Audio effects library
Project schedule and communication plan

Kacper Harezga, Ewa Kobiela, Jan Laskowski
Krzysztof Sobczyk, Grzegorz Machura

24.04.2021

Contents

1 Plan schedule management **2**

2 Define activities **2**

3 Sequence activities **5**

4 Estimate activity durations **6**

5 Develop schedule **6**

6 Communication plan **7**

1 Plan schedule management

The process of developing the project schedule consists of defining the activities that will be needed to fulfill the project scope, estimate duration of specific tasks and prepare a schedule based on these information.

The rocess of schedule creation and management is presented in this document. All files and work used in the process is mentioned in the document or attatched as attatchement.

2 Define activities

In order to produce project deliverables, specific work packets needs to be defined. Below we will recall work breakdown structure document followed by the dictionaty which explains the scope of each work package. Using the WBS document we define activities that will be performed during the process of project creation.

The five main segments that needs to be fulfilled include:

- research - collecting information and learning about digital signal processing in Python. At this stage we prepare for implementing the effects
- time-domain effects and frequency domain effects - this is the main stage of the project. During this part we implement various types of filters and create the library of audio effects
- creating signal path - in order to present the usage of the library, we create a full signal path, which allows user to add chosen effect to input audio signal
- testing - finally, the library and signal path application will be tested to prove it can be used by the user

