**Acknowledgment**

First and foremost, we would like to express our gratitude to our Mentor, Prof. Bhupinder Singh, who was a continual source of inspiration. He pushed us to think imaginatively and urged us to do this homework without hesitation. His vast knowledge, extensive experience, and professional competence in subject enabled us to successfully accomplish this project.

This endeavour would not have been possible without his help and supervision. We could not have asked for a finer mentor in our studies. This initiative would not have been a success without the contributions of each and every individual. We were always there to cheer each other on, and that is what kept us together until the end.

We’d like to thank The University of Mumbai for providing us with the opportunity to work on the project Scrum Master. Last but not least, we would like to express our gratitude to our families, siblings, and friends for their invaluable assistance, and I am deeply grateful to everyone who has contributed to the successful completion of this project.

**Declaration**

I hereby declare that this project work entitled “SCRUM MASTER” has been prepared by me during the year 2023 – 24 under the guidance of Mr. Bhupinder Singh.

This project work is submitted in the partial fulfilment of

there acquirements for the award of the degree of Master of Technology in Artificial Intelligence.

I also declare that this project is the outcome of my own effort, that it has not been submitted to any other university for the award of any degree.

Date:

Avantika Jadhav.

(2294381)

&

Nandini Barve.

(2294376)

**Abstract**

First, as a reminder, Scrum is a subset of Agile software development. The 2020 scrum guide explains that “Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.”

A Scrum master is a facilitator and coach who helps the Scrum team and broader organization understand and apply Scrum theory and practice. The Scrum master helps the Scrum team work more effectively by skilfully removing obstacles and distractions that may impede the team from meeting goals. This individual is the liaison between the Scrum team and people or teams outside the Scrum team.

A Scrum master is a professional who leads a team through a project using Agile project management techniques. A Scrum Master facilitates all the communication and collaboration between leadership and team players to ensure a successful outcome.

**Table Content**

|  |  |  |
| --- | --- | --- |
| **Sr No.** | **Contents** | **Page No.** |
| 1 | Introduction |  |
| 1.1 | Objective and Scope |  |
| 2 | System Analysis |  |
| 2.1 | Identification of Need |  |
| 2.2 | Preliminary Investigation |  |
| 2.3 | Feasibility Study |  |
| 2.3.1 | Technological Study |  |
| 2.3.2 | Economical Study |  |
| 2.4 | Project Planning |  |
| 2.5 | Project Scheduling |  |
| 2.6 | Software Requirement Specification |  |
| 2.6.1 | Overall Description |  |
| 2.6.2 | User Class & Characteristics |  |
| 2.6.3 | Operating Environments |  |
| 2.6.4 | Design Constraints |  |
| 2.7 | Software Engineering Paradigm |  |
| 2.8 | Cost Estimation of Project |  |
| 3 | Review of Literature |  |
| 4 | Design |  |
| 4.1 | Diagrams |  |
| 5 | Implementation |  |
| 6 | Testing |  |
| 7 | User Interface Design |  |
| 8 | Conclusion & Future Scope |  |
| 9 | Bibliography |  |

**CHAPTER 1**

**Introduction**

Agile management emerged as the antithesis of predictive project management, a model that we will refer to frequently in this manual. Both models have their virtues and are more useful in specific industries. Predictive management focuses on planning,

calculating a budget, and setting deadlines. If the final product is delivered in time, without exceeding costs, and it includes all the functionalities of the initial plan, it is considered a success.

As reasonable as it sounds, this has many drawbacks when we try to apply it in constantly and rapidly changing industries. That definition of ‘successful’ serves in a stable environment, where products are the result of scrupulous attention to processes and protocols.

Predictive management is a result of the Industrial Revolution: it comes from the world of construction, automobiles, and factories. If the client is looking for a house, for example, it should be built in such a way that it is durable, safe, and meets the needs of its inhabitants. And, in an ideal scenario, within the planned timeframe and

without exceeding the cost.

But we can find plenty of products today that share nothing in common with the Industrial Revolution ones. Firstly, because they can be abstract, like a movie or a mobile app. You can try out new things during development, empirically testing what works and what doesn’t. Adjustments can be made at any given moment. And

you can start with a first sketch of the basics you need and work your way up. The scenario may change: a functionality that seemed essential at first may be outdated by the delivery date. Or a competitor may launch an exciting new feature that leads

to a review of the product’s priorities. Being competitive requires the ability to respond quickly in uncertain work scenarios. This means there are no stable requirements when designing new products or services. They need to be available for customers as soon as possible, and then continuously maintained and improved. In these products, innovation is a crucial value.

These and more reasons we will see led to question the predictive management model, which didn’t seem to fit the reality of what knowledge companies needed. Understanding as such those organizations that develop products or services based

on knowledge rather than tools and processes. The working environment of these companies is very different from the one that

originated predictive project management. Now, there are markets with such a rapid evolution that it is pointless to try to start projects with a closed plan. There is a need for strategies that deliver tangible results soon, and that allow responding in time to changes. The product is built at the same time as changes and new requirements are introduced. The client starts from a more or less clear vision, but the level of innovation required, as well as the speed at which the business environment moves, does not allow him to foresee in detail how the final result will be.

Today, there are product managers who do not need to know the 200

functionalities of the final product, or if it will be finished in 12 or 16 months. Some customers need to have the first version with minimum functionalities in a matter of weeks, instead of a complete product within one or two years. Their interest is to

quickly put a new concept on the market and increase its value over time.



*Agility starts from a viable minimum and develops the project by*

*adapting to the circumstances as they change*

**SCRUM**

Scrum is an agile development model characterized by:

• Autonomous and self-managed teams that share their knowledge openly and learn together.

