Project 4 Maintenance Plan

Perfective

Our program would ideally search through the whole of a playlist, establish a core of songs representative of the overall content of the playlist, and remove whatever songs do not belong in that core. We may continue to tweak the program in order to ensure the criteria of songs removed matches the range indicated by the slider, or that the criteria that the songs are being compared with is representative of the bulk of the playlist. We may consider new ways to analyze the songs to improve how tracks are compared.

The program uses packages such as Python and node.js that required us to download them off the internet in order to run the program on our own machines. To that end, we plan on running the program on a server which would run the packages alone and provide access to the user. That way, the user would not have to run the packages and the program in the command line but simply navigate to the website.

Adaptive

As time progresses, new musical trends will develop, a new generation of musicians will become popular, and albums will continue to be uploaded to Spotify. Our algorithm may be unable to compare between newer trends and older trends, making it harder to exempt certain songs from the playlist. To this end, we may need to modify our algorithm so that it can anticipate newer trends and be able to classify playlists with regards to them.

The system is web-based so there should be little difficulty running the program between different browsers like Firefox or Chrome and other operating systems like MacOS or Linux.

Corrective

If our project were to be implemented, we would likely add an option to allow the user to provide feedback. These would provide us with suggestions to improving the program, updating our algorithm, and fixing any issues the user encounters. Primarily, these will inform us of issues running the program that require attention.

Preventative

As our deployment plan noted, our software can work with other APIs, not only with Spotify's. The possibility to implement other APIs suggests we could redesign the program to make it easier to replace the Spotify APIs with other APIs. We would consider how the program would work with other APIs and modify the code to address possible issues with implementing other API functions.

We currently have documentation for both the back end and front end, making it easier to update when needed.

Costs

As mentioned above, we would need a server to handle the packages needed. Our most likely option is to use an Amazon server, which would allow the user to access the program easily and not

have to run it on their own computer. Using this kind of server would be inexpensive, costing less than half a dollar per hour and varying by which packages we use.

APIs tend to be expensive to develop, so there is some cost to licensing them from Spotify or the other companies we choose. When choosing other APIs, we can consider industry trends and select those that we see as profitable.