadvantages*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/agile-methodology-advantages-and-disadvantages/.
15. Chandana (2022). *Scrum Project Management – Pros and Cons*. [online] Simplilearn.com. Available at: https://www.simplilearn.com/scrum-project-management-article.
16. Team, E. (n.d.). *Advantages and Disadvantages of Scrum Methodology*. [online] ProjectPractical.com. Available at: https://www.projectpractical.com/advantages-and-disadvantages-of-scrum-methodology/.
17. WADIC (2019). *6 Advantages and Disadvantages of the Waterfall Model | Wadic*. [online] WADIC. Available at: https://wadic.net/waterfall-model-advantages-disadvantages/.
18. A Plus Topper. (2021). *Waterfall Model Advantages and Disadvantages | What is Waterfall Model? Advantages and Disadvantages of Waterfall Model*. [online] Available at: https://www.aplustopper.com/waterfall-model-advantages-and-disadvantages/.
19. JavaTpoint (n.d.). *V-model (Software Engineering) - javatpoint*. [online] www.javatpoint.com. Available at: https://www.javatpoint.com/software-engineering-v-model.
20. admin (2021). *What is V model - Definition, Advantages and Disadvantages*. [online] Sample Test Cases. Available at: https://sampletestcases.com/v-model/.
21. prasanna (2021). *V Model Advantages and Disadvantages | What is V-Model? When and How To Use It?Merits and Demerits*. [online] A Plus Topper. Available at: https://www.aplustopper.com/v-model-advantages-and-disadvantages/.
22. www.tutorialspoint.com. (n.d.). *SDLC - V-Model*. [online] Available at: https://www.tutorialspoint.com/sdlc/sdlc\_v\_model.htm#:~:text=V%2DModel%20%2D%20Design.
23. CAST (2017). *Accelerate Application Modernization and Cloud Migration with Software Intelligence*. [online] Castsoftware.com. Available at: https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects.
24. Award-winning Technology Consultancy specialising in Software Development, UI/UX, and Front-End Development. Independence is the Difference. New York City. (n.d.). *Understanding Risk Management in Software Development: 7 Common Risks*. [online] Available at: https://www.door3.com/blog/understanding-risk-management-in-software-development-7-common-risks.
25. ENISA. (n.d.). *Risk Treatment*. [online] Available at: https://www.enisa.europa.eu/topics/risk-management/current-risk/risk-management-inventory/rm-process/risk-treatment.
26. Ajackus. (n.d.). *Is the Waterfall model still applicable today?* [online] Available at: https://www.ajackus.com/blog/is-the-waterfall-model-still-relevant-in-todays-time [Accessed 1 Feb. 2023].
27. Anon, (n.d.). *Waterfall Model: A Better Way For Software Development*. [online] Available at: https://www.tekkiwebsolutions.com/blog/waterfall-model-for-software-development/.
28. monday.com Blog. (n.d.). *Complete guide to Waterfall methodology*. [online] Available at: https://monday.com/blog/project-management/waterfall-methodology/.
29. Jena, S. (2020). *Types of Feasibility Study in Software Project Development*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/types-of-feasibility-study-in-software-project-development/.
30. Lucidchart (2019). *What Is a Flowchart*. [online] Lucidchart.com. Available at: https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial.
31. Workflow Management Software by Integrify. (2022). *What is a Flowchart, and Why Is It Important?* [online] Available at: https://www.integrify.com/what-is-a-flowchart/#:~:text=Flowcharts%20make%20it%20easier%20for.
32. GeeksforGeeks. (2017). *An introduction to Flowcharts - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/an-introduction-to-flowcharts/.
33. A Plus Topper. (2022). *Advantages and Disadvantages of Flowchart | What are the Advantages and Disadvantages of Flowchart?* [online] Available at: https://www.aplustopper.com/advantages-and-disadvantages-of-flowchart/.
34. Lucidchart (2019). *What Is a Flowchart*. [online] Lucidchart.com. Available at: https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial.
35. Programiz (2019). *Design Flowchart In Programming (With Examples) - Programiz*. [online] Programiz.com. Available at: https://www.programiz.com/article/flowchart-programming.
36. LucidChart (2017). *ER diagram tutorial*. [online] Lucidchart.com. Available at: https://www.lucidchart.com/pages/er-diagrams.
37. www.javatpoint.com. (n.d.). *Software Engineering Entity-Relationship Diagram - javatpoint*. [online] Available at: https://www.javatpoint.com/software-engineering-entity-relationship-diagrams#:~:text=Entity%2DRelationship%20Diagrams [Accessed 1 Feb. 2023].
38. Sengupta, S. (2020). *Explaining an ER Diagram, With Steps and Use Cases*. [online] Vertabelo Data Modeler. Available at: https://vertabelo.com/blog/how-to-use-er-diagram/.