• An ‘incremental’ development strategy rather than complete product planning.

• Basing the quality of the result on the tacit knowledge of people and their creativity. Not on the quality of the processes.

• Overlapping the different phases of development, instead of carrying them out one after the other in a sequential or ‘waterfall’ cycle.

The origin of the term is far removed from that of project management: it comes from rugby. ‘Scrum’ defines the formation in which both teams, crouching and clinging to each other, push for the ball without touching it with their hands.

But for our purposes, we have to go back to 1980s Japan when researchers Ikujiro Nonaka y Hirotaka Takeuchi gave the term a polysemic dimension.

They identified a novel form of development in the industrial manufacturing companies that were obtaining the best results in innovation and time to market: Fuji Xerox, Canon, Honda, Nec, Epson, Brother, 3M y Hewlett-Packard (Nonaka 1986).

They compared their way of working in self-managed teams with the way rugby players advance when in scrum formation, hence the term.

Although this way of working emerged from technology product companies in industrial manufacturing, it started to be also applied to the software industry from 1995 onwards. That year, Ken Schwaber presented in OOPSLA (the Object-Oriented Programming, Systems, Languages & Applications annual conference) a software

development methodology based on a scrum environment, using that same term (Schwaber 1995). This first framework presented a series of phases and ‘artifacts’: pregame, game, postgame, planning, sprints, wrap… Some of them are still in use,

and we will see them. But in general, the rules of the game have changed a lot since then. There is no single authority that determines what ‘scrum’ is and is not. It has changed over time, and it will continue to evolve with the input of the professional

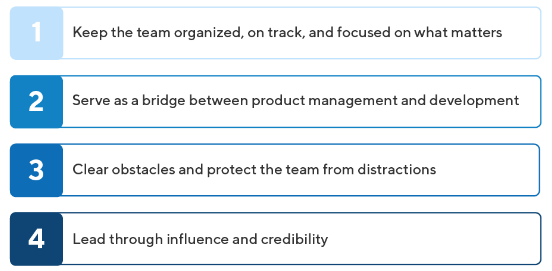
community, which defines the most useful practices. The original spirit, however, remains: practices should help teams to self-manage and maintain a continuous flow of progress, producing results iteratively and frequently.

Among the ‘events’ and practices that have been added over the years, we can find, for example, retrospective meetings, refinement product backlog meetings, DoR (Definition of Ready), story maps...

Scrum Manager® uses the term ‘scrum’ with its original meaning, the one given by Nonaka and Takeuchi.

**Objectives and Scope**

Often a developer who is responsible for keeping the dev team on track. The scrum master’s main objective is to make sure the team works according to the agile values. That they have the resources, time, and disruption-free environment to succeed.



**4 Key Responsibilities of a Scrum Master**

**1. Keep the team organized, on track, and focused on what matters.**

A scrum master’s primary responsibility is to keep the development team organized and progressing on the projects they’ve agreed to, in order of priority. The Scrum Master, in other words, is there to help keep the team on track.

For example, the scrum master might lead the team’s daily stand up (if the team holds them). In these brief morning meetings, the team discusses what they’ve accomplished, what they’re planning to work on that day, and any obstacles they face completing their tasks.

### 2. Serve as a bridge between product management and development.

One ongoing risk in creating any product is that the product management team does not clearly communicate its strategic vision to the developers who will be building the product. If the big-picture[product strategy](https://www.productplan.com/learn/guide-to-product-strategy/) becomes lost in translation, the developers might focus on the wrong priorities or make decisions—to save time, for example—that undermine the product vision.

In an agile organization, the scrum master can mitigate these risks. They act as a communication bridge between product management and development.

### 3. Clear obstacles and protect the team from distractions.

When the developers discover a dependency or other challenge preventing them from completing a sprint task, it will be up to the scrum master to clear that obstacle so the team can continue its work.

When a product manager goes directly to the dev team with an urgent last-minute request—for example, to create custom code for a large customer—it will be the scrum master’s responsibility to remind the PM that the team can work only on the tasks agreed to in the current sprint.

