Final Report

COSC 4P02 – Software Engineering 2 (Winter 2024)

Professor- Naser Ezzati - Jivan

April 28th, 2024

Project Title: Social Media Post Generator

GitHub: https://github.com/mc16dn/COSC-4P02-Group-Assignment/tree/main

Jira: https://brocku-hh18iq.atlassian.net/jira/software/projects/SMPG/boards/1/backlog

Project Members

Name	Role	ID	Brock Email	GitHub Username
Basel Abdel Hafez	Product Owner	6826853	ba19tw@brocku.ca	Hafezberg
Hevar Halabjaee	Scrum Master	6701248	hh18iq@brocku.ca	spiralwind
Rifat Chowdhury	Project Manager	6830301	rc19nv@brocku.ca	Rifat-Chowdhury
Raymond Dong	Developer 1	6592539	rd18sx@brocku.ca	raymonddong2
Matthew Cam	Developer 2	6195556	mc16dn@brocku.ca	mc16dn
Maisam Anjum	Developer 3	6804298	ma19an@brocku.ca	Koorikdat

Contents

Team Member Contributions

Basel Abdel Hafez:

- Created user epic.
- Contributed to user stories.
- Provided the vision behind the end-goal of the program.
- Contributed to the Release Planning doc editing.
- Keeping track of the meetings by meeting logs.
- Encourages team communication by being vocal in meetings.
- Generated many ideas to better the program after extensive research.

- Provided a plethora of permutation options.
- Contributed the bulk of the writing and decision-making behind what's included in every progress report.
- Created the presentation with Hevar and Rifat.

Hevar Halabjaee:

- -Create product backlog
- -all sprint tasks
- -created half of all meeting
- -managed everyone progress and updated everyone
- -tested censorship and tested moviePy
- -made censorship for both TTS and Text Implementation
- -tested video player
- -created video player and button for video player
- -modified visuals of subtitles
- -created one of the video templates (subway surfer)
- -assigned tasks
- -modified visuals for user interface
- -fixed bug for selecting with reddit post you would like to use (on gui)
- -Researched with API we would use for the GUI.
- Created the presentation with Basel and Rifat.

Rifat Chowdhury:

- Contributed to user stories creation and keeping files up to date.
- Managing Miro & created the License file.
- Created a timeline of the different sprints.
- Formatting and updating the Readme file constantly.
- Arranged meetings and kept track of due dates.
- Assisting Basel with the meeting logs.
- Participating in group meetings and discussions to keep track of the project progress for
- Proper documentation.
- Wrote and participated in progress report creation with Basel.
- Did all the formatting for our documents as well as contributed to the bulk writing
- Created the presentation with Basel and Hevar.
- Helped Maisam with the creation if the GUI.
- Helped Maisam with the TTS.
- Helped with testing as much as possible.
- Contributed on possible solutions in meeting to sprint retrospective meeting challenges.

- Made sure to keep the team in track with the due dates.

Maisam Anjum:

- Responsible for initial commits for project structure, git branching.
- Added to testing framework to ensure proper methodologies and documentation were present.
- Approved 10+ pull requests and integrated code into main, ensuring merging didn't cause issues.
- Configured reddit API, made changes to better suit scale of our project and switched to. json format to resolve problems, created a system to fetch reddit posts and scrape for usable data.
- Worked on various tts systems and integrated several models to find best one for our needs, researched for highest quality to lowest cost option for minimum viable product,
- Built initial GUI prototype in tkinter to allow for integration into main app.
- Made 25+ pushes to code across all branches.
- Refactored and optimized code for readability and performance.
- undertook spike testing and unit testing such that the capabilities and issues with system can be defined and addresses, focused on API systems and scalability.

Matthew Cam:

- Researched subs to use for content and created the Git project.
- Created function that grabs URLs of Reddit posts based on sub and amount requested.
- Integrated URLs grabber and content scrapper.
- Researched TTS software and provided code examples.
- Merging audio and video.
- Generation of subtitles and syncing to audio.
- Categorized videos and subs.
- Formatted videos to the correct aspect ratio and length.
- Detection and automatic downloading of necessary dependencies.
- Organized automated testing.
- Automated adding of custom videos to templates.
- Created transitions between GUI pages.
- Ensured that when the user samples audio a short audio file plays.
- Prevented bad input from GUI.
- Automated update of description from post chosen.
- Compiled our problems and solutions for this report.

Raymond Dong:

- Worked on/helped with the tts software’s.

- Helped in testing the multiple tts software used throughout the project.
- helped in performing integration testing throughout
- Researched ways to grab YouTube analytics using YouTube API
- created a doc that leads the user on this registration process to pull from the YouTube API
- Stress Tested the number of requests that can be made with the YouTube API.
- Grabbed metrics from YouTube API into a csv file got the user to read.
- Implemented a graph to allow for the user to easily read the pulled information better for each metric they want.
- Helped in maintaining code readability throughout the development process.

Required Programs:

VLC (64 Bit): https://www.videolan.org/
API Registration:
https://docs.google.com/document/d/1H72scctGxD0zaGiVjhwRCJZP5qJYSNIYLv9UBGZe2ms/edit
ImageMagick: Included with download (please make sure to select "Install Legacy Utilities" on the wizard).

Frameworks Used:

- Python
- Text to Speech
- YouTube
- Reddit
- readme.so https://imagemagick.org/index.php
- Jira (Designing roadmaps, backlogs, sprints)
- Miro (Organizing product roadmaps, meeting schedule & sprint feedback session)
- VLC: https://www.videolan.org
- ImageMagick: https://imagemagick.org/index.php
- moviePy: https://pypi.org/project/moviepy/
- Tkinter
- playsound
- GTTS: https://pypi.org/project/gTTS/
- Mutagen: https://pypi.org/project/mutagen/

Tutorial:

We created a YouTube video as a guide to using our program. It can be found here:

https://www.youtube.com/watch?v=0LPsPId1vhk

Purpose of the Program:

This program was made to capitalize on a recent uptake in Reddit TTS videos on YouTube Shorts and TikTok, one such example can be found here:

https://www.tiktok.com/@crazy.shortstories/video/7329937672993180974?q=reddit&t=1706570724915

The thought process and vision behind the program was to automate the creation of these videos while monitoring the metrics associated with each upload. These metrics would assist the user in noticing what permutations lead to the best results. Permutations, in this context, can mean anything from the video playing in the background of the finished product to the color of the subtitles. Due to budget and time constraints, our program was intended to be a proof of concept from the start. With the right tools and some extra time, our program has the potential to create videos that are indistinguishable from currently viral videos, while simultaneously advising the user on which videos' permutations resulted in the most success.

