**LinkedHU\_CENG**

**Project Plan**

1. **Introduction**

The Project Plan Document describes the entire life cycle of the project, which is a student-based social media platform, from the design phase to the deployment phase. This document determines the roles and competence level of the project team, the main strategies to be followed in the project process and the techniques that measure the performance of these strategies, the objectives and milestones of the project, the deployment of the project and the maintenance method after this deployment phase.

1. **Project organization**

Each member of the "Exploding Gradients" team has a specific role besides being a software developer. Each member is responsible for the related tasks that his/her role fits with the current problem or job description. The roles have been assigned to team members as follows:

| **Team Member** | **Software Developer** | **Software Project Manager** | **Software Analyst** | **Software Architect** | **Software Configuration Manager** | **Software Tester** |
| --- | --- | --- | --- | --- | --- | --- |
| Sümeyye Meryem Taşyürek | X | X |  |  |  |  |
| Humeyra Uçar | X |  | X |  |  |  |
| Mert Doğramacı | X |  |  | X |  |  |
| Murat Çelik | X |  |  |  | X |  |
| Tuğçe Kızıltepe | X |  |  |  |  | X |

The project will be acted by consensus of all team members. In addition to advancing through consultation during the project process, we have identified a main and an assistant team member for the roles. In this way, it is aimed to provide a control mechanism in the process. Our assistant roles were planned as follows:

Murat Çelik - Software Project Manager

Sümeyye Meryem Taşyürek - Software Tester

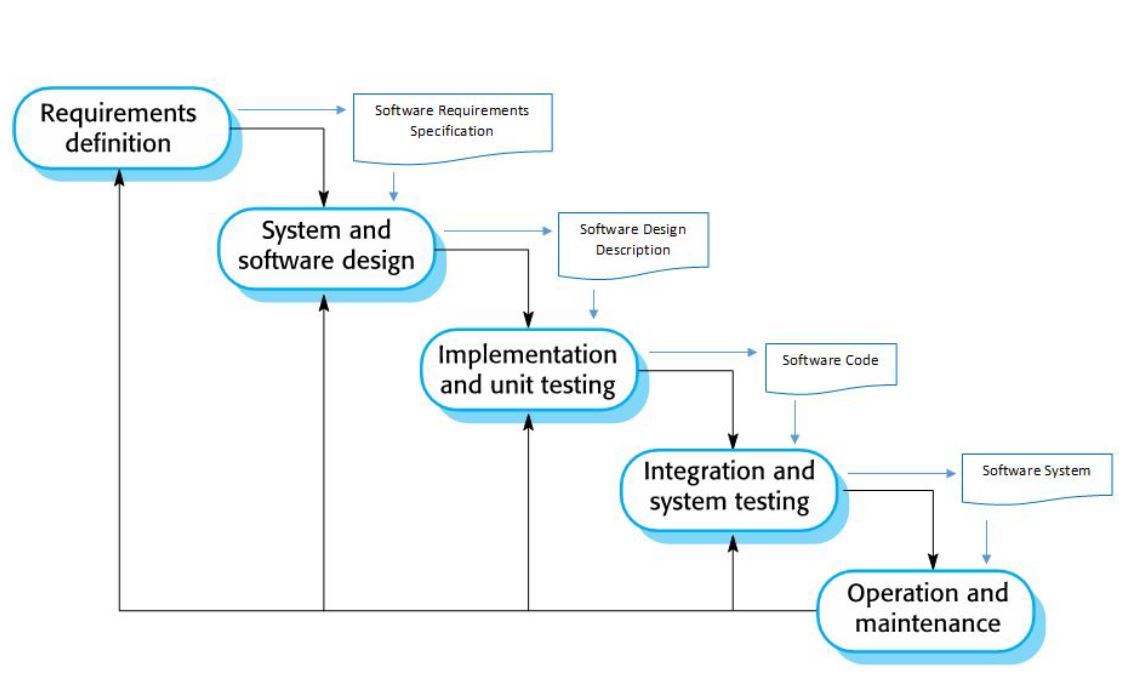
Humeyra Uçar - Software Architect

Mert Doğramacı - Software Configuration Manager

Tuğçe Kızıltepe - Software Analyst

1. **Development process and measurements**

We will apply the Waterfall model in the development process of the project. This model has five common phases with each stage completely wrapped up before the next phase begins. The success of the Waterfall model depends on the amount and quality of the work done, in advance documenting everything. In principle, the result of each phase is one or more documents that have been approved.



The software process of the Waterfall model is not just a simple linear model, instead it involves feedback from one phase to another. Also, the plan-driven nature of the waterfall model helps to coordinate the work.

**Requirements**

The Waterfall methodology depends on the belief that all project requirements can be gathered and understood upfront. The system’s services, constraints, and goals are established. Then, they are defined in detail and served as a system specification.

**Design**

Here, software developers design a technical solution to the problems set out by the project requirements, including scenarios, layouts, and data models. Also, they allocate the requirements to either hardware or software systems by establishing an overall system architecture.

**Implementation**

When the design is finished, the technical implementation begins. In this phase, the software design is realized as a set of programs or program units.

**Verification or testing**

Before a product can be released to customers, testing needs to be done to ensure the product has no errors and all of the requirements have been met.

