Decibel Threshold Event Displayer

Scrum & Project Management Report

vonal3, gernd1, deged2

January 17, 2025

Contents

List of Tables

List of Figures

1 Introduction

1.1 Initial Situation

According to the Federal Office for the Environment (DE: BAFU) one in seven people in Switzerland is affected by noise pollution [1]. The pollution comes primarily from road traffic, followed by railways and then air traffic. In addition to these noise sources, construction sites, nightclubs and public facilities also produce noise. Because of this, Switzerland has set up upper noise limits that must be respected. However, it is of course the case that these limits are not always adhered to. Affected people must then either accept this or take action against it. For the latter, they must gather evidence to prove their noise disturbance to the police and the courts. This evidence then comes from an audio recording which the affected person made them self.

1.2 Product Goals

To help the people affected by noise pollution, we want to create an application which processes a given sound file (.WAV) and analyzes it. It should detect when a specified threshold has been exceeded and then summarizes the result in a PDF document. The document should then contain all necessary information for filing a complaint. As our application will have a wide range of end users, two of our main design goals are to make it as user-friendly as possible and to make it platform independent, so it can be used with any PC operating system and ideally mobile device.

1.3 Problems with Audio Files

A WAV file (.WAV) contains samples of the recorded audio, where each sample represents the amplitude at a given moment. Those amplitude values are relative to each other and not absolute. It is therefore impossible to determine the actual dB(A) (loudness relative to the human ear) someone would perceive without any further information [2–4]. Because of this, we require the user to also give information about the minimal and maximal dB(A) measured in the given audio file. To do this, we recommend using a smartphone app like DecibelX for IOS, which allows the user to record the audio and also conveniently inspect the minimal and maximal dB measured in that recording. With those two values, we can then map the relative values from the WAV file to its db(A) values.

2 Specification

The following chapter describes the system specification. The specification is derived from the requirements in the project description as well as the discussions with the stakeholder. Assumptions and constraints are described in the following sections and were validated by the defined product owner and the stakeholder.

2.1 System Delimitation

The system delimitation is split into the static system environment described in 2.1.1 System Environment and the dynamic process environment described in 2.1.2 Process Environment.

2.1.1 System Environment

The static system environment is split into the three contexts *System*, *System context*, and *Out of scope*. *System* includes the application as well as any software dependencies. The *System context* includes people or objects which have an influence on the application. *Out of scope* describes particularly what could have an influence on the application but has been deliberately excluded by the project team (see 2.2.2 Pre-Conditions and Boundaries).

System:

- Frontend (User interaction)
- LaTeX and pgfplots
- WAV file analyzer

System context:

- Stakeholder (tutor)
- User
- Noise producers
- Lärmliga
- Recording Device

Out of scope:

- Integration in legal complaints

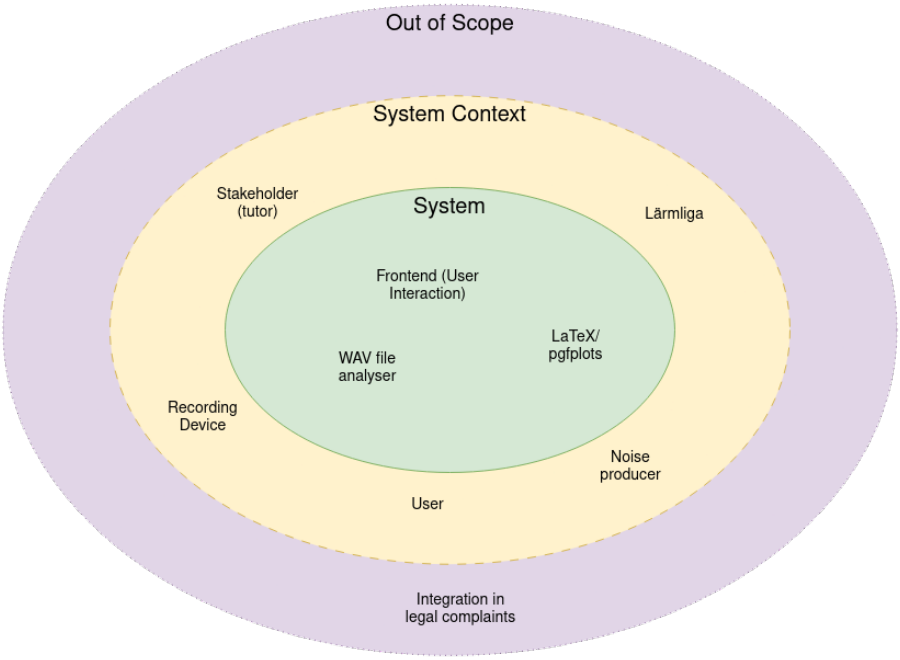


Figure 1: System Environment

2.1.2 Process Environment

As in the previous chapter 2.1.1 System Environment, the process environment is also split into *System*, *System context*, and *Out of scope*.

System:

- Selecting (uploading) a WAV file
- WAV file analysis (validation, parsing, conversion to absolute db values, threshold filtering)
- Plotting analysis result (LaTeX / pgfplots) and PDF generation

System context:

- Recording noise
- Producing noise

Out of scope:

- Device calibration (see 2.2.2 Pre-Conditions and Boundaries)