Agile Methodology:

Creating our program through an agile design process has assisted us more than expected. While studying the agile methodology in COSC 4P01, it certainly seemed interesting at the time, but we never expected it to be this practical. The following are the decisions that were made possible due to this design process:

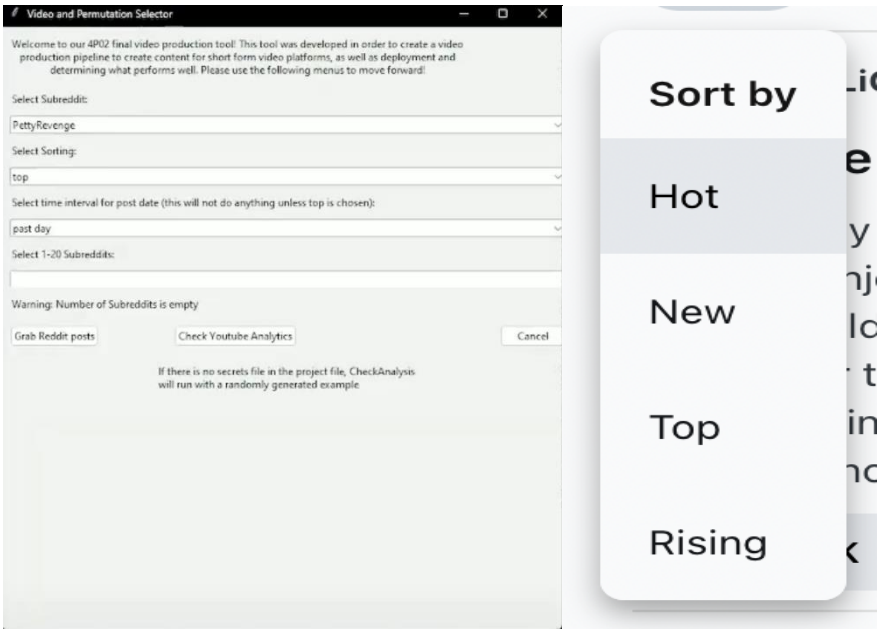
- We modified the Daily Scrums present in the typical Scrum process with Weekly Scrums. This allowed us to adapt to our lecture schedules and other course workloads to seamlessly organize meetings consistently. We also combined our Sprint Retrospective and Sprint Planning Meetings into one meeting, which was preferable for the members of our team.
- Throughout the development of the program, many ideas were cut and added from the final product. For instance, we initially had planned to create a decision tree that analyzes the metrics collected from the user's uploaded videos and output the pattern it recognizes. The agile methodology allowed us to remove this item from our backlog without much consequence.
- Due to time constraints, our third sprint was delayed by a week. However, we were able to quickly mold our schedule and adapt. While this delay wasn't ideal, the consequences were minimalized due to our software design methodology.

While the agile design process did wonders for our program's evolution. It did not come without any tradeoffs. Version control was made much harder due to the program's constantly evolving release plan. Naturally, if we had more experience with this methodology this con would not have given us many problems. Another issue was the uncertain nature of our workflow, many of our members scheduled their working sessions for this program to account for other courses' work. If an especially busy week was ahead of us, we would implement the less intensive aspects of the program while scheduling the more time-consuming tasks for later. Sometimes, due to a sudden decision, this balance was thrown off and we were forced to complete our work during a very busy period.

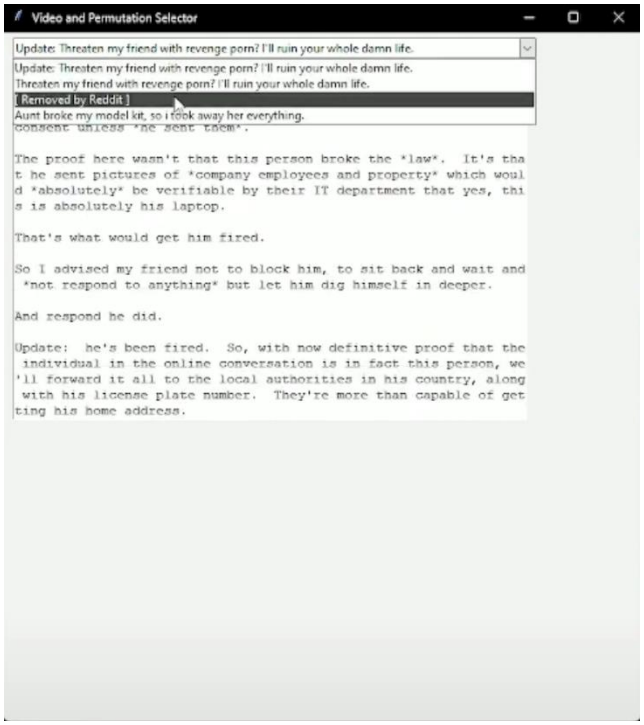
Results and Issues Faced:

Execution:

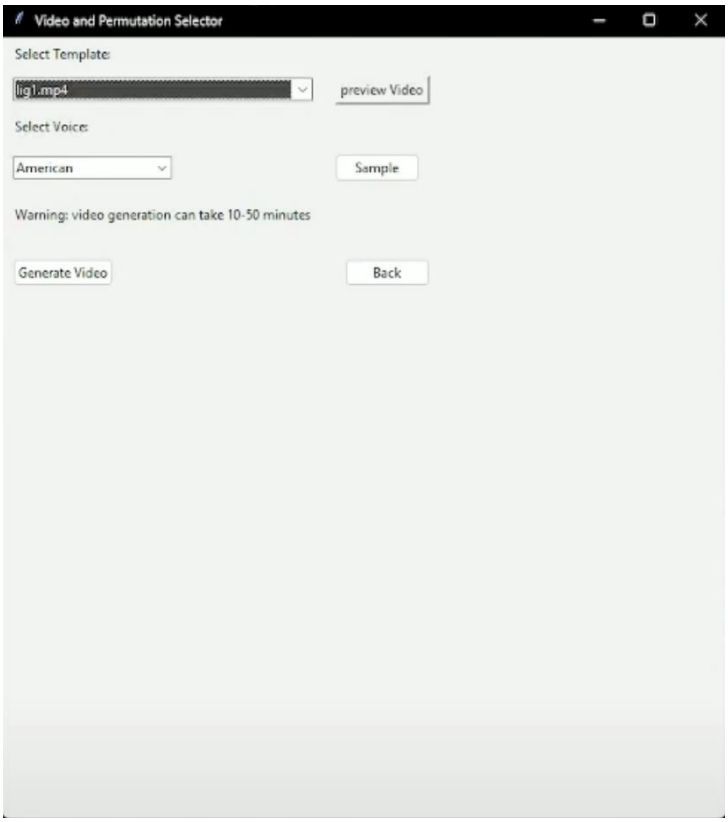
In this section of the report, we will be showing our final product in a sample execution.