### 4. Lead through influence and credibility.

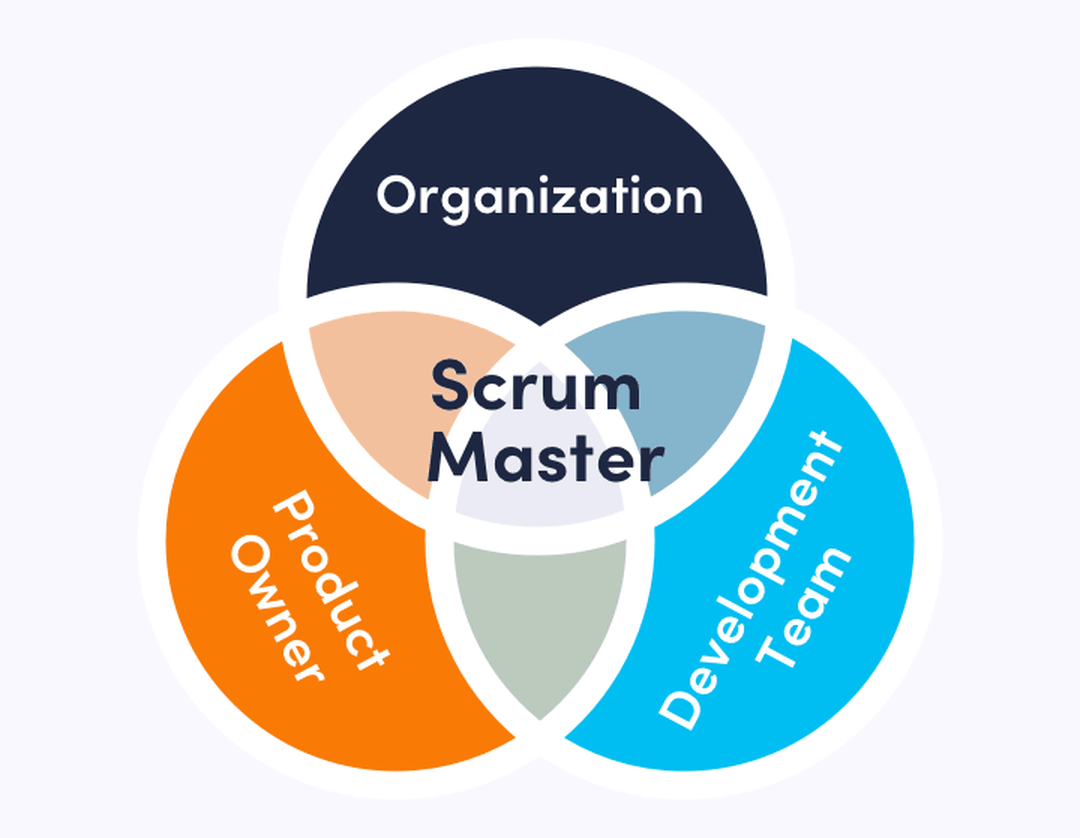
As the Agile Alliance explains, a Scrum Master has a lot of responsibility but no real authority in the company hierarchy. (Product managers: Sound familiar?) This is one reason some companies describe their scrum master as a team coach or agile coach. These professionals lead their team, but they must do so from the earned position of a trusted expert and guide—not as a manager or supervisor who can dictate orders.

To succeed in this role, a Scrum Master needs to build credibility with the team. This won’t happen immediately but will have to be earned over time. Here are some of the strategies you can use as a Scrum Master to earn this trust.

**CHAPTER 2**

**IDENTIFICATION OF NEED**

The Scrum Master is a servant-leader for the Scrum Team. The Scrum Master helps those outside the Scrum Team understand which of their interactions with the Scrum Team are helpful and which aren’t. The Scrum Master helps everyone change these interactions to maximize the value created by the Scrum Team.



According to the Scrum Guide, Scrum Master serves on three levels:

* Product Owner by taking over tasks like facilitating and organizing events;
* Development Team by coaching the development team in self-organization and cross-functionality;
* Organization by helping to understand and implement Scrum as well as to reach the best level of productivity.

**5 reasons why you need a Scrum Master**

So, the big question is, does it make any sense to have a certified Scrum Master in your company? To cut the long story short, it is not possible to enjoy the full benefits of Scrum without the Scrum Master. Without the coach, the team may become a complete mess. Here are the top 5 things a Scrum Master can help your organization with:

1. Implement best practices on project management

A Scrum Master takes care of building and onboarding project teams so they could cooperate well on the particular project. He/she helps a Product Owner get a clearer vision of the product and communicate it to the development team in order to make that vision a reality. Scrum Master implements the best agile practices in the development process in order to develop good quality software in an effective manner. One of the roles of the Scrum Master is to also provide feedback to the team and drive the culture of agility and learning. This has a positive impact on the continuous improvement of a Scrum team.

1. Keep all parties well informed and on track

As a Scrum Master, I host and facilitate the scrum teams' meetings, especially daily scrum meetings, in order to get updates on the progress of the project. Those meetings help to identify potential obstacles and find the right solution to them. What is more, our role includes ensuring the project stays on track and the team meets the deadlines with the desired outcome.

1. Introduce Agile engineering practices

Usage of continuous integration (CI) and testing environment are the Agile engineering practices the Scrum Master will encourage the development team to implement. He/she also ensures the code quality and refactoring practices are addressed in the development team. Another Agile technique Scrum Master advocates is a pair-programming session, where two developers collaborate in real-time solving an issue or implementing a new feature. Altogether, these engineering practices reduce development time and improve the quality of the product.

1. Make the Scrum team effective

A Scrum Master is concerned with creating an environment where the team can be effective by facilitating daily scrum meetings (standups), setting up retrospectives, sprint reviews and sprint planning sessions. We help the team to maintain the burndown charts that show if the sprint is progressing according to the schedule. This technique helps detect any issues as soon as they appear so that they can be discussed during daily stand-ups and focus on resolving them early to keep up with the pace.

A Scrum Master also shields the team from any interruptions during the sprint and removes obstacles and distractions which means maximum efficiency of the development team.