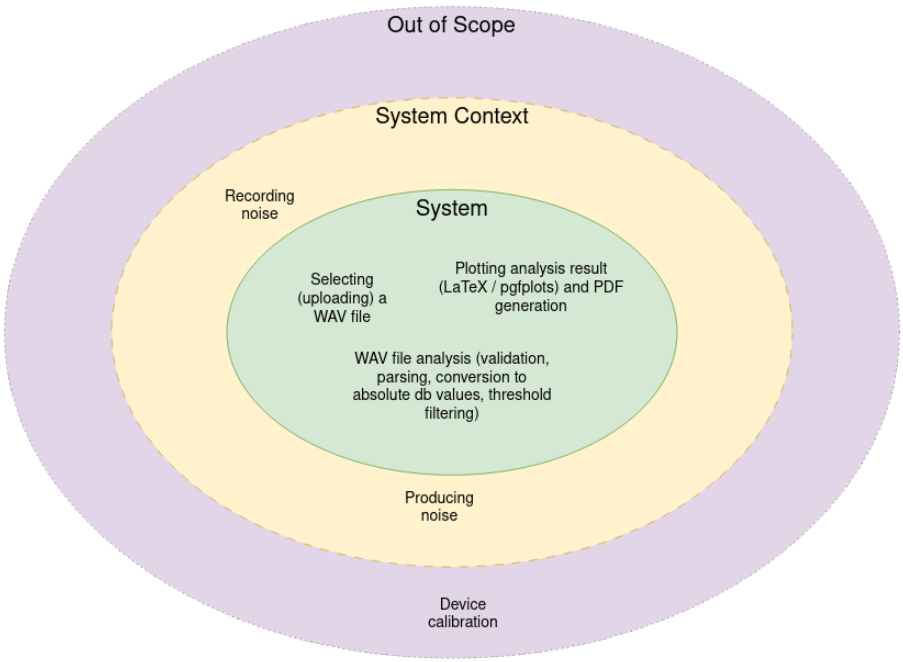


Figure 2: Process Environment

2.2 Requirements

In the following section the requirements are detailed. Also, the project boundaries and pre-conditions are described.

2.2.1 Functional requirements

The project team identified the following functional requirements:

ID	Requirement	Priority
R1	Allow users to upload a WAV audio file.	MUST
R2	Validate the uploaded file to ensure it is in WAV format, providing an error message if it is not.	MUST
R3	Analyze the uploaded WAV file and calculate the noise level in decibels (dB).	MUST
R4	Allow users to download the plotted noise data as an image or PDF.	MUST
R5	Allow users to input metadata for the audio file, such as location, date, and time.	MUST
R6	Plot the noise level data over time, with the x-axis representing time and the y-axis representing noise level (in dB).	MUST
R7	Generate a PDF report including the plotted noise data and user input metadata.	MUST
R8	Provide clear feedback and error messages.	SHOULD
R9	Intuitive and responsive UI for selecting files, configuring options, and viewing results.	SHOULD
R10	Allow the user to configure custom thresholds for noise levels.	COULD
R11	Allow the user to change the language of the application.	COULD

Table 1: Functional Requirements

2.2.2 Pre-Conditions and Boundaries

Research done by the project team has revealed that reading absolute decibel values from WAV files is more difficult than anticipated. The reason for this is that normal consumer microphones are not calibrated for scientifically accurate measurements [**?**]. Furthermore, it is trivial for a user to increase the volume of a WAV file using free, easy-to-use software like Audacity [**?**, **?**]. Therefore, the project team has decided on a tentative working hypothesis together with the

Stakeholder.

Pre-Conditions:

- The user must use a properly calibrated microphone or use a phone app which has built-in calibrations like Decibel X [**?**] [**?**].
- The user must have a way to export their audio recording as a WAV file.
- The user must have access to a web browser capable of running WebAssembly.
- The user must not edit the exported WAV file in any way except to decrease the length of the recording.

Boundaries:

- The application supports only the analysis of pre-calibrated, unedited WAV files.
- The application supports only single-channel WAV files.
- The application does not allow generating a legal complaint or directly integrating its output in a legal complaint.
- The user is presumed to be using properly calibrated equipment. There is no validation done in the application to ensure proper calibration.
- The user is presumed to be honest and to not have edited the WAV file. There is no validation done in the application to ensure the file has not been edited.

2.2.3 Privacy and Legal requirements

Because we are analyzing audio, privacy is an important point to consider. For analyzing the audio, we will only look at the amplitude of individual samples and take the rms over 300ms. No further analytics will be made. The get the rms, a number of individual samples are combined into one, which will then be plotted in the resulting document. Because of the rms, it is impossible to reconstruct the original audio from the plot. Additionally, we don't use any library to analyze the audio so we have full control over what's happening.

Furthermore, the application will run solely on the client and the server are only providing the HTML, CSS and JavaScript files, hence no data is sent back to the server. We also don't store any information from the user like IP addresses or the user agent. However, the deployment will use GitHub pages, meaning we don't have any control about the information GitHub stores about the user.

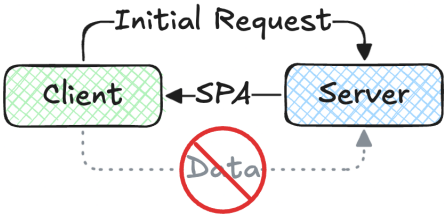


Figure 3: Privacy Illustrated

2.3 Usability

In order to best understand the targeted user base, the project team identified personas 2.3.1 Personas. Using this specification, the project team created prototypes for the user interface as well as the PDF report generated by the application 2.3.3 UX-Prototyping.

2.3.1 Personas

Noise is a problem almost every demographic can be affected by. Nonetheless, the project team has attempted to capture the most common attributes of people likely to use the application.

Name	Daniel
Age	42
Sex	Male
Occupation	Business consultant
Marital Status	Married & has two young children
Lifestyle	Daniel usually has to get up early in the morning because his customers are spread across the entire German-speaking part of Switzerland, and they expect him to meet them in their offices.
Goals	Daniel wants to expand his client base in hopes of possibly quitting at his workplace and founding his own consulting firm.
Frustrations	Daniel needs a lot of sleep which has been difficult ever since his first child was born. Much worse however, is that his new upstairs neighbors are up late every night, playing loud music, stomping across the floor, or arguing loudly. Daniel has already asked them to be more considerate but to no avail.
Expectations	Daniel expects to find a way to prove to the police that his neighbors are consistently breaking the law and interfering with his and his families' lives. He expects this solution to be free, easy to use, and reliable.

