Project Plan

FileSender

SURF

Product Manager: William van Santen

Mentor: Rogier Spoor

Windesheim University of Applied Sciences

Supervising Teacher: Rob Kaesehagen

Student: Aaron Jonk (s1170298)

Date: 18/05/2025

Version: 1.1

Version management

Version	Date	What
0.1	06/02/2025	Initial draft of the project plan.
0.2	10/02/2025	Revised based on feedback from William van Santen and Rogier Spoor.
1.0	11/02/2025	Changed project methodology from just scrum to a combination of iterative waterfall & scrum. Updated the project planning to stay in line with the methodology.
1.1	18/05/2025	Review based on feedback from Jan Meijer & Guido Aben (both part of the FileSender board), and a clean-up.

Distribution

Version	Date	Recipient
0.1	06/02/2025	William van Santen & Rogier Spoor – For review, via email.
0.2	10/02/2025	William van Santen & Rogier Spoor – For review, via email.
1.0	11/02/2025	Rob Kaesehagen – For review, via OnStage.
1.0	18/02/2025	Published on Codeberg
1.1	18/05/2025	Published on Codeberg

Contents

1 Introduction

This document outlines the project plan for FileSender, a file-sharing platform with over 15 years of history. Originally created by two students, the project was later further developed by two interns, evolving into a fully functional, production-ready system.

Over the years, FileSender has grown significantly, with numerous features added to enhance its functionality. However, this ongoing development has also increased the complexity of the program. Several security vulnerabilities have been identified (some publicly disclosed and others not) raising concerns about the overall security and stability of FileSender.

With 15 years of accumulated code, the program has gradually become difficult to maintain, resembling a complex maze of interdependent components.

SURF has determined that a complete rewrite is necessary to address these issues while introducing new features such as end-to-end encryption for large files. This project aims to deliver a modernized, secure, and maintainable version of FileSender, incorporating stakeholder feedback throughout the development process.

2 Project Definition

2.1 Project Description

The primary goal of this project is to fully rewrite the current FileSender platform while incorporating additional features requested by stakeholders. The official project start date is yet to be finalized. However, the programming process of the project will formally begin once the initial documentation required for the start phase is completed.

The project started on February 3, 2025, when we began engaging stakeholders to gather their requirements and feature requests. We anticipate that the programming phase will start between February 17 and February 28, 2025, at the same time the documentation is expected to be ready

The project will remain in service until the end of June 2025, which is when the MVP is targeted for completion. The MVP will serve as a production-ready version of the new FileSender implementation. However, there are additional features that will need to be implemented after the MVP is delivered. At this point, it is difficult to estimate how much longer the full development process will take beyond the MVP phase.

2.2 Project Deliverables

2.2.1 Initial project documentation

One of the key deliverables is the initial project documentation. These documents outline the project plans leading up to the completion of the MVP and may include guidance for the further continuation of the project.

2.2.2 Software (MVP)

This deliverable is the MVP for FileSender version 4. The MVP will incorporate a selection of the core features from previous versions while integrating newly designed features based on stakeholder requirements. The delivered product should be production-ready, fully testable, and thoroughly documented, including clear comments within the code. Additionally, the product should come with a Docker image and be deployable as a Linux service.

2.2.3 CI/CD pipelines

The CI/CD pipelines will be an essential component of the project, ensuring continuous testing and seamless integration. All pull requests to the repository will be automatically tested through the pipelines. These pipelines will also be delivered as part of the final project deliverables.

2.2.4 Final documentation

At the conclusion of my internship at SURF, I will provide documentation to support the project's continuation. This will include instructions on how to:

- Continue development and extend the project.
- Set up a development environment.
- Deploy and install the new FileSender in a production environment.

2.3 Scope

The scope of this project includes the full redevelopment of existing FileSender features with additional features requested by stakeholders. However, the project does not include the installation of the program for the system administrators. Detailed installation instructions will be provided on this matter upon delivery.

3 Approach

3.1 Project Methodology

We will be using a customized project management methodology that combines Iterative Waterfall and Scrum elements. This hybrid approach provides the project with a structured framework for flexibility and iterative development. The project is divided into three key phases: Preparation, Development, and Wrap-Up, with the main development phase consisting of eight two-week sprints (a total of 16 weeks).