1. Cooperate with the Product Owner

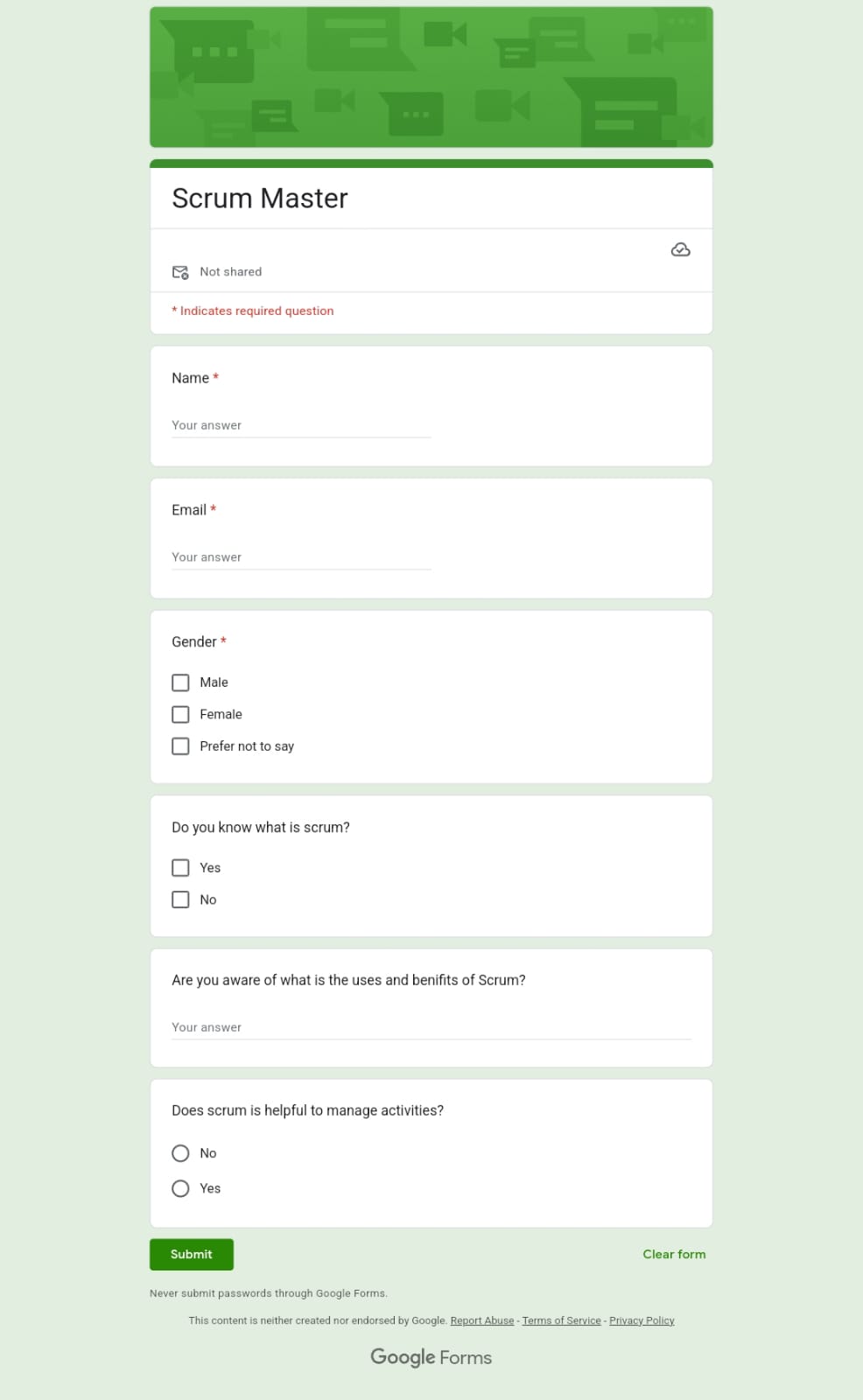
Cooperation with the Product Owner is also an important role of the Scrum Master. While the Product Owner is responsible for the product being created using Scrum, the Scrum Master is the “process owner” who keeps the team on track procedurally. Working closely with the Product Owner, Scrum Master helps ensure to keep the product backlog in a good shape for each sprint. Scrum Master helps to walk the Product Owner through the more technical user stories while encouraging collaboration with the Scrum team. Scrum Masters keep a good relationship between the team and the Product Owner as well as others outside the team.

**2.2 PRELIMINARY INVESTIGATION**

To get a general idea of what people might know about scrum master and what are their opinions about them I did a mini survey. I asked some of my friends to fill the form and collected their responses. I got responses from ( females and males) people and the results are as follows:

Survey link:-

[**https://forms.gle/sLhSeMc1eegFC1Wn9**](https://forms.gle/sLhSeMc1eegFC1Wn9)

****

**Conclusion from the survey**

* Exploring cultural differences in the use of strategies;
* Exploring why it is that Scrum Masters don’t consider other Scrum Masters and/or Agile Coaches as important allies;
* Delving more deeply into what kind of internal politics work. The use of politics often has a bad smell associated with it. But many kinds of politics, like the building of coalitions, finding allies and building connections, are vital to change efforts.

#### **FEASIBILITY STUDY**

A feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. A quality information system costs money, time and requires a consistent step by step well thought out approach in its development.

The correct application of information technology is a critical success factor for the optimal functioning and competitive position of an organization, therefore, the integration of social needs, the resources and the possibilities offered by technology in an organization is required.

After doing a little research and study on this topic I came to the conclusion that this project is feasible. It is technologically, Legally, Economically and Operationally feasible. I am going to use all the existing technologies and open-source languages in the project. This project is not going against any law and it is economically feasible.

**TECHNOLOGICAL FEASIBILITY STUDY**

* This project uses the technical resources which are available commonly.
* The technical resources also fulfills all the demands for the enhancement of the application. This application is highly feasible as it can be built using the prevailing technologies.
* It is basically an android application so all the users can access it anywhere, anytime easily

**ECONOMIC FEASIBILITY STUDY**

* Taking into consideration the benefits and costs of the application helps to determine the feasibility, costs, and benefits associated with a project before financial resources are allocated.
* As the cost of the project is zero, this project is economically feasible, as I am using free version of the software.
* The system will be beneficial on the perspective of economy, because the major technology we are using to build this system are open-source technology.

**LEGAL FEASIBILITY STUDY**

* This assessment investigates whether how well a application can be implemented within existing legal policy.
* All the aspects of this application are not against any legal requirements.
* This application is legally feasible.