**Deployment and maintenance**

The maintenance phase begins once the software is deployed to the market or released to clients. Maintenance involves correcting errors which were not discovered in earlier stages, improving the implementation of system units and enhancing the system’s services as new requirements are discovered.

| **Tasks** | **Estimated Time Interval** | **Work Load (in Weeks)** |
| --- | --- | --- |
| Software Project Management | 18.02.2022 - 13.05.2022 | 13 weeks |
| Software Project Vision & Plan | 18.02.2022 - 04.03.2022 | 1.5 - 2 weeks |
| Software Project Requirements | 04.03.2022 - 18.03.2022 | 2 weeks |
| Software Project Design | 18.03.2022 - 08.04.2022 | 3 weeks |
| Software Project Development | 28.03.2022 - 13.05.2022 | 7 weeks |
| Software Project Test | 28.03.2022 - 13.05.2022 | 7 weeks |

In the table above, estimated time intervals are inscribed. Besides, we are planning to do our weekly meetings on Mondays for 2 hours, at the beginning of the week to plan our weekly subtasks. Also, we might do weekly reviews on Thursdays for 1 hour about how our subtask is progressing. We are planning to use Jira software to assign and categorize tasks, specify deadlines and watch our progress.

1. **Project milestones and objectives**

| **Phase** | **Iteration** | **Primary objectives** (risks and use case scenarios) | **Scheduled start or milestone** | **Target velocity** |
| --- | --- | --- | --- | --- |
| Inception | I1 | **Objectives**   1. Software Vision Document 2. Project plan document   **Risks**   1. Poor project planning and time management. 2. The project budget may be underestimated.   **Mitigate Risk 1:** Keep up with the timeline and be aware of deadlines.  **Mitigate Risk 2:** To not overrun the budget, track the budget and plan for contingencies. | 07.02.2022 / 04.03.2022 | 6 hours |
| Planning | I2 | **Objectives**   1. Software Requirements Document 2. Use Case Definition 3. Test Case Definition 4. Graphical User Interface Design   **Risks**   1. The software tools to be used may be insufficient or incorrectly selected. 2. Software requirements may be insufficient or poorly defined. This can prevent time management and budget management from being done correctly.   **Mitigate Risk 1:** The software tools to be used are investigated in detail and the most applicable one is used.  **Mitigate Risk 2:** Requirements at the high-level are understood and initial scope of work is determined. Requirements are identified as a result of communication and agreement between the stakeholders and the development team. The requirements are documented. | 04.03.2022 / 18.03.2022 | 20 hours |
| Elaboration & Development | I3 | **Objectives**   1. Architectural Notebook 2. List of System Test Case Definitions 3. Risk Management Report 4. Configuration/Change Management Report 5. Demo Presentation   **Risks**   1. Test cases may be insufficient or required test cases are not defined. 2. Software architecture is determined but it may not meet constraints.   **Mitigate Risk 1:** Requirements are defined in line with the customers' requests and test cases are defined accordingly.  **Mitigate Risk 2:** Before starting development, make sure that the software architecture is correct and does not conflict with constraints. | 18.03.2022 / 08.04.2022 | 30 hours |
| Development & Monitoring & Controlling | I4 | **Objectives**   1. Software Design Document (UML models) 2. Coding Standard   **Risks**   1. If technologies are not chosen successfully or in accordance with constraints, this can cause a waste of time. 2. Clean code may not be used or the code may not comply with coding standards.   **Mitigate Risk 1:** While choosing technologies, it is necessary to take into account the performance, easinees, reliability, and constraints  **Mitigate Risk 2:** Coding standards must be specified before starting development, then must be followed by the development team. | 08.04.2022 / 29.04.2022 | 25 hours |
| Test & Deployment | I5 | **Objectives**   1. Software Test Result Report 2. Risk Management Report 3. Configuration/Change Management Report 4. Presentation of the product 5. Final demo of software product   **Risks**   1. During testing, some significant bugs are reported. These bugs may require change of the implementation. 2. The software product may not be completely implemented.   **Mitigate Risk 1:** In the development phase, test carefully and do unit tests.  **Mitigate Risk 2:**  While the product is being developed, be aware of the deadlines. | 29.04.2022 / 13.05.2022 | 30 hours |

1. **Deployment**

Since we are using the Waterfall Model while implementing our project, a phase of the project has to be completed before moving onto the next phase. Therefore, before handling the deployment, implementation and all testing phases should be done successfully. Deployment must follow the project schedule. Except for that, the source codes of the project are planned to be updated every week or biweekly via GitHub.

In the progress of the project, the best deployment tool will be selected by considering the whole process.

The final product will be available for users to download from GitHub.

1. **Lessons learned**

* To manage the project process successfully, first of all, the team should organize well. Each person’s responsibilities should be clear and determined before starting.
* To prevent potential problems in the process, every team member should follow the schedule.
* Before starting the project, the team should analyze and determine project requirements according to the Waterfall Model. It may be difficult to fix the problems after some points.
* It is important to determine the estimated times to the parts of the process and do not waste extra time on a specific problem. This may cause you to miss deadlines or have to do last minute work.
* Communication between team members is as important as making correct planning. The communication should be continuous and related members should communicate more. This is teamwork and everyone should follow teamwork rather than personal work or personal decisions.
* To write the documents well at every step improves both the quality of the work and the product.