39. Thampy, R. (2022) *7 stages of SDLC: 7 phases of SDLC Software Development Life cycle*, *BETSOL*. Available at: https://www.betsol.com/blog/7-stages-of-sdlc-how-to-keep-development-teams-running/ (Accessed: February 1, 2023). *What is a relationship in database? (definition and types)* (no date). Available at: https://in.indeed.com/career-advice/career-development/what-is-relationship-in-database (Accessed: February 1, 2023).
40. creately.com. (2022). *Ultimate Entity Relationship Diagram Tutorial (ER Diagrams) | Creately*. [online] Available at: https://creately.com/guides/er-diagrams-tutorial/.
41. Ben (2019). *A Guide to the Entity Relationship Diagram (ERD)*. [online] Database Star. Available at: https://www.databasestar.com/entity-relationship-diagram/.
42. Blogspot.com. (2017). *Advantages and Disadvantages of ER Model in DBMS*. [online] Available at: https://pctechnicalpro.blogspot.com/2017/04/advantages-disadvantages-er-model-dbms.html.
43. Assignology. (n.d.). *Benefits and Disadvantages of ER Diagrams | Healthcare Essay Example*. [online] Available at: https://assignology.com/benefits-and-disadvantages-of-er-diagrams/.
44. Nimrakhan89 (2020). *Advantages and disadvantages of er model in DBMS. Types of database m…*. [online] Available at: https://www.slideshare.net/Nimrakhan89/advantages-and-disadvantages-of-er-model-in-dbms-types-of-database-models.
45. creately.com. (2022). *Ultimate Entity Relationship Diagram Tutorial (ER Diagrams) | Creately*. [online] Available at: https://creately.com/guides/er-diagrams-tutorial/.
46. Mátyás Lancelot Bors (2018). *What is a Finite State Machine?* [online] Medium. Available at: https://medium.com/@mlbors/what-is-a-finite-state-machine-6d8dec727e2c.
47. Wikipedia. (2021). *Extended finite-state machine*. [online] Available at: https://en.wikipedia.org/wiki/Extended\_finite-state\_machine.
48. www.quora.com. (n.d.). *What are extended finite state machines? What is the difference between EFSM and FSM? - Quora*. [online] Available at: https://www.quora.com/What-are-extended-finite-state-machines-What-is-the-difference-between-EFSM-and-FSM.
49. Moore, K. and Gupta, D. (2011). *Finite State Machines | Brilliant Math & Science Wiki*. [online] Brilliant.org. Available at: https://brilliant.org/wiki/finite-state-machines/.
50. WayPath. (2021). *What Is Data-Driven Development | It’s Benefits*. [online] Available at: https://waypathconsulting.com/data-driven-development-and-its-benefits/.
51. CloudZero (n.d.). *Data-Driven Software Development: How To Make Strategic Dev Decisions*. [online] www.cloudzero.com. Available at: https://www.cloudzero.com/blog/data-driven-development#:~:text=Data%2Ddriven%20development%20is%20a.
52. Rudeck, E. (n.d.). *The Benefits of Data-Driven Design*. [online] www.concurrent-engineering.co.uk. Available at: https://www.concurrent-engineering.co.uk/blog/the-benefits-of-data-driven-design#:~:text=Data%2Ddriven%20design%20helps%20manufacturers.
53. ‌UИSCЯAMBL. (2021). *6 Inspiring Examples Of Data-Driven Companies (Key Takeaways Included) - Conversational Analytics & Business Intelligence*. [online] Available at: https://unscrambl.com/blog/data-driven-companies-examples/.
54. GetApp. (n.d.). *diagrams.net Reviews - Pros & Cons, Ratings & more*. [online] Available at: https://www.getapp.com/it-management-software/a/draw-dot-io/reviews/.
55. Anon, (n.d.). *Draw.io: Comprehensive Review on Features, Price, and More*. [online] Available at: <https://www.mindonmap.com/blog/drawio-review/>.
56. PCMAG. (n.d.). *Draw.io Review*. [online] Available at: https://www.pcmag.com/reviews/drawio.
57. pdfslide.net. (n.d.). *(PDF) Pros and cons of Microsoft Visio*. [online] Available at: https://pdfslide.net/software/pros-and-cons-of-microsoft-visio.html?page=5 [Accessed 1 Feb. 2023].
58. GetApp. (n.d.). *Visio Reviews - Pros & Cons, Ratings & more*. [online] Available at: https://www.getapp.com/it-management-software/a/microsoft-visio/reviews/.
59. www.softwareadvice.com. (n.d.). *Visio Software Reviews & Ratings | 2023 | Software Advice*. [online] Available at: https://www.softwareadvice.com/workflow/visio-profile/reviews/ [Accessed 1 Feb. 2023].

‌