**OPERATIONAL FEASIBILITY STUDY**

* This assessment involves undertaking a study to analyse and determine whether and how the needs can be met by completing the project.
* This project is operationally feasible as almost everyone uses smartphones these days and they have access to the internet.
* This project is going to help users to gain more knowledge about different fields and help them find their ideal career choice.
* This project will be planned properly before going to the implementation part.
* I will be taking all the requirements of the users into consideration so that it can meet the users’ needs.

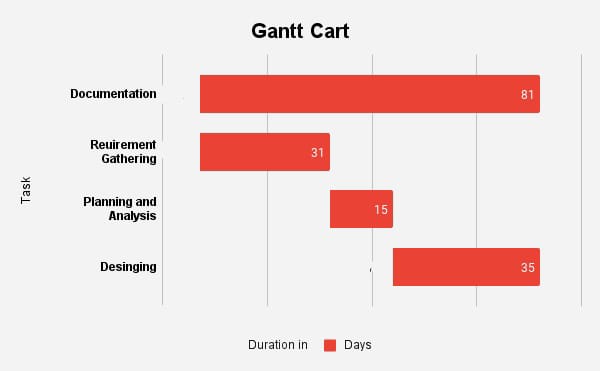
**PROJECT PLANNING**

|  |  |  |  |
| --- | --- | --- | --- |
| **PROJECT PLANNING** | | | |
| **Sr no** | **Task Name Duration** | **Start** | **Finish** |
| 1 | Analysis 7 weeks |  |  |
|  | Investigation Data Gathering 35 days | 02/11/2023 | 12/11/2023 |
|  | Feasibility study 5 days | 13/11/2023 | 18/11/2023 |
|  | Scheduling 7 days | 19/12/2023 | 25/12/2023 |
| 2 | Design 8 weeks |  |  |
|  | Modular Design 7 days | 26/12/2023 | 02/01/2024 |
|  | Database Design 7 days | 03/01/2024 | 10/01/2024 |
|  | User Design 14 days | 11/01/2024 | 15/01/2024 |
|  | Test Cases 14 days | 16/01/2024 | 30/01/2024 |

#### **PROJECT SCHEDULING**

**Gantt chart:**

* A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time.
* A Gantt chart is a project management tool assisting in the planning and scheduling of projects of all sizes.
* On the left of the chart is a list of the tasks involved in this project and on the top is a time scale.
* Task duration of each activity is represented by a bar, it is calculated by start date and the end date of each task.

****

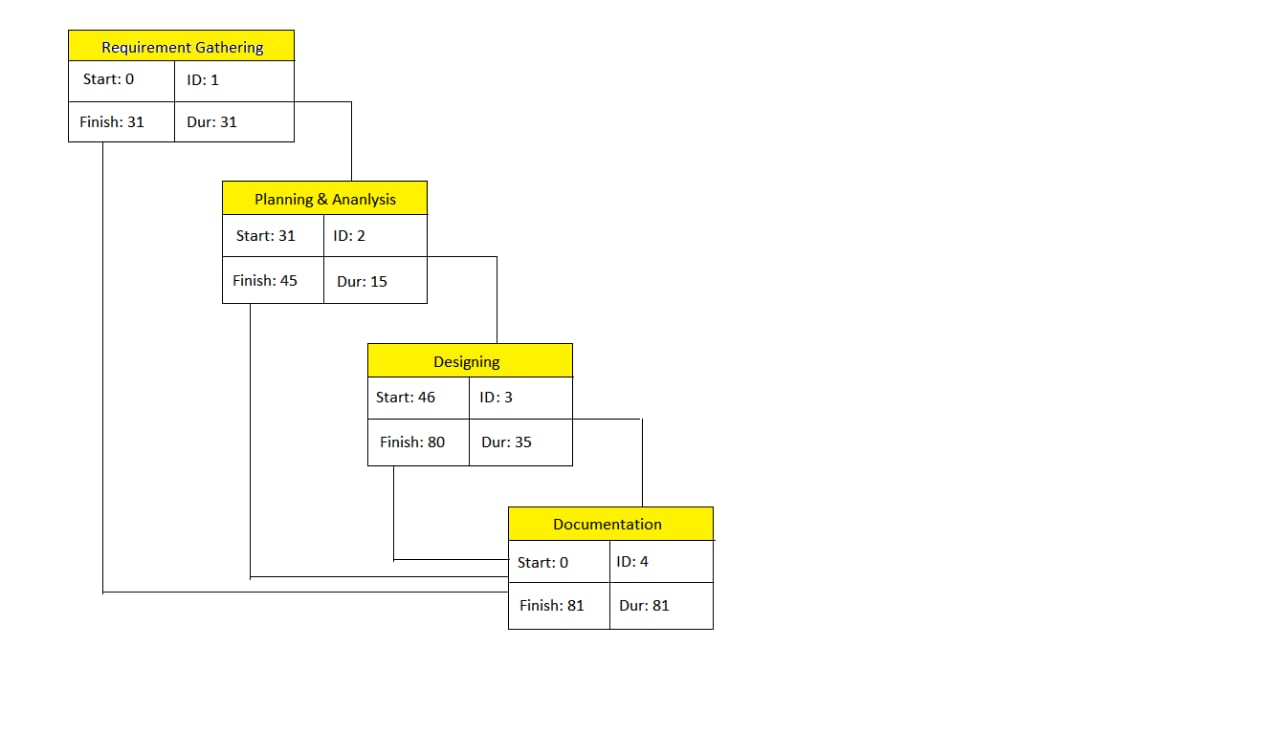
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Activity | Documentation | Requirement  Gathering | Planning &  Analysis | Designing |
| Start Date | 01-11-2023 | 01-011-2023 | 01-12-2023 | 06-01-2024 |
| Duration | 81 days | 31 days | 15 days | 35 days |

**PERT Chart:**

A PERT chart helps to analyse a project's tasks and estimate the amount of time required to complete each task in the project.