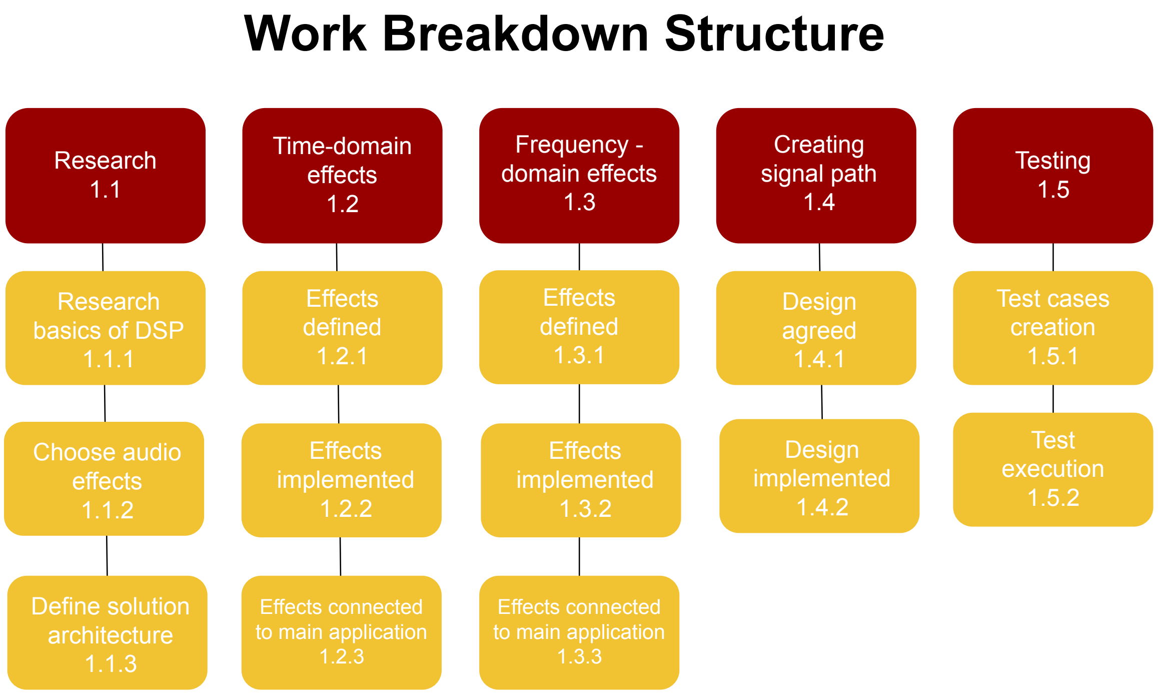
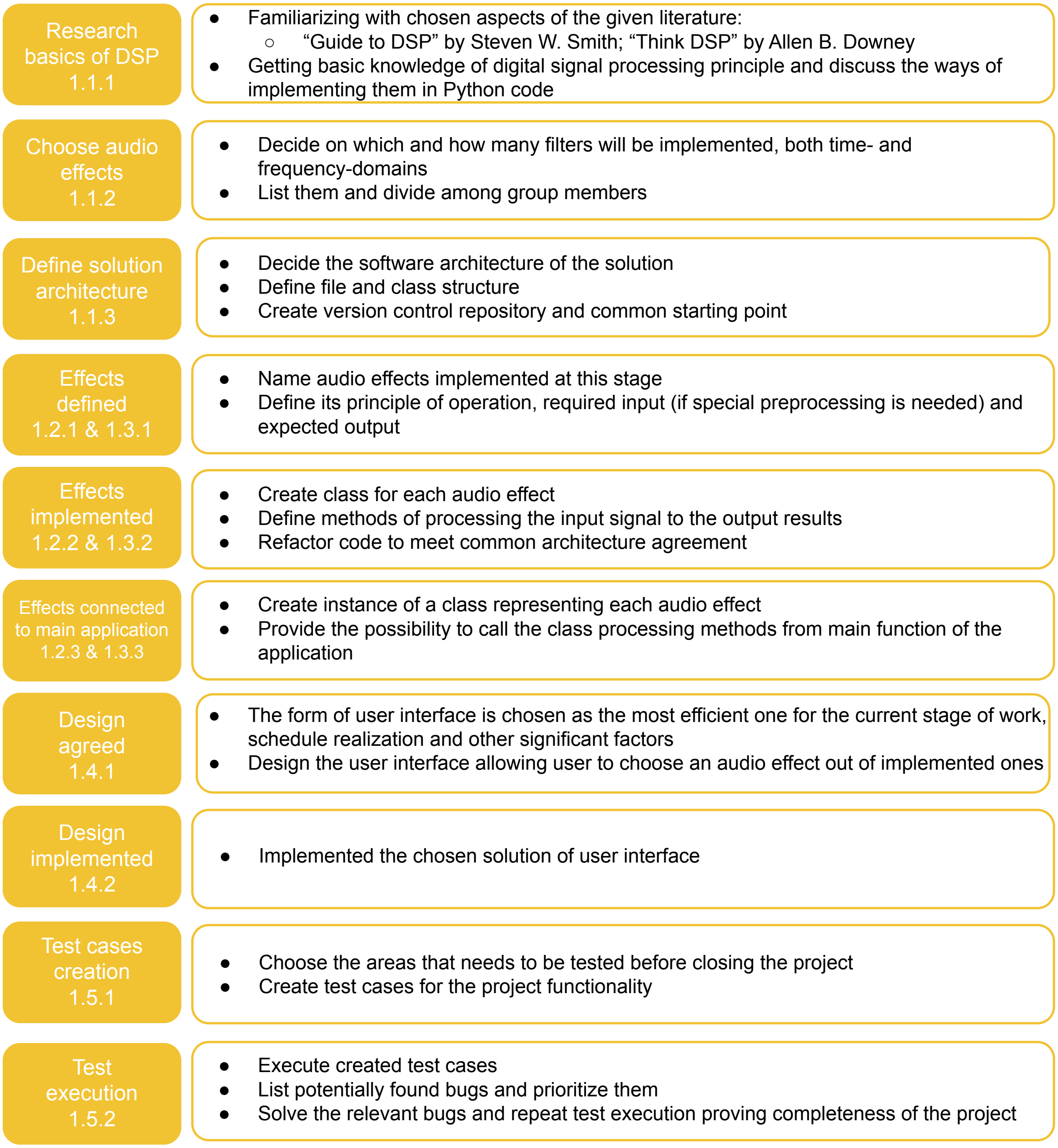

Work Breakdown Structure
Research 1.1
Research basics of DSP 1.1.1
Choose audio effects 1.1.2
Define solution architecture 1.1.3
Time-domain effects 1.2
Effects defined 1.2.1
Effects implemented 1.2.2
Effects connected to main application 1.2.3
Frequency - domain effects 1.3
Effects defined 1.3.1
Effects implemented 1.3.2
Effects connected to main application 1.3.3
Creating signal path 1.4
Design agreed 1.4.1
Design implemented 1.4.2
Testing 1.5
Test cases creation 1.5.1
Test execution 1.5.2