This is the first window the user will see upon running the program. The first drop-down menu is filled with subreddits to pull and create a video form. The second drop down menu takes advantage of Reddit's built-in filtration system (as shown in the second screenshot) to pull a different set of posts for the user to choose from. The third drop down menu further adds to the user's pool of options by filtering the date of the Reddit posts as well. The final field takes an input from the user between the numbers 1 and 20 that decides how many posts get pulled from the user's previously selected filters.

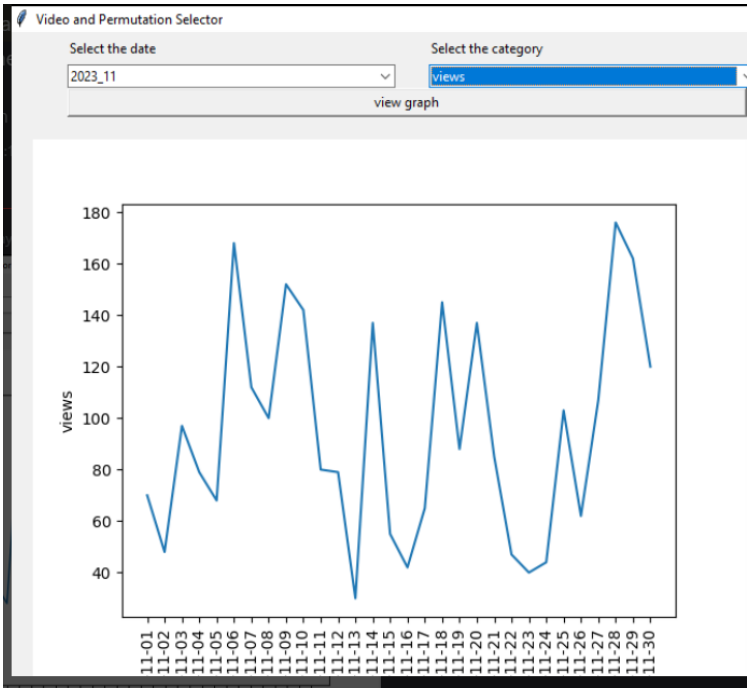


The above picture shows the Reddit posts that have been pulled based on the user's prior input. The user can select any of these posts to create a video with.



The above picture shows the final step in the video creation process. Here, the user selects the clip playing in the background of the video, along with the TTS voice's accent. When the user clicks "Generate Video", the program will get to work and generate the final video. Here are three examples of the final product: **#1**, **#2** **and** **#3**

This is a graph generated by our metric grabber.



Issues:

Throughout the program's development, we encountered many obstacles that tested our programming aptitude, communication, and delegation of tasks.

I. Problem: Grabbing reddit URLs for content scrapper.

Solution: Subreddit URLs are divided into three categories, the subreddit itself, the ordering (new, top, hot), and the time frame. By inputting those three variables the program can use the Reddit URL's json which in turn provides the posts in whatever order the user specified. The output is a list of URLs for posts that fulfill the users requested information which can in turn be fed to the content scrapper which individually grabs the posts information that is tied to the provided URL.

II. Problem: Reddit prevents multiple requests in quick succession.

Solution: Create an error handling that retries the requests until it is available. This created a further issue as waiting for the server to allow more requests can drastically increase time to complete. As this is an issue on Reddit's half and not ours the only solution was to warn users automatically when they attempt a request of 10 or more posts.

III. Problem: Program didn't work without certain programs installed.

Solution: If a dependency can be installed through pip, then it will create an error handling for imports that checks if an import is possible and if not, it will use pip to install the dependency. For Imagemagick.io, which made the subtitles possible, it could not download through pip or any other automated way. To combat this an error handling message checks for Imagemagick.io and

if it finds it missing then it will send an error message to the user which will inform the user to activate the provided EXE to download the dependency.

IV. Problem: Censored words had to be read aloud.

Solution: Words were censored with *, for example fuck is f***, for the audio the text detects censored words and replaced them with REDACTED to be read out loud. For example, “What the fuck” is censored in the text as “What the f***” but is read aloud as “What the REDACTED”.

V. Problem: Program needed the video and audio to be part of a single file.

Solution: Almost the entirety of the final video generation was handled by me. Audio was generated by the user’s input and created audio.mp3, the program would then take audio.mp3 and merge it with a selected template generating temp.mp4, then in turn snippets of temp.mp4 would have subtitles added resulting in end.mp4. After the generation is finished any temp files are deleted.

VI. Problem: Subtitles needed to sync with the audio.

Solution: Because TTS programs do not offer a way of previewing the length of a text, if it were to become an MP3 file a brute force method had to be used. By dividing the audio into sound breaks (periods, commas, exclamation marks) the TTS program then creates an MP3 of the sentence then mutagen in turn checks the length of the MP3 to determine how long text needs to stay on video. To determine how long an audio break takes, multiple examples were made, for example (Line one. Line two) as both the sum of two different MP3 length and one whole then determining the difference between the two in length which gave an answer of 0.1 second difference. Then generating the subtitles was a simple matter of ensuring that the text only appears for how long the audio lasts with 0.1 second subtracted to account for the pause between sentences. While TTS can take a significant amount of time, testing shows that it increases at an exponential rate meaning that the generation of ten sentences takes significantly longer than generating ten sentences separately. While there is a difference between audio and text in terms of censorship, the timing considers the audio-censored text to ensure proper timing.

VII. Problem: Videos couldn’t last longer than the templates’ length.

Solution: If audio was longer than the templates then a new video was generated that loops the old video until the necessary length is achieved. This added time to the generation but otherwise worked perfectly.

VIII. Problem: GUI was limited to one page.

Solution: Because certain information had to be grabbed before moving on to another section multiple pages had to be introduced. The first page is dedicated to grabbing the various posts, when transitioning to the next page it grabs all the information the user requested, in the case of the program not finding anything it doesn’t change and instead gives the user an error message.

When the next page is reached a drop down contains all the posts and a text box is if gives the information of each post that automatically updates when a post is selected. After the post is selected a series of templates is offered based on the category chosen with custom templates always being provided, the choice of what voice is to be used and samples of each voice, then all that is left is generating audio.