PERT stands for “Program Evaluation and Review Technique”.

It basically gives an idea of total time required for doing each and every task while developing a project. In other words it is used for planning, scheduling, organizing and coordinating tasks of the project. Now-a-days PERT chart is not commonly used for planning purpose instead.

****

**SOFTWARE REQUIREMENT SPECIFICATION**

**HARDWARE REQUIREMENTS**

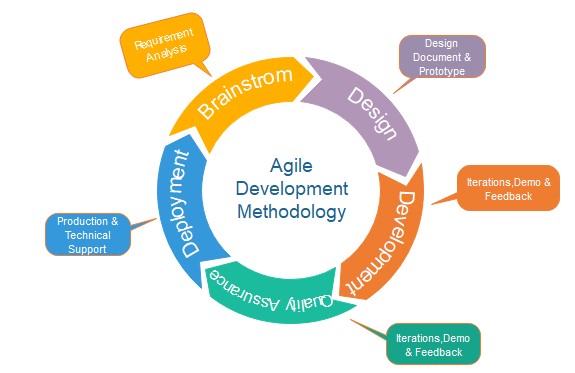
* + - **3 GB RAM** to **8GB** recommended
    - **2GB** HDD space but **4GB** recommended.
    - Windows, Linux or Mac all platforms are OK.
    - Monitor resolution needs to be **1280x800** minimum.

**SOFTWARE REQUIREMENTS**

#### **SOFTWARE ENGINEERING PARADIGM**

For developing this application Agile model is used. The meaning of Agile is swift or versatile. "Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing.



**Different Phases of Agile:-**

1. **Requirements gathering:** In this phase, we defined the requirements. We had to plan the time and effort needed to build the project. Based on this information, we were able to evaluate technical and economic feasibility
2. **Design the requirements:** When we identified the project, work we started to define requirements. We started to create user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing application.
3. **Construction/ iteration**: After defining the requirements, we started coding our project and started to implement the best design for the user interface. The project had to undergo various stages of improvement, so it includes simple, minimal functionality.
4. **Testing:** In this phase, my team examined the product's performance and we also looked after the bug or loop holes.
5. **Deployment**: In this phase, a product is deployed in the user's work environment.
6. **Feedback:** After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

**COST ESTIMATION OF PROJECT**

The Scrum Master application is build using most of the technologies are already existing and are easily available. All the resources that we are going to be used in this application are open source. For hosting this application, we are using local host and the free version of the 000webhost server, so the cost estimation for developing this application is nothing till now. If required for future scope we can use the better technologies or paid versions of hosting and the costing of that can be included later.

**CHAPTER 3**

## Review of Literature

## Although the Scrum method is popular on the Internet and at companies, it is not a simple task to find scholar material on the subject. However, this study shows that this scenario is about to change. The increase in publications on Scrum has been remarkable along the years (Figure 2). For instance, if this study had been carried out in 2006 only 11 articles would have been found in our database.

## This growth may also be seen in (Figure 3), which shows that 73% of the literature on Scrum was published in the last two years (2007 and 2008). Besides that, taking into account that the research was carried out in October 2008, we believe these numbers have grown even more since then. A hypothesis for this increase along the years is the gradual implementation of the method by companies, which, consequently, is leading scholars toward the subject.

## Another aspect to be pointed out is the large concentration of publications about Scrum available at the IEEE and ACM database. Those two databases together account for 94% of the publications found. (Figure 4) shows that as well as the participation of AIP and Science Direct.

## (Figure 5) shows the result of the classification by type of study. As we can see the most common types are Case Study and Theoretical/Conceptual methods. Perhaps the low number of those classified as Literature Review is due to the lack of material on the subject. The few research studies which were classified as Survey and Action Research may disclose how immature the subject is

Next, as it can be seen in (Figure 6), the method of work investigation shows that most studies were qualitative. There were only three quantitative articles which included Salo and Abrahamsson (2008) e Sulaiman et al. (2006).

As expected, due to how young the subject is, no article was found which could be classified as retrospective analyses. As we can see in (Figure 7) almost all the articles are current, with the exception of the work done by Mann and Maurer (2005), which performs a longitudinal analyses (a two-year case study) to measure the impact of Scrum on client satisfaction.

(Figure 8) brings information that deserves our attention. It shows the authors affiliation. Differently from what we see with other subjects, most Scrum researchers are in companies, mainly software related and not in universities.

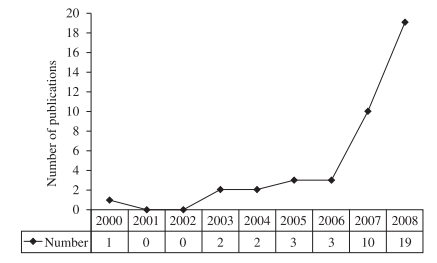


Figure 2 : Number of publications by year of publication.

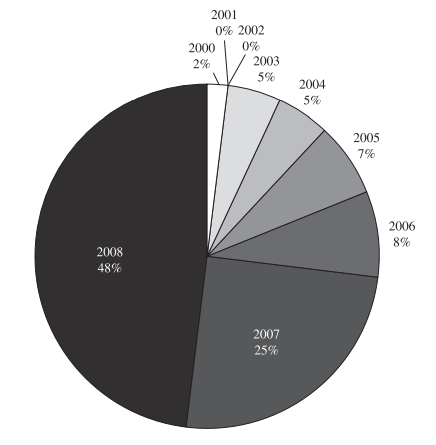


Figure 3: Publications distribution by year of publication.