Table 2: Persona 1 (Daniel)

Name	Julia
Age	25
Sex	Female
Occupation	Student
Marital Status	Single
Lifestyle	During the day Julia is mostly attending lectures at university, running errands, or going after one of her hobbies. In the evenings she studies, often long into the night.
Goals	Julia wants to land a scholarship at a prestigious university, meaning she has to be top of her class.
Frustrations	Julia's apartment has windows facing a busy street, and she regularly faces difficulties focussing on her studies due to the noise. She is convinced her landlord has neglected to install the noise isolation required by law.
Expectations	Julia is looking for a way to make her landlord install better noise isolation to the apartment complex so she and her neighbours can spend their evenings without being bothered by the noise of passing cars and trucks.

Table 3: Persona 2 (Julia)

2.3.2 Process Model

The project team has created the following process model Figure 4, which is a high level illustration of how the application should work.

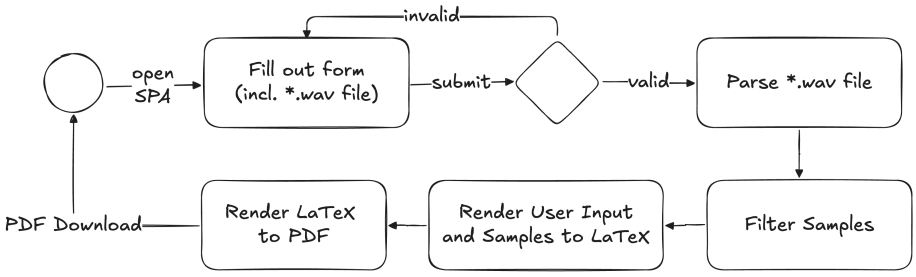


Figure 4: Process Model

2.3.3 UX-Prototyping

Based on the functional requirements 2.2.1 Functional requirements and the process model Figure 4, the project team identified two main UX components of the application and created their respective UX-Prototypes:

- UX Prototype – Website Figure 5, Figure 6
- UX Prototype – PDF Report Figure 7

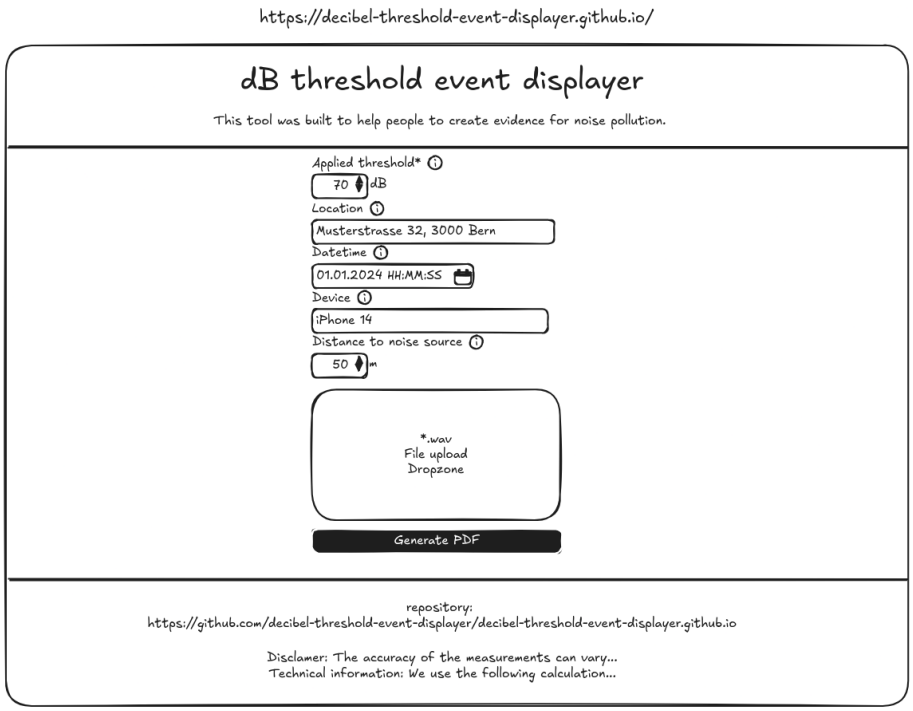


Figure 5: UX Prototype – Website

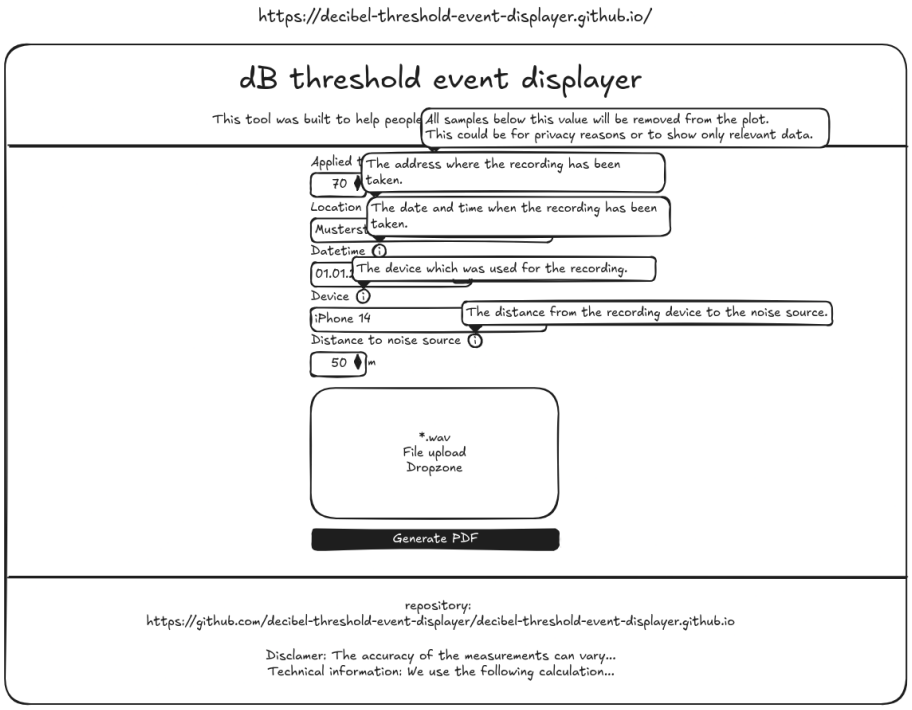


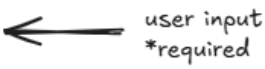
Figure 6: UX Prototype – Website with open tooltips