IX. Problem: Templates provided did not work with the program.

Solution: The program took MP4s, and the videos provided were GIFs or YouTube videos. For the gifs they had to be converted to MP4 format and the correct aspect ratio, after which any possible downsizing of the file's audio took place. The YouTube videos had to be downloaded and converted into clips to be used as originally their length was hours.

X. Problem: Automated testing was disjointed.

Solution: Took everyone's automated testing and placed them into a single testing.py file. Every test was put into a category and order as such, tests that fed into each other were made to ensure cohesion between different people's code.

XI. Problem: GUI input would result in errors.

Solution: Limited input so that only appropriate input was allowed. For example, number of posts only allows numerical inputs and if any input results in a number higher than twenty or lower than 0 then it is prevented.

XII. Problem: User had no idea what each voice sounded like.

Solution: Created three sample audios, each saying, "This is the x voice" and created a sample button that played the audio to the corresponding voice chosen.

Testing

Many different testing methods were used to get the program to its final version:

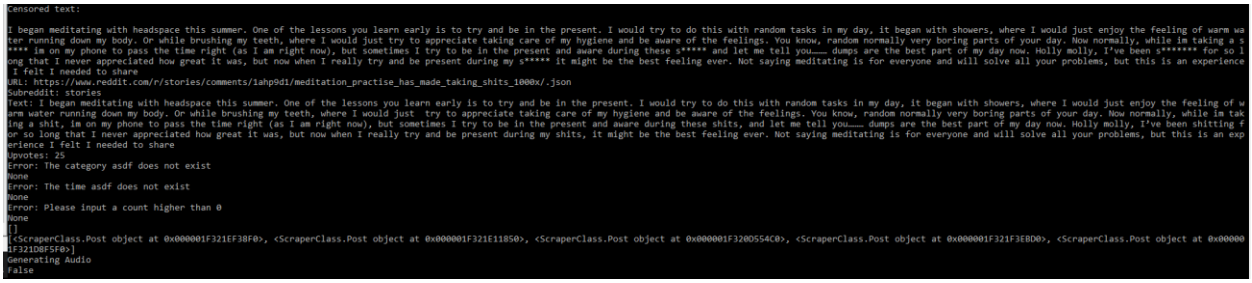
I. **Acceptance Testing:** Went through each function to ensure it accepts and disregards the relevant inputs. To that end, we tested many possible false inputs to the GUI and fixed any discrepancies.
II. **Spike Testing:** We wanted to see how big of an effect making rapid requests to Reddit would create. As a result, we learned that the program's execution time is heavily tied to the amount of pull requests made to Reddit. This was key in reducing our execution time.
III. **Integration Testing:** When new features were added to the GUI, we ensured that they interfaced with each other with no problems. This involved testing every possible combination on the drop-down menu in the GUI and ensuring that the outputs function as expected when the posts are listed.
IV. **Functional and Regression Testing:** Extensively testing each module to ensure it is working as intended. For example, the video player's subtitle addition. Subtitles were the

biggest hurdle we had to overcome. And this testing method did a lot of the heavy lifting in correcting those problems. To completely fix this issue, we applied Regression testing to subtitle size, time, outline, and length.

V. **Performance Testing:** We ran the program under many different inputs to analyze our execution time. Thanks to the Spike testing previously conducted, and other improvements, we managed to cut down our execution time by upwards of 50 percent.

VI. **Manual Testing:** We employed this method for a large variety of minimal problems. This simple testing method lends itself well to simple problems and was also critical to our success.

As seen in our testing.py file, we applied automated testing in the earlier stages of the program to ensure our grabber did not malfunction.



The above execution tests the censorship, scraping, URL grabber, and the TTS respectively. We absolutely had to make sure this runs optimally before we got to work on generating videos.

Expanding on the subtitle problem we endured. They resulted in failed execution such as **#1 and #2.**Thanks to the combination of testing methods we employed and described above, this was summarily fixed.

Meeting Logs

I. Meeting 1 (Saturday January 20th):

Agenda: Release Planning & Backlog Meeting

1 - License.md: GPL or MIT.

2 - We decided to use Miro for recording our progress and scheduling our meetings. Rifat will make the board and invite us later.

3 - Sprint Planning: Biweekly for now, weekly later if necessary.

4- We made the decision to use Jira to record our sprints and tasks.

5- Product Owner and Project Manager will handle all the reporting. We will reference our Jira often to stay updated on sprint progress.

6 - Talked about the rough idea of the project. Hevar mentioned an app that can pull from Reddit very easily and we can use that in the program.

Questions to ask the TA (Will be updated on Monday as well):

1 - How often do we need to report to you? Will it be after every sprint/meeting, or will it simply be at the assignment's pace? Do we need to record our meetings in detail?

2 - How will the program itself be graded? Will the complexity of it be accounted for? Will the amount of testing be graded? Etcetera.

3 - How detailed do the sprint reports need to be? How general can we be when it comes to the tasks we record on Jira?

4 - How should we handle pairing? Will it be better for 2 dev's to work on the same thing for the duration of the sprint, or is it better to have each dev work on one aspect instead?

5 - What exactly is expected of the Product Owner and Project Manager? We already made the decision to handle all the reporting, but what else can we do to even the workload?

II. Meeting 2: (Monday January 22):

Agenda: Sprint Review & Retrospective

1. We confirmed the license GPL v3.0 as our license format.

2. For the Ta's meeting we decided to create 4 user stories:

User Statistic analyze, User entrepreneurs, Users develop, User youtuber.

3. For the Ta's meeting -15 mins project basic discussion & 5 mins Ta answering question.

Questions to ask the TA (added with the Saturday's meeting questions):

6. Is there any minimum push requirements for everyone for the project?

7. How'd you suggest approaching the 5% bonus marks for the assessment part (slide 17 in the course outline)?

III. Meeting 3: (Monday January 29th):

Agenda: Sprint Planning

1 - Maisam created multiple branches to maintain version control.

2 - Maisam did some research on Reddit's API and found out that simply adding. Json at the end of the URL creates a file that contains all the relevant post's information. This will be our method for pulling reddit posts.

3 - Matthew researched subreddits and it prompted us to discuss comment-dependent versus post-dependent content. This will have to be decided a little later, but the issue has been discovered. We decided to leave this decision to later down the line.

4 - Matthew brought up censorship and how we should tackle it. Firstly, we need to define what censorship entails. So far, swear words, sensitive words, strong words (kill, murder, etc...) can be replaced with words like "unalive" etcetera.