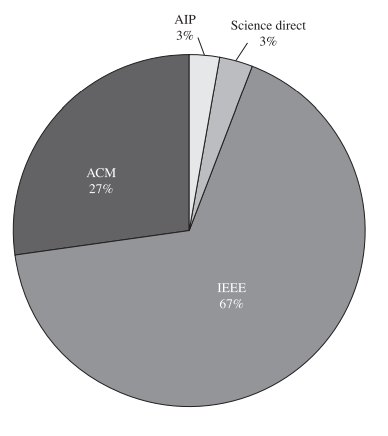


Figure 4: Publication distribution by database.

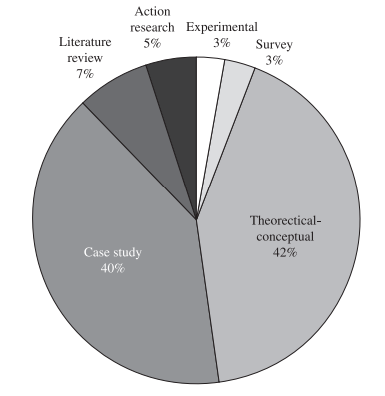


Figure 5. Publication distribution by method of study.

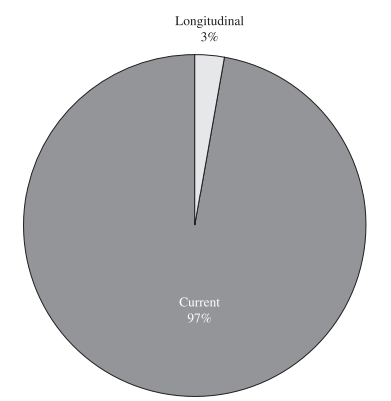


Figure 7. Publication distribution by analyses period.

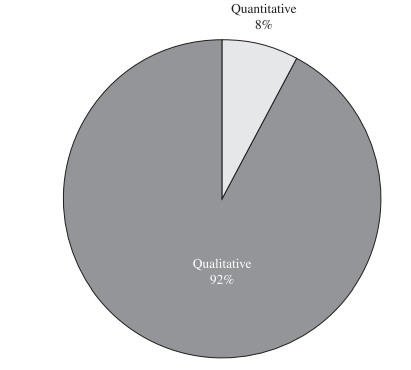


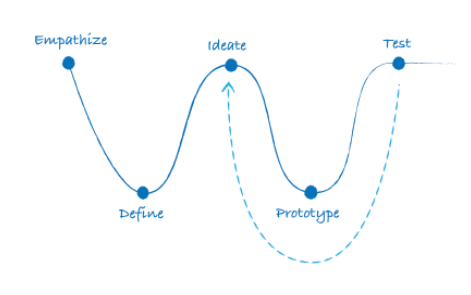
Figure 6. Publication distribution by approach.

**CHAPTER 4**

**Design**

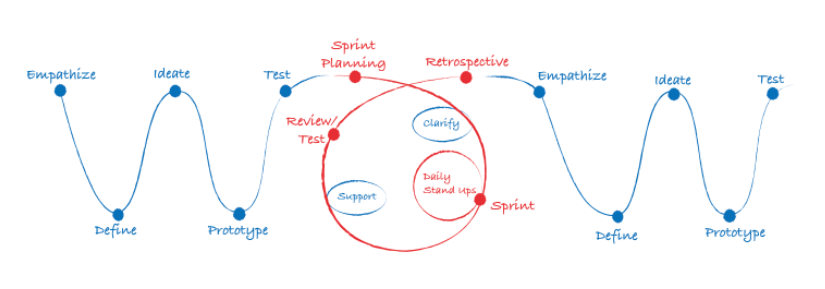
The Design Thinking Process and the Agile Scrum Process can be confusing terms in design and development circles, but I think it is important to understand the differences and how they can be used together. As a UX-designer you have probably, or will, stumble upon it somehow.

As you may know, or can read about in my earlier article, Design Thinking is a mindset focused on creatively solving problems and creating innovative solutions. It's often used during the discovery process to understand users and business needs, and can be seen in tools and exercises such as design sprints. Scrum on the other hand, is an efficient Agile development process that divides work into small, achievable chunks called sprints. It's typically used to deliver working software quickly and efficiently. By combining Design Thinking and Scrum, teams can effectively and quickly solve complex problems, create innovative solutions, and deliver high-quality products that meet the needs of users. It's important to understand the strengths of both approaches and how they can work together to achieve success.

****

Design Thinking is a human-centered approach to problem-solving that involves empathy, experimentation, and iteration. The process typically involves five steps: empathize, define, ideate, prototype, and test.

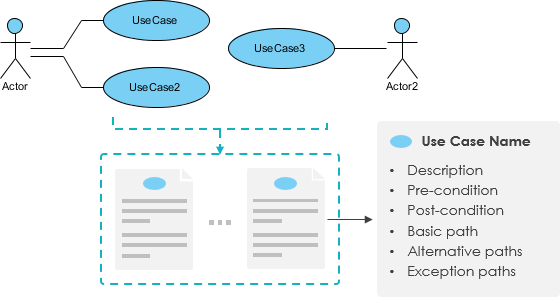
1. **Empathize:** In this step, we try to understand the needs, wants, and pain points of the users by gathering data through methods such as interviews, focus groups, and usability testing. This helps me to create a deep understanding of the user and their context, and identify opportunities for innovation.
2. **Define:** Defining the problem that we are trying to solve based on the insights that we have gathered from the empathize step. We also identify the goals and objectives of the project, and define the user needs and business goals that the solution should address.
3. **Ideate:** Generate a wide range of ideas for potential solutions to the problem. We use techniques such as brainstorming and Mind Mapping to come up with as many ideas as possible, collaboration and divergent thinking.
4. **Prototype:** Creation of low-fidelity prototypes of the ideas that we generated in the ideate step. Prototyping allows for testing of the ideas quickly and cheaply, and gather feedback from users and stakeholders.
5. **Test:** Testing the prototypes with users and gather feedback to refine and improve the solution. We also gather data through methods such as usability testing and A/B testing to validate assumptions and understand how the solution performs.