Figure 1: Work breakdown structure



Research basics of DSP 1.1.1	● Familiarizing with chosen aspects of the given literature: ○ “Guide to DSP” by Steven W. Smith; “Think DSP” by Allen B. Downey ● Getting basic knowledge of digital signal processing principle and discuss the ways of implementing them in Python code
Choose audio effects 1.1.2	● Decide on which and how many filters will be implemented, both time- and frequency-domains ● List them and divide among group members
Define solution architecture 1.1.3	● Decide the software architecture of the solution ● Define file and class structure ● Create version control repository and common starting point
Effects defined 1.2.1 & 1.3.1	● Name audio effects implemented at this stage ● Define its principle of operation, required input (if special preprocessing is needed) and expected output
Effects implemented 1.2.2 & 1.3.2	● Create class for each audio effect ● Define methods of processing the input signal to the output results ● Refactor code to meet common architecture agreement
Effects connected to main application 1.2.3 & 1.3.3	● Create instance of a class representing each audio effect ● Provide the possibility to call the class processing methods from main function of the application
Design agreed 1.4.1	● The form of user interface is chosen as the most efficient one for the current stage of work, schedule realization and other significant factors ● Design the user interface allowing user to choose an audio effect out of implemented ones
Design implemented 1.4.2	● Implemented the chosen solution of user interface
Test cases creation 1.5.1	● Choose the areas that needs to be tested before closing the project ● Create test cases for the project functionality
Test execution 1.5.2	● Execute created test cases ● List potentially found bugs and prioritize them ● Solve the relevant bugs and repeat test execution proving completeness of the project

Figure 2: Work breakdown structure dictionary

3 Sequence activities

The sequence activities diagram below represents relations and interaction between project objects in sequential order. This diagram consists of four objects: the user, directory, libraty and methods of the filter class.

Firstly, the user needs to save the input file into predefined directory. Then they will choose in library an audio effect to be applied. Having these presonditions, the library reads input file from the directory and calls processing node function frm choosen audio effect class. In the class file the effect is added to the base signal and the class returns .wav file. The file is written into the directory by the library. Finally, user can collect the processed file.