db_threshold_result_<timestamp>.pdf

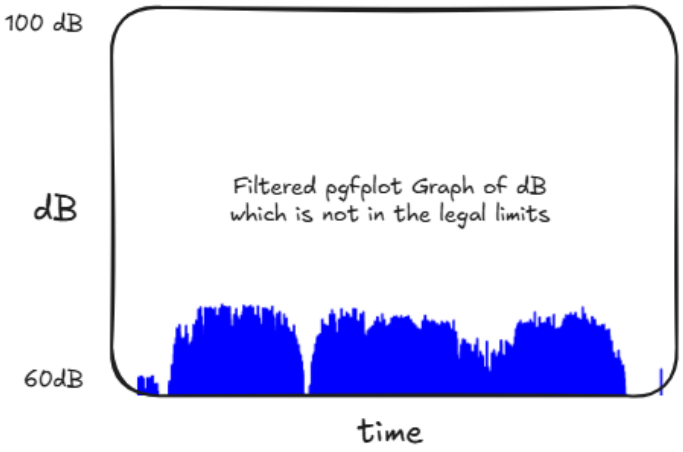
dB threshold result

Recording information

location: SIPBB
datetime: 00.00.2024 00:00:00
device: IPhone 14
distance to noise source: 20m
applied threshold: 60dB*



duration: 5min
duration over legal limit: 1min (20%)
peak: 70dB
averrage dB: 55dB



generation date: 00.00.2024
website:
https://decibel-threshold-event-displayer.github.io/
repository:
https://github.com/decibel-threshold-event-displayer/dec...

Disclamer: The accuracy of the measurements can vary...
Technical information: We use the following calculation...

Figure 7: UX Prototype – PDF Report

3 Scrum Roles

The project team has, in coordination with the tutor, determined that the association Lärmliga (`https://laermliga.ch/`) will not be directly involved in the project. Instead, the tutor Dr. Simon Kramer, will take on the role of the stakeholder and render the decisions to be made by the customer.
The other scrum roles were determined internally by the project team. At first, the decision was made for Dr. Simon Kramer to also be the Product Owner. However, the project team realized it would be impractical to include Dr. Simon Kramer in every product decision and especially in the prioritization of the backlog. The project team felt that, due to their relative inexperience, it would be best to have the Product Owner in the team, so decisions could be made quickly and validated with the stakeholders afterward, if necessary.

Name	**Role(s)**
Dr. Simon Kramer	Stakeholder, Tutor
Dominic Gernert	Product Owner, Developer
Lukas von Allmen	Scrum Master, Developer
Darius Degel	Developer

Table 4: Scrum Roles

4 Sprint Goals

The sprint goals are defined by the project team at the Sprint Planning right after the conclusion of the previous sprint. They are formulated to be compliant with the S.M.A.R.T principles. The project team decides the sprint goals based on the backlog prioritization and the issues assigned to the next sprint. The sprint goals are defined and reviewed in a Markdown file in the repository (Gitlab). If we did not achieve a sprint goal and not stated otherwise, it's implicitly assumed to be done in the following sprint.

4.1 Sprint 1: 10.10. - 23.10.2024

For the first sprint, the project team decided to focus on research and prototyping. This can be understood as a feasibility study. Due to the team's relative inexperience with LaTeX and due to the project scope, they felt that having a working prototype had to be made before a design decision could realistically be made.

Goal	Reached
Prototypes with two different technologies are implemented and their pros and cons are evaluated	Yes
Tech stack for the project has been chosen	Yes
License is defined	No
Git repository and documentation skeleton are created	No

Table 5: Sprint Goals of Sprint 1

In sprint 1, the project team managed to complete the first two goals (prototyping and tech stack evaluation). However, they were unable to evaluate the licensing because the decision about which tech stack should be used was made only at the end of the sprint. The project team underestimated the issue weights, leading to goals that were not reached.

4.2 Sprint 2: 24.10. - 06.11.2024

For the second sprint we already had to set the focus on the intermediate presentation, further we had to finalize the groundwork such as specifying Requirements, System delimitation, UX-Prototypes and how to process audio files. Although the project team defined more goals for the second sprint than the first, it is important to note that the total weight of tasks assigned to this sprint is equal to the first sprint and estimates are more conservative.

Goal	Reached
Intermediate presentation is prepared and presented	Yes
Requirements are specified	Yes
UX-Prototype is defined	Yes
System delimitation is specified	Yes
Decibel values can be calculated	Yes
License is defined	Yes
Git repository and documentation skeleton where created	Yes

Table 6: Sprint Goals of Sprint 2

In this sprint the whole team made a big effort to achieve all the goals and finished the most important groundwork in the project, and we have now a clear way forward to build the product. Even though the criteria for the intermediate presentation were available on a short notice and creating a presentation in

LaTeX turned out to be a bit challenging, the presentation was prepared in time and went well.

4.3 Sprint 3: 07.11. - 20.11.204

Based on the important groundwork in the last sprint, we could now start with the implementation of the application and the deployment and distribution setup. The amount of sprint goals is similar to the previous sprint, but we went with a bit less story points as we will also have to absolve the BFH project week 3.