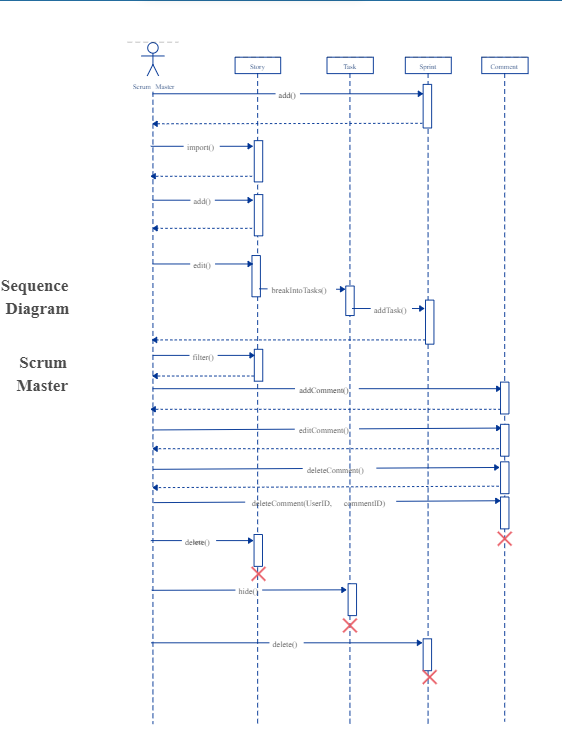
**Use Case Diagram:**

Detailed Use Case Specification

A Use Case Specification is a textual description of the functionality provided by the system. It captures actor-system interaction. That is, it specifies how a user interacts with a system and how the system responds to the user actions. It is often phrased in the form of a dialog between the actor and the system. The Use Case Specification is represented in the Use Case Diagram by an oval, and is what most people think of when they hear the term Use Case.



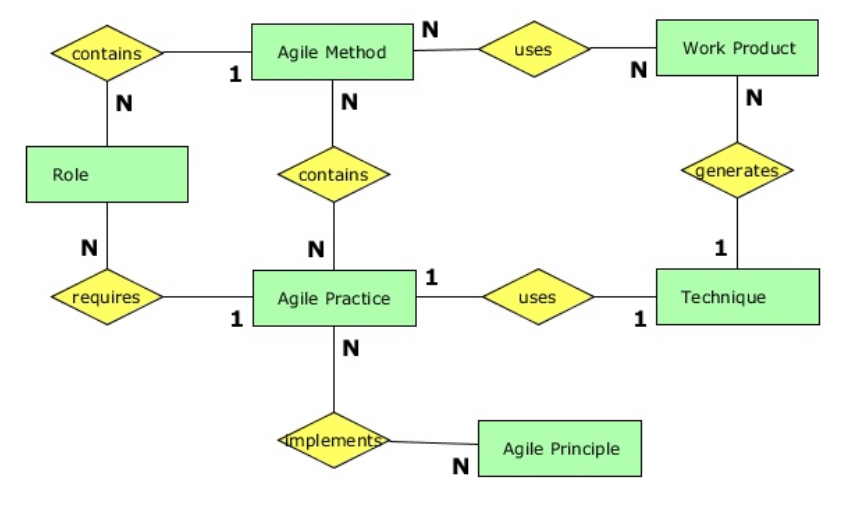
**Sequence Diagram**



Use Creately’s easy online diagram editor to edit this diagram, collaborate with others and export results to multiple image formats.

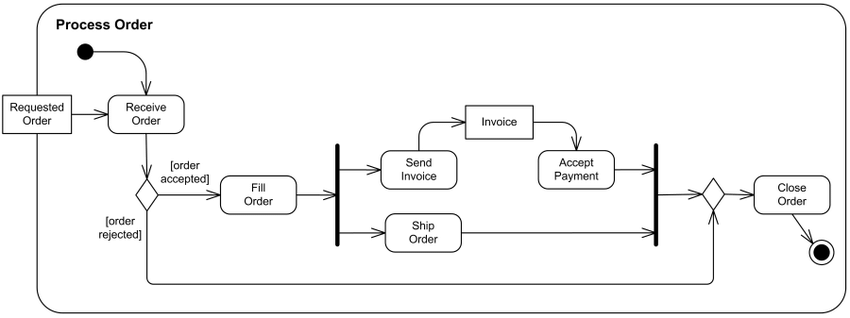
**Entity Relationship Diagram**

An Entity Relationship Diagram is a diagram that represents relationships among entities in a database. It is commonly known as an ER Diagram. An ER Diagram in [DBMS](https://www.simplilearn.com/what-is-database-management-article) plays a crucial role in designing the database. Today’s business world previews all the requirements demanded by the users in the form of an ER Diagram. Later, it's forwarded to the database administrators to design the database



**Activity Diagram**

* Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.
* An activity diagram is a behavioural diagram i.e. it depicts the behaviour of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed
* Activity diagram is basically a flowchart to represent the flow from one activity to another activity.
* The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

****