3.1.1 Stages and Sprints

The project will follow three main stages, each with clear deliverables and opportunities for iterative refinement. Sprints will drive progress during the Development Phase, while the overall structure ensures that each phase builds upon the previous one.

Preparation Phase
This phase focuses on defining project requirements, creating the project plan, developing personas, and producing any other necessary documentation before the development work begins

Development Phase
The core of the project will be developed during these 16 weeks, divided into eight two-week sprints. Each sprint will focus on building functional features, incorporating feedback, and refining the product iteratively

Wrap-Up Phase
The final 2-3 weeks will be dedicated to finishing any remaining documentation, implementing final feedback, and preparing the project for delivery or presentation.

3.1.1.1 Sprints within the Development Phase

The first sprint will be used to set up the development environment and create the necessary boilerplate to establish the foundation for the project. Subsequent sprints will be used to develop the application incrementally, with each sprint focusing on a specific set of user stories or tasks from the backlog

By organizing work into sprints, progress is continuously reviewed and adapted, allowing for early detection of issues and ongoing improvements throughout the project.

3.1.2 Sprint Reviews and Feedback

At the end of each two-week sprint, a sprint review will be conducted to evaluate progress, showcase completed work and gather feedback from stakeholders. These reviews will be used for ongoing adjustments to the backlog and sprint planning, ensuring that the project stays on track and meets expectations.

Deliverable Reviews
Each sprint review will present a summary of the user stories or features that were completed during the sprint, along with any feedback received and how it will influence upcoming work.

Incorporating Feedback
Feedback from sprint reviews will be integrated into future sprint planning or added to the backlog if it involves additional functionality or improvements.

Phase Reviews
After the completion of major phases (such as the Preparation or Development Phase), a comprehensive review will take place to reflect on the work done and ensure the project is ready to move forward. For example, after the Preparation Phase, the requirements, personas, and project plan will be reviewed to confirm readiness for development.

3.1.3 Why This Methodology Was Chosen

This methodology was selected because it combines the step-by-step structure of the Iterative Waterfall model with the continuous improvement and feedback loops of Scrum. The iterative nature ensures that development is responsive to evolving requirements, while the structured phases help maintain focus and direction.

By integrating Scrum practices (such as user stories, sprints, backlog management, and sprint reviews) the project will benefit from incremental progress and regular feedback without the overhead of team-specific practices like daily standups. The simplified review process, focused on sprint and phase reviews to ensure that progress remains visible, and stakeholders can stay involved without unnecessary formality.

Additionally, this methodology aligns with SURF's existing practices, making communication with stakeholders smoother and more effective.

3.2 Tools

3.2.1 Microsoft Teams

Microsoft Teams will be used for storing and managing all draft documentation for this project. This will make the management of files easy in a team environment while synchronized across devices from anywhere.

3.2.2 Codeberg

Codeberg, an open-source alternative to GitHub, will be used for version control, code reviews, project management, and managing CI/CD pipelines. Project management tasks will be handled through "Codeberg Projects", which enables the linking of user stories to issues, and then the linking of those issues to pull requests.

Once the documentation is finalized, the final versions will be published and maintained in the Codeberg repository.

3.2.3 Signal

A Signal group chat has been set up with all stakeholders (except those from Windesheim) to facilitate quick and easy communication. This is useful for rapid responses, discussions, and updates, to keep the project running smoothly.

3.3 Additional Approach Per Product

3.3.1 Initial project documentation

The initial project documentation will consist of multiple documents, each tailored to its intended audience and purpose:

University Assessors
I'll provide certain documents to my university assessors with a detailed understanding of the work being done at SURF.

SURF and Supervisors
Documents like this project plan will be reviewed by SURF and once approved, it will be shared with the supervising teacher to keep them informed.

Requirements Document
This document will be reviewed by both FileSender and SURF stakeholders to ensure the captured requirements meet their needs and expectations.

3.3.2 Software (MVP)

During development, we will utilize Git for version control, ensuring that any issues can be resolved easily by rolling back to previous versions if needed.