Goal	**Reached**
Write documentation for interface	Yes
Read and parse *.WAV files correctly	Yes
Filter audio data correctly	No
Enable repository mirroring for distribution	Yes
Implement MVP frontend application	No

Table 7: Sprint Goals of Sprint 3

Because the BFH special week 3 took our focus entirely (we were in the same team), we could not complete all tasks in this sprint. Nonetheless, we were able to finalize the first implementation and deployment tasks. Getting started again with JavaScript was a bit challenging, but with the realization of a rudimentary JavaScript test environment we could set another important cornerstone for a well working application.

4.4 Sprint 4: 21.11. - 04.12.2024

The focus off this sprint is to progress on the implementation of the application. The project team only defined 3 goals, because they already contain more than enough story points. After this sprint we should be able to test each system component individually.

Goal	**Reached**
Filter audio data correctly	No
Implement MVP frontend application	Yes
Improve JavaScript test environment	No

Table 8: Sprint Goals of Sprint 4

We only completed one of the three sprint goals, because achieving the definition of done, especially completing the documentation, for each task often took more time than we initially estimated. However, we made massive progress and could implement and test all the system components individually.

4.5 Sprint 5: 05.12. - 18.12.2024

In sprint 5 the project team will be able to connect all the system components and test the application for the first time as a whole.

Goal	**Reached**
Finalize and document 'Filter audio data correctly'	No
Document 'Improve JavaScript test environment'	Yes
Create LaTeX template and fill in placeholders	No
Render LaTeX to PDF	No

Table 9: Sprint Goals of Sprint 5

On paper we achieved only one of four sprint goals as in the end the project team has not enough time for completing the documentation part of the tickets. As a consequence, we decided to add more weight to the tasks, even if we would estimate them lower. On the other hand we made great overall progress and could generate our first PDf plot via the webpage from an uploaded *.WAV file, which means we have a first version of the application where all system components are connected.

4.6 Sprint 6: 19.12.2024 - 01.01.2025

According to our plan this should be the last sprint and there is a lot of finalization, documentation and cleanup do be done on all different fronts. We decided to go with 5 sprint goals which is really ambitious, because afterward the project would be more or less done.

Goal	**Reached**
Finalize and document 'Filter audio data correctly'	No
Finalize LaTeX template and fill in placeholders	No
Finalize LaTeX to PDF	No
Finalize SPA	No
Define legal requirements	Yes

Table 10: Sprint Goals of Sprint 6

As we already knew, the sprint goals where really ambitious and more than half of this sprint is during the holiday season, so we actually completed only one of the five sprint goals. We still managed to get a lot of things done and even though we still have open tasks on all fronts, we are almost finished with the application.

4.7 Finalization: 01.01. - 15.01.2025

The project team decided that it does not make sense to start another sprint just for the sake of it, the leftover sprint goals and tasks will be done in a finalization round, which can be thought of an extension of sprint 6, as it was the last sprint anyway.

Goal	**Reached**
Finalize and document 'Filter audio data correctly'	Yes
Finalize LaTeX template and fill in placeholders	Yes
Finalize LaTeX to PDF	Yes
Finalize SPA	Yes
Final presentation is prepared and presented	Yes

Table 11: Finalization Goals

5 Requirements

5.1 Epics

Epics were defined at the very beginning of the project and are largely product focussed. Epics are defined in the Epics section of the Planning tab in Gitlab (Gitlab). Epics are subject to change but for the sake of completeness, they are included in this document anyway (see Figure 8).

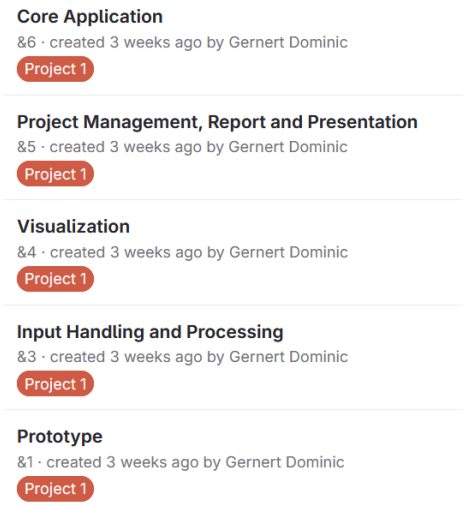


Figure 8: Epics

5.2 User Stories

User Stories are created as Issues in Gitlab. They are associated with an Epic and are prioritised. Issues contain a *Definition of Ready* (5.2.1 Definition of Ready), Acceptance Criteria, as well as a *Definition of Done* (5.2.2 Definition of Done). When an issue is selected for a sprint, sub-tasks are created, estimated and assigned. The issue itself is assigned to the person with the most assigned sub-tasks. The full list of issues is available in Gitlab's issue list or the Development Board.

5.2.1 Definition of Ready

The *Definition of Ready* is defined as a checklist on every User Story. In every Sprint Planning the issues selected for the next Sprint are validated by crossing off the checklist. All necessary adjustments are made together by the project team before the issue is estimated and planned. The *Definition of Ready* is part of the Issue Template in the repository (see Gitlab).

Definition of Ready:

- Requirements and Acceptance Criteria Defined
- Acceptance Criteria must be testable
- Understood by the Team
- Sized and estimated
- Prioritised in the Backlog

- No major Impediments

5.2.2 Definition of Done

Similarly to the *Definition of Ready* (5.2.1 Definition of Ready), the *Definition of Done* is a static checklist that is part of the issue template. The tasks are checked off by the reviewer, which is the Product Owner unless specified otherwise.