5 - We discussed security in the program and what it entails, we decided it mainly means we need to try to ensure that metrics and the like are not easily leaked or stolen.

6 - We discussed code pacing. And decided our current pace is fine for now. But there will likely be a crunch period which we will try to avoid.

7 - Basel brought up the point of measuring data obtained both from the program and through user metrics. It was decided that concessions will be made by the dev team to be able to measure this data.

8 - Basel brought up the issue of pulling metrics from our posts to serve in the decision tree. We agreed that it wouldn't pose a big challenge if we use websites like social blade. Details will have to wait for its relevant sprint, but we have a good idea on how to go about doing it.

9 - Raymond brought up a question about what qualifies as a "manageable" post length. Hevar suggested measuring the text-to-speech's time through google and the decision should be made based on that. Raymond decided he would do this through measuring WPM (words-per-minute). But further detail will have to wait for the creation of the voice-over module. This will warrant extensive testing.

10 - We decided to research and see if Reddit has a system of collecting screenshots. This brought up a permutation possibility which is either having subtitles on the screen or a screenshot of the post itself. We can upload a different version of this to two channels and see which one generates more attention.

IV. Meeting 4: (Monday February 5th):

This meeting is taking place just before the sprint ends. Not much was discussed but we ensured everything was going to plan.

Agenda: Sprint Review

1 - 6 out of the 8 sprints were marked as done

2 -Discussion was done regarding one of the sprint backlogs - sanitizing user posts for Censorship and security.

3- We discussed when the TA meeting was taking place.

4- We planned the Sprint Retrospective Meeting and set plans for the next sprint's Sprint Planning Meeting.

V. Meeting 5: (Sunday February 11th):

This meeting will serve as a Sprint Retrospective Meeting, we will tie up any loose ends and reflect on what we could have done better.

Agenda: Sprint Retrospective

1- It's come to our attention that integrating the swear word database for censorship along with our reddit post grabber is proving more difficult than anticipated. This task will now be moved to the next sprint.

2- We cited a database we used for swearing words and decided on a reliable way to cite our sources from now on.

3- What everyone learned from the first sprint proved to be too little compared to the detail that's expected of us. So, Basel and Rifat will take measures to make some stuff up. As well as ask questions to the dev team to get some more context.

4- We decided to compile our work and what we learned. The Reddit API proved to be difficult to work with, so we had to come up with a roundabout way of interfacing with Reddit. Also, unfortunately it is not possible to run the program through GitHub directly which made our work more difficult. We decided that a more reliable and readable way of formatting our code to allow the other developers to grab what they need from another member's commits into their own branch will prove helpful. An extended looping issue in our pulling algorithm was also discovered during one of our tests, which will need to be addressed very soon before building on top of the foundation we made this sprint, as execution is taking much longer than desired.

VI. Meeting 6: (Tuesday February 13):

This meeting will serve as the sprint planning session of the second sprint.

Agenda: Sprint Planning

1. Old Sprint Review: - The team initiated the meeting by discussing the outcomes of the previous sprint. Any challenges or unfinished tasks were reviewed, and decisions were made on whether they should be included in the upcoming sprint.

2. API Integration: - The team engaged in a comprehensive discussion on the integration of multiple open-source solutions for the API. Various options were explored, and decisions were made to enhance the functionality and efficiency of the system.

3. Sprint 2 Kickoff: - Confirmation was given that Sprint 2 is scheduled to commence tomorrow. The team expressed readiness and commitment to achieving the sprint goals.

4. Potential Additional Sprints: - Considering the complexity of the data, there was a proposal for two more sprints, with the possibility of a third if required. The decision was not finalized, and further assessment will be made during the ongoing sprints.

5. User Backlogs for Sprint 3: - The team discussed and added user backlogs for Sprint 3, demonstrating a proactive approach to ensure a well-prepared backlog for future sprints.

6. Time Constraints Discussion: - Deliberations were held on time constraints, acknowledging the importance of managing time effectively to meet project milestones. The team committed to staying vigilant and proactive in addressing any potential challenges related to time.

7. Next TA meeting discussion - Discussion regarding the next meeting with the TA.

8. Assigning tasks to the dev team - The Scrum master (Hevar) assigned individual tasks to the dev team (Maisam, Matt & Raymond). More roles might be added depending on the need.

9. Generating Video Templates - Produce 18 video templates for the coming sprint 3.

10. Possible Challenges of Sprint 3 - Discussion was made on the censorship of the voice.

VII. Meeting 7: (Monday February 26th):

Agenda: Sprint Review

1 - Hevar made the decision to extend this sprint's duration by one week. Our current topic of discussion is analyzing how to increase our efficiency with completing these tasks by breaking down the current issues using Jira.

2- We discussed tomorrow's TA meeting to get in the correct mindset. As a result of this discussion, we are going to take measures to compile everything we recorded so far to show the TA.

3- The developer team discussed how to handle text-to-speech (voices, censorship, and timing).

VIII. Meeting 8: (Monday March 4th):

Agenda: Sprint Review & Weekly Scrum

1 - Basel brought up the issue of censorship causing the context of a sentence to change or not make sense entirely. It was agreed that, to combat this, we would show the first letter of the swear word, the asterisks for the rest of the letters.

2 - We decided to investigate the subreddit swearing guidelines, to see what people do instead of swearing. Do they replace the words with another uncensored variation, or do they avoid their use entirely?

3 - Discussed the final details of the TTS system before the sprint ends to ensure we are on schedule. The main point was an agreement to add a bleep sound, though this will be done next sprint. For now, we will simply replace these words with "redacted."

4 - A short update on version control was conducted to ensure everyone is on the same working version.

IX. Meeting 9: (Monday March 11th):

Agenda: Sprint Retrospective & Sprint Planning

1 - Last Sprint went with success. We finished all the tasks on time and there were no major roadblocks.

2 - Matt raised the question of producing the 18 video templates. We are deciding on how to divide the work for the video template. We agreed on 2-3 videos for each person where each video is 2 mins by this week before the end of this sprint.

Aspect ratio - You tube short aspect ratio 9/16, resolution - 1920x1080P, Genre - Games, Videos - no audio & 1-2 min in length:

3 - We talked on all the different platform timings on short and decided on having You tube shorts as our platform.

4 - We went through the backlogs for the current sprint and assigned tasks based on the dev team discussions (Maisam, Matt, Raymond).

5 - Maisam had a question on how to deal with uploading and having multiple accounts for the interface.

6 - We had a bit of discussion on when the next meeting will be.