New features and updates will be introduced through pull requests. Each pull request will be reviewed by a SURF team member or one of the FileSender System Administrators with access to the repository, as many of them are also developers. The pull request process will include:

- Code verification using a linter within the CI/CD pipeline.
- Unit tests will run automatically within the CI/CD pipeline.
- If the CI/CD pipeline detects errors, the pull request will be blocked until issues are resolved.

3.3.3 CI/CD pipelines

The CI/CD pipelines will be added to the repository through pull requests, just like any other feature or update. The pipeline code will undergo peer review, and automated tests will run to ensure functionality. This approach maintains consistency and enforces that code quality remains standard across the project.

3.3.4 Final documentation

The final documentation will be designed to support the smooth transition of the project after my internship. It will be reviewed by FileSender System Administrators to ensure it is clear, comprehensive, and useful. This documentation will include:

- Development setup instructions
- Guidelines for continuing development
- Deployment and production installation steps
- Recommendations for next steps and future requirements

4 Project Management

4.1 Stakeholders

4.1.1 SURF

SURF serves as the primary stakeholder for this project as they placed significant importance on its successful completion.

4.1.2 System Administrators

The current system administrators of FileSender will act as secondary stakeholders. Their feedback, requirements, and preferences influence the project's decisions, therefore, it is important to account for their needs throughout development.

4.1.3 Windesheim

Windesheim plays a role as both an interface stakeholder (representing the university as an institution) and as a supervising teacher, who provides guidance and support. While Windesheim does not have direct influence over project outcomes, their involvement is essential.

4.2 Communication with Stakeholders

Communication with stakeholders is generally tailored to the nature of each stakeholder:

SURF and System Administrators
Communication is typically informal, primarily conducted through the Signal messaging app or face-to-face meetings.

Windesheim
Communication with Windesheim is generally more formal, as they have specific requirements and expectations.

4.3 Delivery

At the conclusion of my internship at SURF, a final presentation will be delivered to summarize the project's outcomes. This presentation will be given to the university assessors, SURF team members, and possibly some FileSender System Administrators. For stakeholders who cannot attend in person (such as those outside the Netherlands), a smaller remote presentation will be arranged.

The following project deliverables will be handed over to SURF via the Codeberg repository:

- Source code
- CI/CD pipelines
- The latest versions of all project documentation

4.4 Expectations from Stakeholders

Expectations from stakeholders are straightforward and manageable. During the initial preparation phase, regular communication through messaging and meetings will be maintained to align with the project's goals and requirements. Once the project progresses, this interaction will be streamlined to regular sprint review meetings.

In these sprint reviews, stakeholders will have the opportunity to request new features or suggest changes. If necessary, the project scope can be adjusted based on this feedback. Additionally, progress will be presented every two weeks as well as the plans for the upcoming two weeks.

As the project approaches completion during the wrap-up phase, a dedicated two-week period will be taken to finalize tasks, review documentation, and create any additional documentation as needed. During these final weeks, stakeholders from SURF and possibly System Administrators will be invited to a concluding presentation at Windesheim to showcase the project's outcomes.

Finally, in the last week of my internship, I will gather formal feedback from the stakeholders at SURF, with input from my supervisor to ensure the project's evaluation and closure are thorough.
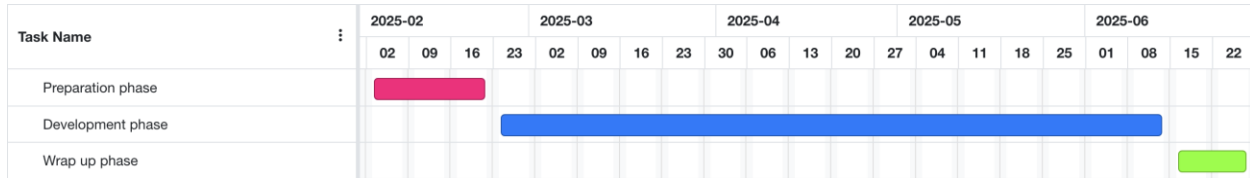
5 Planning

Task Name		2025-02				2025-03				2025-04				2025-05				2025-06			
		02	09	16	23	02	09	16	23	30	06	13	20	27	04	11	18	25	01	08	15	22
Preparation phase		██	██																			
Development phase					███	███	███	███	███	███	███	███	███	███	███	███	███	███			
Wrap up phase																				██		

Figure 1

5.1 Preparation Phase

The Preparation Phase is focused on defining the foundation of the project and outlining key requirements. Although this phase is allocated three weeks in the project plan, the goal is to complete it within two weeks, using the saved week for the Wrap-Up Phase.