Definition of Done:

- Acceptance Criteria met
- Tested and no critical bugs
- Documentation updated
- Reviewed and approved

5.3 Product Backlog

Issues are prioritised first and foremost by the Product Owner. This prioritisation is implemented as tags in Gitlab and is can be viewed on the Development Board (see Figure 9)

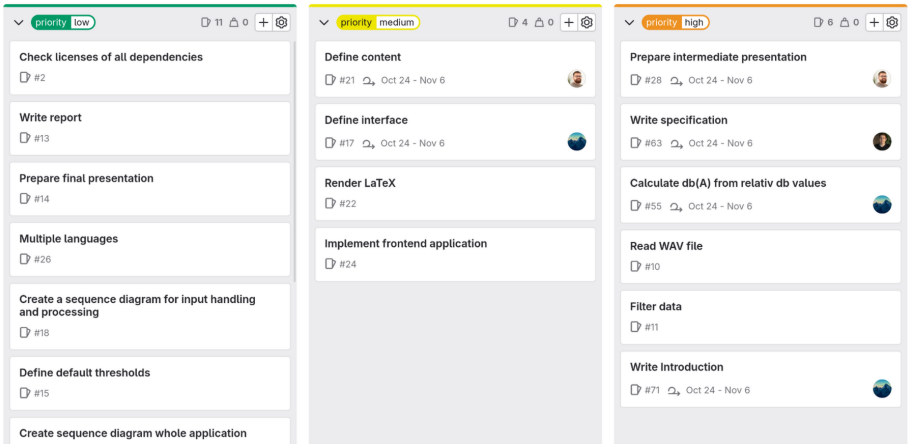


Figure 9: Product Backlog

5.4 Sprint Backlog

The Sprint Backlog of the currently running sprint is displayed as a column on the Development Board. In combination with the prioritisation (see 5.3 Product

Backlog), this makes for a convenient way for the project team to select issues for the next sprint based on priority (see Figure 10).

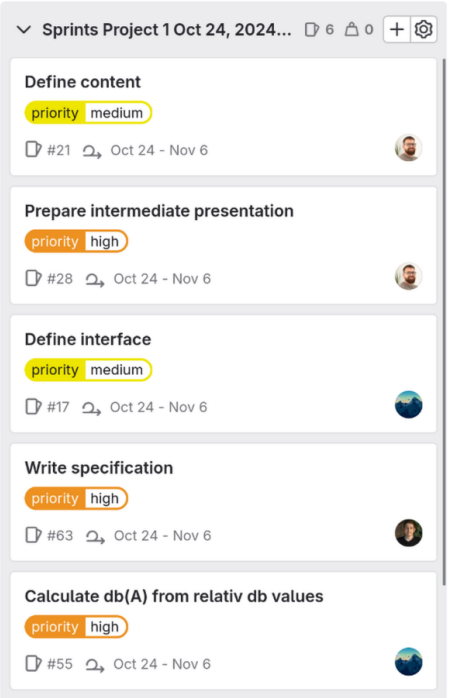


Figure 10: Sprint Backlog

5.5 Impediment Backlog

Impediments are also created as Issues in Gitlab. They are however, assigned the label *Impediment* and are displayed on a separate Impediment Board (see Figure 11 and Gitlab). Impediments are created either ad hoc or during the Daily Standup meeting. There is a template for Impediments analogous to the Issue Template (see Gitlab).

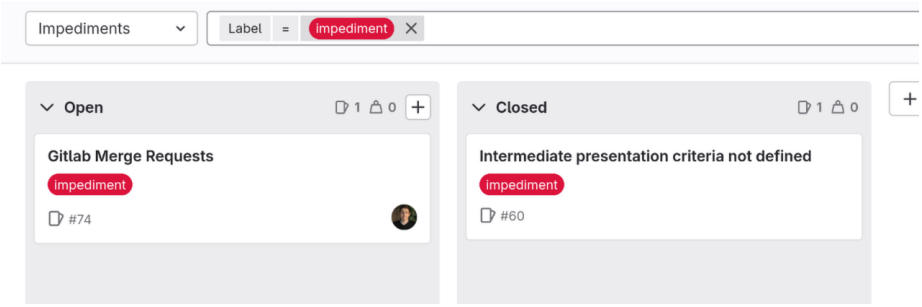


Figure 11: Impediment Board

6 Scrum Adaptations

6.1 Gitlab

Mr. Frank Helbling was added to the GitLab "Decibel Threshold Event Displayer" group of our project as a reporter (Gitlab Group). He has access to inspect Epics on group level. Issues and Issue Boards can be accessed on project level. The Daily Journals and Sprint Notes can be accessed in the doc/scrum folder (Scrum Notes).

6.2 Product Owner

As described in section 3, the Product Owner role has gone from our tutor, Dr. Simon Kramer, to Dominic Gernert. Although not specifically mentioned in the Scrum Guide [5], in the experience of the project team and an article on applied frameworks [6], the Product Owner is usually someone with a business background and not a developer. This makes sense, because they need to be able to make decisions about the product that only a business expert could make. Since this is not the real world but a school project, the project team had to reduce the coordination effort. Thus, the team decided to make Dominic Gernert the Product Owner. Since he is not a business expert for our project, decisions are still made as a collective. If however, there is a disagreement on a decision, the Product Owner's suggestion is followed. For bigger decisions, the Stakeholder Dr. Simon Kramer is involved.

6.3 Daily Scrum

As suggested by Mr. Frank Helbling in the Scrum presentation [7], the project team has decided to confine their Daily Scrum to 15 minutes. They are scheduled weekly on Wednesday at 18:15. The Scrum meetings that took place until the completion date of this document took about 30 minutes, however. The

likely reason for this is that the project team meets only weekly, but each person actually works on the project at multiple days of the week. This leads to more information, questions, and impediments being generated and having to be discussed during the Daily Scrum. The project team is still trying to keep their Daily Scrum short (less than 20 minutes) in the future.

6.4 Release Plan

The project team has decided against implementing a Release Plan. Mainly because it was never explicitly demanded by the tutor, Dr. Simon Kramer. The other reason for forgoing the Release Plan is that there is no real customer awaiting product updates and Dr. Kramer is capable of running the application himself should he want to do so. Product demonstrations by the team are not on a fixed scheduled and are done if necessary or requested by the stakeholder.