X. Meeting 10 (Monday March 11th):

Agenda: Sprint Review

1 - Brought up dummy accounts and user access to both accounts. Hevar added this to the backlog.

2 - Delegated video templates and genre classifying to Basel. Made the decision to delegate prototype GUI ideas to Basel and Rifat who will begin working on this starting next sprint (March 19th).

3 - Basel brought up the idea of classifying different permutations (videos, sounds, font(?), TTS voice(?)) and then adding a filter to the GUI for better user experience. If time allows, we will flesh out this feature, if not, we will still classify the permutations with genres to better the decision trees' intelligence.

4 - Set a more concrete timeline for project development and decided to proceed under the assumption that we need to have the program finished by April 16th.

XI. Meeting 11 (Monday March 18th):

Agenda: Sprint Review.

1 - fixed image magic not working for everyone even though the directories were still there

2 - discussing removal of binary tree.

3 - maybe modifying the text to speech module to use a more realistic version

4 - checking in to see if we are up to date with tasks

XII. Meeting 12 (Monday March 25th):

Agenda: Sprint Retrospective & Sprint Planning Meeting

1 - Discussed progress and exchanged what was done was requires more time. Currently, the focus is improving user experience through instructions.

2 - Discussed scheduling and priorities when it comes to the GUI and Progress Report 2. We decided that prioritizing the progress report is more important, so Basel and Rifat will be finishing it first. Additionally, we decided to have Basel and Rifat ready the GUI prototypes ready by April 4-6th. This gives the team enough time to implement it.

3 - We confirmed our deadlines and oriented ourselves for when everything is due.

4 - Hevar and Matt discussed video storage and pulling from local folders. We made sure everything was in order.

5 - We discussed which programming language would be best for GUI design. The final decision will be made soon.

6 - We decided to assign checking for a better TTS to Maisum, he will oversee choosing and implementing that.

7 - The final decision for permutation storage was made: We will utilize folders and subfolders that each store a genre of permutation. E.g. "Horror Background Videos folder, Light-hearted Background Videos folder, Horror Music folder, etc..."

XIII. Meeting 13 (Monday April 1st):

Agenda: Weekly Scrum

1 - Went over progress report 2's final checks and brainstormed ways to lower execution time.

2 - Ran some tests on code runtime to further reduce it.

3 - Made the decision to add our own exceptions to code for easier and more informative testing. This was added to the backlog.

4 - Discussed the feasibility of adding a progress bar to the GUI.

XIV. Meeting 14 (April 8th):

Agenda: Weekly Scrum.

1. The team used this meeting as a bug–fixing session, which is what we needed most.
2. Final checks to ensure we will finish the program on time.
3. Delegation of tasks by the Scrum Master.

XV. Meeting 15 (April 15th):

Agenda: Sprint Retrospective.

1. Initial plans for the presentation were set.
2. A final bug-fixing discussion was conducted.
3. Final delegation of tasks by the Scrum Master.

XVI. Meeting 16 (April 20th):

1 - Discussed how to handle the presentation. It's been decided that we will alternate between Basel and Hevar. Basel will handle everything except the technical aspects of the program, while Hevar will elaborate more on code design and challenges faced.

2 - The team touched base on what is left to achieve for the program to be completed. We had our best execution today, after fixing font, settling timings, optimizing execution time and finishing up the GUI's warnings, we can safely say that the program is in its final form.

3 - We attempted to run the program on a separate PC, this took up most of the meeting as the prerequisites required took some time to install.

4 - We had a final team troubleshooting session to ensure everything is in place for the presentation.

Sprint Backlogs

Sprint 1

Completed issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Story points
SMPG-1	register with reddit api	Task		DONE	MA	-
SMPG-2	use reddit api to pull from given url	Task		DONE	MA	-
SMPG-4	make sure the post that is grabbed is a manageable length and does not cut...	Task		DONE	RD	-
SMPG-5	sanitizing user posts for Censorship and security	Task		DONE	HH	-
SMPG-11	research which subreddits are the best to grab stories from	Task		DONE	MC	-
SMPG-36	research how to grab and read stories from reddit	Task		DONE	MA	-
SMPG-37	allow the reddit grabber to grab stories from different reddits	Task		DONE	MC	-

Issues completed outside of sprint

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Story points
SMPG-3	fetching from random 10 top posts from a given subreddit	Task		DONE	MC	-

Date	Event	Issue	Completed	Scope
Sun, Jan 28 2024, 8:00pm	Sprint started	SMPG-1 register with reddit api SMPG-2 use reddit api to pull from given url SMPG-3 fetching from random 10 top posts from a given subreddit SMPG-4 make sure the post that is grabbed is a manageable length and does not cut off any part of the story SMPG-5 sanitizing user posts for Censorship and security SMPG-11 research which subreddits are the best to grab stories from SMPG-36 research how to grab and read stories from reddit SMPG-37 allow the reddit grabber to grab stories from different reddits	0	0
Sun, Feb 04 2024, 4:40pm	Issue completed	SMPG-11 research which subreddits are the best to grab stories from	0	0
Mon, Feb 05 2024, 6:03pm	Issue completed	SMPG-4 make sure the post that is grabbed is a manageable length and does not cut off any part of the story	0	0
Mon, Feb 05 2024, 6:03pm	Issue completed	SMPG-2 use reddit api to pull from given url	0	0
Mon, Feb 05 2024, 6:03pm	Issue completed	SMPG-1 register with reddit api	0	0
Mon, Feb 05 2024, 6:03pm	Issue completed	SMPG-36 research how to grab and read stories from reddit	0	0
Tue, Feb 06 2024, 6:28pm	Issue completed	SMPG-37 allow the reddit grabber to grab stories from different reddits	0	0
Tue, Feb 06 2024, 6:28pm	Issue completed	SMPG-5 sanitizing user posts for Censorship and security	0	0
Thu, Feb 22 2024, 2:27pm	Sprint completed	SMPG-1 register with reddit api SMPG-2 use reddit api to pull from given url SMPG-3 fetching from random 10 top posts from a given subreddit SMPG-4 make sure the post that is grabbed is a manageable length and does not cut off any part of the story SMPG-5 sanitizing user posts for Censorship and security SMPG-11 research which subreddits are the best to grab stories from SMPG-36 research how to grab and read stories from reddit SMPG-37 allow the reddit grabber to grab stories from different reddits	0	0