During this phase, I will:

- Create essential documentation, including the project plan, requirements, and personas.
- Identify project objectives, milestones, and deliverables.
- Establish a backlog of user stories to guide development during the sprints.
- Set up the development environment and tools, ensuring a smooth transition into the Development Phase.

By the end of this phase, all necessary documentation will be reviewed to confirm readiness for development.

5.2 Development Phase

The Development Phase forms the core of the project and spans 16 weeks, divided into eight two-week sprints. This phase is dedicated to the iterative development of the MVP, ensuring continuous progress, refinement, and feedback.

During this phase, I will:

- Create and prioritize user stories within the product backlog based on project goals.
- Develop features incrementally, with each sprint delivering working functionality that can be reviewed and refined.
- Conduct regular sprint reviews at the end of each sprint to present progress, showcase completed features and gather feedback from stakeholders.
- Continuously refine the backlog to adapt to feedback and evolving requirements.

This phase ensures that progress is transparent and that feedback is incorporated throughout the project to achieve the final objectives efficiently.

5.3 Wrap-Up Phase

The Wrap-Up Phase is initially planned for two weeks, with a potential third week if the Preparation Phase is completed early. This phase will focus on finalizing all aspects of the project, ensuring it is ready for submission and presentation.

During this phase, I will:

- Finalize and submit any remaining project documentation, including reports and user manuals.
- Address any outstanding tasks or last-minute adjustments based on final feedback.
- Prepare for the final project handoff, including any necessary deliverables, demonstrations, or presentations.

This phase ensures that the project is polished and complete, with all required elements ready for evaluation and delivery.

5.4 Unworkable Hours

In addition to the planned phases, there are specific days when I will be unavailable for work. Below is a brief overview:

Date	Reason
5 Feb	Meeting at university
21-25 Apr	University peer-review week (one day off)

Table 1

During the week of April 21-25, I will still be working, but one day will be allocated for university-related peer-review meetings.

6 Personal Learning Objectives

Continuous growth is essential which is why I have set key goals and objectives that I will work towards throughout the project.

6.1 Keep Notes on Everything

6.1.1 What?

I will maintain detailed notes on everything I do during the project. This includes daily records of tasks completed, notes from meetings, decisions made, and any challenges or obstacles I encounter.

6.1.2 Why?

Maintaining thorough notes will allow me to:

- Reflect on my previous thought processes and decisions
- Identify areas for improvement and growth
- Track the progress of the project more effectively

By doing this, I can continuously learn and refine my approach to both problem-solving and project management.

6.1.3 How?

To achieve this, I have purchased a notebook and pen from the bookstore. I will carry this notebook with me at all times to ensure that I can record notes consistently. My focus will be on making note-taking a daily habit, ensuring no important details are overlooked.

6.2 Enhance Code Quality and Maintainability

6.2.1 What?

I'll follow best coding practices, write clean and readable code, and focus on maintainability by including comments, and documentation.

6.2.2 Why?

Maintaining clean and readable code reduces technical debt, makes debugging easier, and ensures that future developers or collaborators can understand my codebase without difficulty.

6.2.3 How?

I'll adhere to style guides by reviewing and refactoring code weekly and maintain proper version control using Git. I'll also perform self-code reviews before committing any major changes.

6.3 Collaborate and Seek Feedback

6.3.1 What?

I'll actively seek feedback from mentors, project supervisors, or peers on my design, code, and progress throughout the project.

6.3.2 Why?

Feedback will help me spot errors, validate design decisions, and gain new perspectives to improve the overall quality of my project.

6.3.3 How?

I'll schedule regular check-ins with my mentor and set up peer code reviews. I'll document feedback in my notebook (connected to Objective 6.1) and ensure I address the comments before the next review session.