6.5 Retro

Instead of following the Keep-Try-Drop model for the Sprint Retrospective, the project team decided to focus on successes, problems, and improvements. A reason for that is the focus on the individual by the team due to the low number of members. This focus on successes and problems provides a way for each team member to explain their frustrations and discuss possible improvements. This adaptation is comparatively small and mostly one of wording. The project team feared however, that using the Keep-Try-Drop model, they would have to justify using certain tools when running into minor problems. Instead, the team hopes to focus more on the product than the tool chains.

7 Project Setup Review

7.1 Identifying the initial situation

For identifying the initial situation we analyzed 12.1 Project description and used multiple online resources such as the websites of Lärmliga Schweiz and Federal Office for the Environment FOEN. Based on this information, we were able to derive some of the requirements and specifications for our application, which now provides a convenient way for affected people to document noise pollution.

7.2 Topic analysis

To solve the issue at hand, one needs knowledge about how audio recording and measurement works. We invested the time needed to analyze the topic and gather the know-how needed to process and analyze audio files. A short summary of the problematic can be found under 1.3 Problems with Audio Files. This analysis provided the technical basis for achieving the product goal.

7.3 Stakeholder / Stakeholder Management

Initially we identified a list of possible stakeholders:

- Tutor: Dr. Simon Kramer
- People affected by noise pollution
- Noise producers (Construction Sites, Night Clubs, Highways)
- Lärmliga Schweiz
- Federal Office for the Environment FOEN

For the context of this module we decided to only consider our tutor as the sole stakeholder, because otherwise the stakeholder management would cost much more time without much benefit. This way the stakeholder management turned out to be straight forward, as Dr. Kramer provided his expectations and the scope of the project beforehand, only impediments and key decisions had to be discussed together.

7.4 Organization

The project organization and how the project team implemented scrum is documented under the following three chapters:

- 3 Scrum Roles: The distribution of the Scrum Roles has been proven practicable, even though some of the project members didn't have any experience in their respective roles
- 6 Scrum Adaptations: The project team was really happy about the way we adapted Scrum in this project as it was really product focused and did not introduce too much overhead
- 5 Requirements: We found the definition of ready and the definition of done were especially helpful, as it made sure the tickets had a minimum quality, a clear achievable goal and are well understood by the team.

Further the project team mainly worked together remotely which reduced the unnecessary over head of commuting.

7.5 Installations

The project team used the following tools for carrying out project 1:

- Project Management: GitLab
- Version Control (Code and Documentation): GitLab
- Documentation and Presentation: LaTeX

- E-Mail: MS Outlook
- Team communication (Chat and Video Calls): MS Teams
- Visualizations: Excalidraw
- Diagrams: draw.io

8 Review

8.1 Product goals

The following list summarizes the 1.2 Product Goals:

1. Analyze Audio File
2. Summarize findings in a PDF
3. Easy to use

From our point of view, the first two product goals are clearly achieved, as we built a working application which can parse and analyze audio files and then summarizes the findings in a PDF document. The third goal of 'easy to use' is not as straightforward to measure, but we argue that this is achieved by considering the following list of best practices:

1. Cross-platform and no download/installation required: Achieved by building a public web application
2. No login or personal information required
3. Clarity, consistency, responsiveness and familiar patterns: Achieved by using Bootstrap, from Twitter
4. Help and documentation: Achieved with integrating tooltips where needed and show human-readable error messages
5. Performance and fast load times: Achieved by realizing the application as a minimal SPA and preloading all necessary LaTeX resources in the background while the user is filling out the form

8.2 Sprint goals

The sprint goals were defined in their respective sprints and are documented under 4 Sprint Goals. We often did not fully achieve the defined sprint goals in the respective sprint because we underestimated the amount of work going into solving a ticket, especially the effort for the documentation. So even though on paper it looks like we did not reach our goals, we made good progress in each sprint and finalized the goal in the followup sprint. Keep in mind that the sprint goals were achieved implicitly and are not mentioned again in the followup sprint. In other words, we still achieved all sprint goals, just not in the sprint they were defined.

8.3 Product delimitation

The product delimitation is split into 2.1.1 System Environment and 2.1.2 Process Environment. With the help of these two delimitations, we were able to clearly differentiate what is relevant for the product and what is out of scope.

8.4 Deliverables

All deliverables were specified in their respective tickets and according to our definition of done they must be documented in the project report. We do not list all deliverables here, as they can be viewed as part of the projects report: Project Report. Our main deliverable is the final application, which is available online under the following link: dB threshold event displayer.

8.5 Product backlog / Sprint backlog

The product and sprint backlogs are documented under 5.3 Product Backlog and 5.4 Sprint Backlog respectively. The project team was happy to use them as defined. Even though there are still tickets left in the product backlog, we have no open tickets with the priority medium or high. This means we achieved a minimal viable product within the module's time frame.

9 Retrospective

9.1 Retrospective I: Scrum roles and stakeholders

- Product Owner: Dominic already has quite some experience with Scrum and thus made a perfect job from the beginning.
- Scrum Master: Even though Lukas held the role for the first time he could fulfill it after a bit of initial support from Dominic.
- Developers: Thanks to the clear definition of ready and definition of done, the developers always knew what to do in every task.
- Stakeholder: The project team enjoyed working with Dr. Kramer as the expectations were communicated beforehand and clearly.

9.2 Retrospective II: Scrum Events and Artifacts

- Product Backlog: The Product Owner did a perfect job with specifying and managing the product backlog. This helped the team to progress without any blockers.
- Sprint planning meeting: The planning was always really smooth, as it was clear which the logical next steps are.