Sprint 2

Backlog

Epic | Insights | View settings

voice of the reddit bot 22 Feb – 6 Mar (8 issues) — Complete sprint

~~SMPG-40~~ Research useable voice modules	DONE	MC
~~SMPG-15~~ make the voice of the bot read any text	DONE	MA
~~SMPG-20~~ develop multiply different options for the voice (around 3)	DONE	MC
~~SMPG-18~~ integrate the voice with the reddit grabber so that it can read the text	DONE	RD
SMPG-16 make censorship for the voice so that it avoid inappropriate words	TO DO	HH
SMPG-17 test the censorship and voice so that there is no errors involved	TO DO	HH
SMPG-19 test the integration so that there is no errors because of the implementation	TO DO	RD
SMPG-21 allow easy switching between voices and test that it doesn't cause errors	TO DO	MA

+ Create issue

Completed issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Story points
SMPG-1	register with reddit api	Task		DONE	MA	-
SMPG-2	use reddit api to pull from given url	Task		DONE	MA	-
SMPG-4	make sure the post that is grabbed is a manageable length and does not cut...	Task		DONE	RD	-
SMPG-5	sanitizing user posts for Censorship and security	Task		DONE	HH	-
SMPG-11	research which subreddits are the best to grab stories from	Task		DONE	MC	-
SMPG-36	research how to grab and read stories from reddit	Task		DONE	MA	-
SMPG-37	allow the reddit grabber to grab stories from different reddits	Task		DONE	MC	-

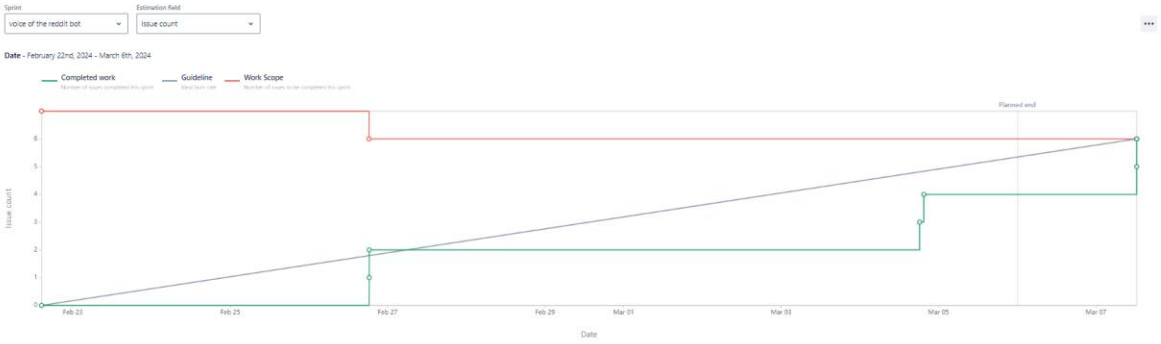
Issues completed outside of sprint

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Story points
SMPG-3	fetching from random 10 top posts from a given subreddit	Task		DONE	MC	-

Date	Event	Issue	Completed	Scope
Thu, Feb 22 2024, 2:27pm	Sprint started	SMPG-15 make the voice of the bot read any text SMPG-16 make censorship for the voice so that it avoid inappropriate words SMPG-17 test the censorship and voice so that there is no errors involved SMPG-18 integrate the voice with the reddit grabber so that it can read the text SMPG-19 test the integration so that there is no errors because of the implementation SMPG-20 develop multiply different options for the voice (around 3) SMPG-21 allow easy switching between voices and test that it doesn't cause errors SMPG-40 Research useable voice modules SMPG-26 combine the video with the text from reddit and the voice to produce content that can be uploaded	0	0
Mon, Feb 26 2024, 6:17pm	Removed from sprint	SMPG-26 combine the video with the text from reddit and the voice to produce content that can be uploaded	0	0
Mon, Feb 26 2024, 6:23pm	Issue completed	SMPG-18 integrate the voice with the reddit grabber so that it can read the text	0	0
Mon, Feb 26 2024, 6:23pm	Issue completed	SMPG-15 make the voice of the bot read any text	0	0

Sprint 3



Date	Event	Issue	Completed	Scope
Thu, Feb 22 2024, 2:27pm	Sprint started	SMPG-15 make the voice of the bot read any text SMPG-16 make censorship for the voice so that it avoid inappropriate words SMPG-17 test the censorship and voice so that there is no errors involved SMPG-18 integrate the voice with the reddit grabber so that it can read the text SMPG-19 test the integration so that there is no errors because of the implementation SMPG-20 develop multiply different options for the voice (around 3) SMPG-21 allow easy switching between voices and test that it doesn't cause errors SMPG-40 Research useable voice modules SMPG-26 combine the video with the text from reddit and the voice to produce content that can be uploaded	0	7
Mon, Feb 26 2024, 6:17pm	Removed from sprint	SMPG-26 combine the video with the text from reddit and the voice to produce content that can be uploaded	0	7 → 6
Mon, Feb 26 2024, 6:23pm	Issue completed	SMPG-18 integrate the voice with the reddit grabber so that it can read the text	0 → 1	6
Mon, Feb 26 2024, 6:23pm	Issue completed	SMPG-15 make the voice of the bot read any text	1 → 2	6
Mon, Mar 04 2024, 6:13pm	Issue completed	SMPG-21 allow easy switching between voices and test that it doesn't cause errors	2 → 3	6
Mon, Mar 04 2024, 7:29pm	Issue completed	SMPG-16 make censorship for the voice so that it avoid inappropriate words	3 → 4	6
Thu, Mar 07 2024, 12:23pm	Issue completed	SMPG-17 test the censorship and voice so that there is no errors involved	4 → 5	6
Thu, Mar 07 2024, 12:23pm	Issue completed	SMPG-19 test the integration so that there is no errors because of the implementation	5 → 6	6
Thu, Mar 07 2024, 12:24pm	Sprint completed	SMPG-15 make the voice of the bot read any text SMPG-16 make censorship for the voice so that it avoid inappropriate words SMPG-17 test the censorship and voice so that there is no errors involved SMPG-18 integrate the voice with the reddit grabber so that it can read the text SMPG-19 test the integration so that there is no errors because of the implementation SMPG-20 develop multiply different options for the voice (around 3) SMPG-21 allow easy switching between voices and test that it doesn't cause errors SMPG-40 Research useable voice modules	6	6