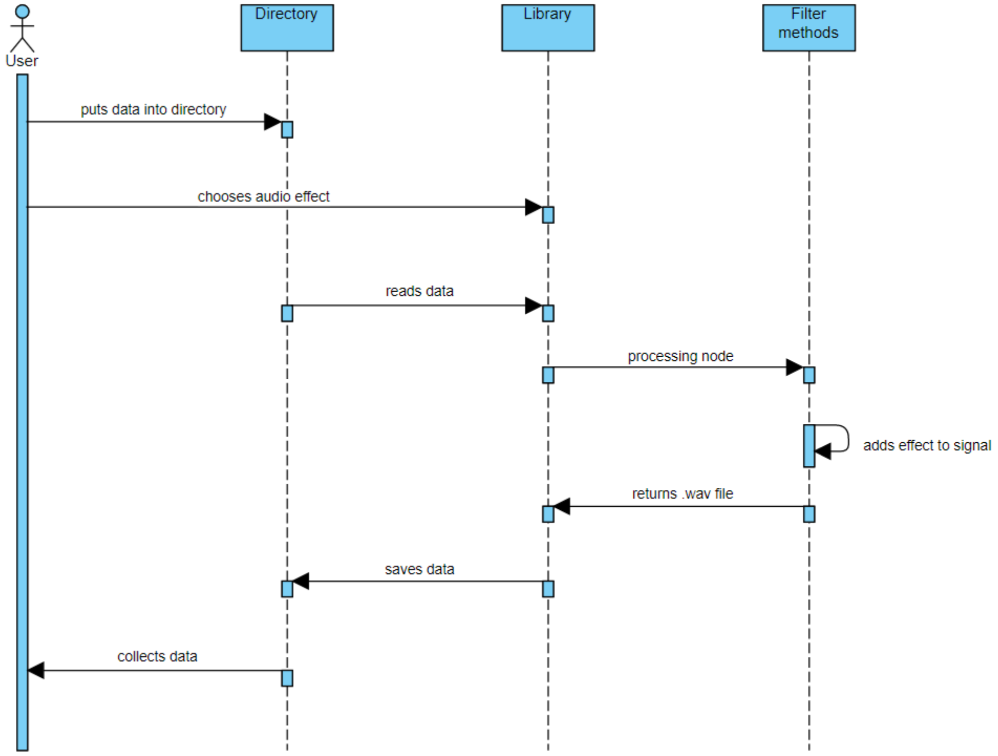


Figure 3: Sequence activities diagram

4 Estimate activity durations

The process of estimating schedule of the project is based on work patkets defined in work breakdown structure documment, which we also call tasks or activities. Using defined activities we predict how much time will they consume:

- Research basics of DSP - 2 weeks
- Choose audio effects - 1 week
- Define solution architecture - 2 weeks
- Define effects (both time- and frequency-domain) - 1 week per each
- Implement effects (both time- and frequency-domain) - 2 weeks per each
- Connect effects classes to common signal path (both time- and frequency-domain) - 1 week per each
- Agree on design of user interface - 2 weeks
- Implement user interface - 2 weeks
- Create test cases - 3 days
- Execute test cases - 3 days

5 Develop schedule

According the the estimated duration of activities the schedule was developed. We tried to execute tasks one by one, however due to the fact that the project should be finished in 14 weeks and total estimated time for all tasks is equal 18 weeks, we allowed some overlapping. Thus we decided that time and frequency-domain filters will be implemented in parallel by different members and stakeholders will be notified when approximately half of them are ready. We will also inform Dolby company after finishing each of five segments.

The Gant chart with activities scheduled in weekly intervals is presented below. Applying the weekly approach also allows us to plan weekly meetings in which we can discuss progress from previous week - as each week either one work package will be finished or halfway done.

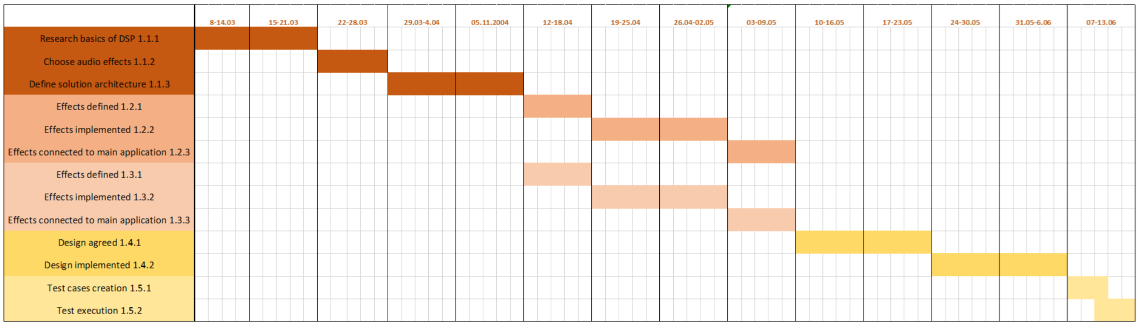


Figure 4: Gant chart

6 Communication plan

The communication between the project team and stakeholders will be conducted via e-meetings. The Dolby company proposed their own BlueJeans communication tool which we will use to present progress of the project and receive feedback. In the communication with the university party we will be using Microsoft Teams. For internal communication we will be using instant messaging in Facebook Messenger and the meetings will be conducted in Discord application.

Language using in internal communication and communication with Dolby is Polish, for both official and unofficial docummentation. However, the project docummentation must be written in English due to the language in which the university course is conducted. However the informal communication may be handled in Polish.

We do not specify any dates or frequency of meetings with both primary stakeholders. The plan assumes that the project members will reach a meeting request and the project manager will contact the stakeholder via email message and establish the meeting time. During the meeting we will discuss improvements or progress, ask for further information or explaining of technical matters or rise questions.

The plan also assumes frequent team meetings, occuring each weekend in the form of catch-up on individual progresses and report any problems or blockers occuring up to date.