- Sprint backlog: The Sprint backlogs were always a bit overambitious, as we almost never achieved all the sprint goals.
- Daily Scrum meetings: The 'Dailies' were actually held once a week, nevertheless they helped us to resolve issues on the current tasks and kept all team members in sync.
- Burndown Charts: As we did not work on the project every day, the burndown charts looked not like a gradual line downwards but instead showed a clear trend of the team doing work on weekends and within the time slot of the Proejct 1 module.
- Finished work: As we had a clear definition of done, the finished work was always documented in the projects report.
- Sprint Review: The project team really enjoyed the sprint reviews, as it made the progress visible.
- Sprint Retrospective: At the start of the project, we decided to use a minimal approach for the retrospective and the project team was really happy with this decision.

9.3 Retrospective III: Tools/Instruments

- GitLab: The project team was not always happy with the workflows provided by GitLab, such as the issue boards and the differentiation between group and project.
- MS Teams: This tool is an amazing, unified communication channel for working together, but the Linux support could be better.
- MS Outlook: MS Outlook just works as expected.
- LaTeX: Even though we did not have a lot of experience with LaTeX, we decided to use it for all documentation and presentation purposes. The project team enjoyed working with LaTeX, but for non-academic contexts we would have chosen less complex tools such as MS Word and MS PowerPoint.
- Excalidraw: This easy to use online tool helped us to create a lot of the diagrams and UI-Prototypes.
- draw.io: Another easy to use online tool for creating diagrams where the results look a bit more old-school / professional.
- Controlling:
  - Burndown Chart: The Burndown chart was a good visual indicator to see how the sprints progress is going, but as we often underestimated tasks it could be quite misleading for an external observer.
  - Product Backlog: Thanks to the detailed initial product backlog we could always see where we stood in the project.

10 Lessons learned

10.1 Insights into framework conditions

For the team members which already had quite some experience with Scrum, there was still the learning curve of setting up everything from scratch. The other member had quite some experience with alternative forms of agile software development and could learn a lot about the Scrum specific methods. In our point of view, the most important Scrum ceremonies were the sprint review and planning which we always held subsequently. The sprint review resulted in some interesting demos, kept the whole team in sync, showed the current progress, and already indicated what the next steps could be. On the other hand there is no sprint without the sprint planning and after it, every team member knows what he has to do next. In the context of project 1 the key takeaway for the project team is, that Scrum also works when nobody works full-time on the project and all the work is done remotely.

10.2 Cooperation

Team: The key challenge for the project team in terms of cooperation was the limbo between working, other modules, and project 1. This resulted in the reality that we only worked together in the modules official time block. In the beginning a project member had issues being on time for meetings, but luckily he could resolve this issue. Besides the minor challenges and issues, the team work was amazing, and we enjoyed working together.

Specialist advisor / Stakeholder: We consider the cooperation with Dr. Kramer as good, as he communicated his expectations clearly from the start. If we had issues or questions we received an answer within reasonable time or could schedule a meeting with him in the modules official time block.

Project Management Coach: The cooperation needed with the project management coach was minimal as all requirements were clearly documented in moodle. When something was unclear we received an answer within reasonable time.

11 Bibliography

References

[1] Federal Office for the Environment FOEN. Noise and vibrations: In brief. `https://www.bafu.admin.ch/bafu/en/home/topics/noise/in-brief.html`. Accessed: 2024-11-4.

[2] how to know the real-world db level of a file? https://community.adobe.com/t5/audition-discussions/how-to-know-the-real-world-db-level-of-a-file/m-p/13939424. Accessed: 2024-11-4.

[3] How can i calculate audio db level? https://stackoverflow.com/questions/2445756/how-can-i-calculate-audio-db-level. Accessed: 2024-11-4.

[4] Extracting sound pressure from wav file. https://dsp.stackexchange.com/questions/35250/extracting-sound-pressure-from-wav-file. Accessed: 2024-11-4.

[5] Ken Schwaber and Jeff Sutherland. The 2020 scrum guide. https://scrumguides.org/scrum-guide.html. Accessed: 2024-11-03.

[6] Carlton Nettleton. Is the product owner part of business or part of it? https://appliedframeworks.com/blog/is-the-product-owner-part-of-business-or-part-of-it. Accessed: 2024-11-03.

[7] Frank Helbling. BTI3031 - Intro Scrum (part 3). https://moodle.bfh.ch/pluginfile.php/3229204/mod_resource/content/11/BTI3031%20-%20Intro%20Scrum%20%28part%203%29.pdf. Accessed: 2024-11-03, authentication required.

12 Appendix

12.1 Project description

The **goal (what)** of this project is to deliver a FLOSS-licensed, platform-independent piece of software (computer program), called the *Decibel Threshold Event Displayer*, that

1. takes as inputs a WAV-file and a list of sound level thresholds in decibels (e.g., legal day and nighttime noise maxima above which your health deteriorates);

2. filters out all data points in the file that correspond to sound events below the lowest of the above thresholds; and

3. displays the remaining data points (as a blue vertical comb plot) on a horizontal time axis (with the dates and times corresponding to the data points) as well as the thresholds (as horizontal red lines) in decibel, and statistically summarises the data set with the help of the LaTeX-package pgfplots.

The **purpose (why)** of this project is to empower poor folks who suffer from insomnia due to ambient noise (`https://laermliga.ch/`) by arming them with the (peaceful) means of proving their noise hell (a smartphone app such as `https://apps.apple.com/ch/app/dezibel-x-pro-l%C3%A4rm-messger%C3%A4t/id1257651611` together with your software) to the police and the courts of law.

The code should be minimal, modular, and self-explaining.

The project report should be concise (maximally informative, minimally long). It must contain this project description as a quotation.

12.2 Declaration of Authorship

The project team, namely Dominic Gernert, Lukas von Allmen, and Darius Degel, hereby declare that the report submitted is our own unaided work. All direct or indirect sources used are acknowledged as references.