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