Projects / Social Media Post Generator

Backlog

Epic · Insights · View settings

Create a and combine video · Add dates · (9 issues) · 0 0 0 · Start sprint

SMPG-6 producing 18 video templates	TO DO
SMPG-41 allow user to upload file	TO DO
SMPG-43 create generic background for text and make it readable	TO DO
SMPG-39 create subtitles for reddit post on video automatically	TO DO
SMPG-26 combine the video with the text from reddit and the voice to produce content that can be uploaded	TO DO
SMPG-12 produce a method for the bot to upload to multiple different social platforms	TO DO
SMPG-35 Test that the metrics given returned from the video are sound and accurate	TO DO
SMPG-9 grab video metrics from a given channel (watch time , total clocks, total view and total shares)	TO DO
SMPG-44 test that the given metrics are similar to the ones stated in the creator hub for the video	TO DO

+ Create issue

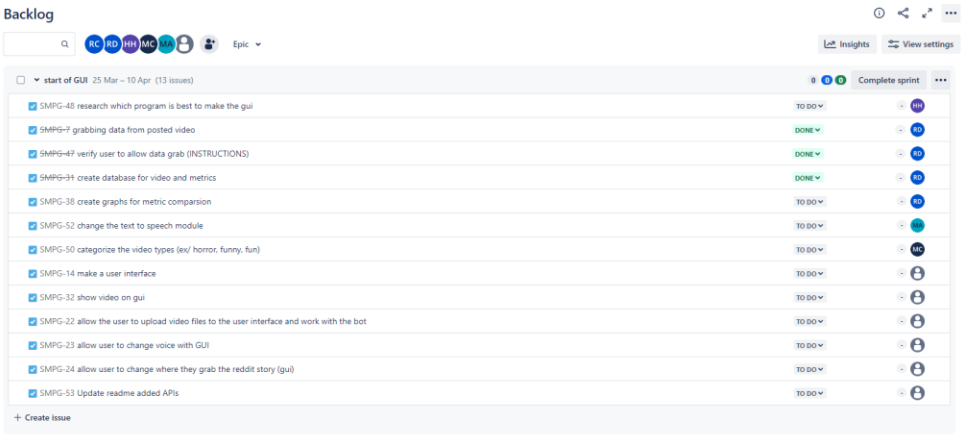
Completed issues — View in issue navigator

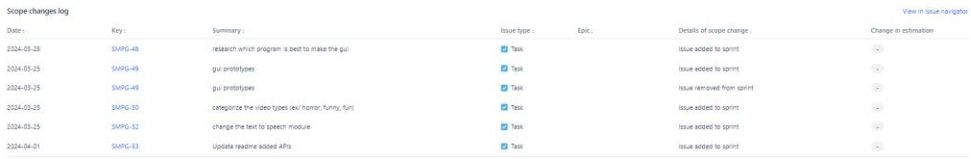
Key	Summary	Issue type	Epic	Status	Assignee	Story points
SMPG-6	producing 18 video templates	Task		DONE		-
SMPG-35	Test that the metrics given returned from the video are sound and accurate	Task		DONE	RD	-
SMPG-41	allow user to upload file	Task		DONE	MA	-
SMPG-12	produce a method for the bot to upload to multiple different social platforms	Task		DONE		-
SMPG-9	grab video metrics from a given channel (watch time , total clocks, total view and total...	Task		DONE	RD	-
SMPG-43	create generic background for text and make it readable	Task		DONE	HH	-
SMPG-39	create subtitles for reddit post on video automatically	Task		DONE	MC	-
SMPG-44	test that the given metrics are similar to the ones stated in the creator hub for the video	Task		DONE	RD	-
SMPG-26	combine the video with the text from reddit and the voice to produce content that ca...	Task		DONE	MC	-
SMPG-46	insert reddit title template above text	Task		DONE		-

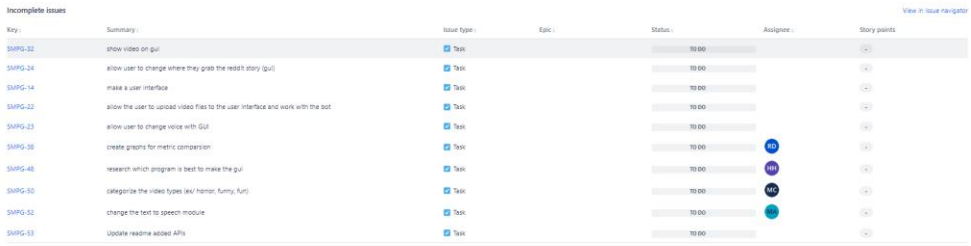
Issues completed outside of sprint

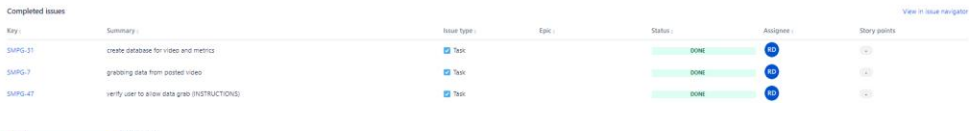
Key	Summary	Issue type	Epic	Status	Assignee	Story points

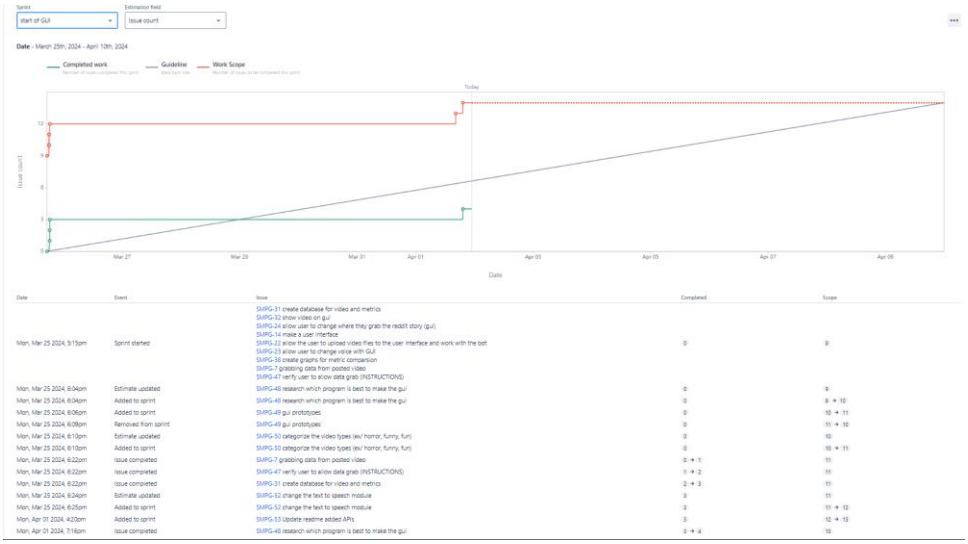
Sprint 4


Backlog
Epic
Insights
View settings
start of GUI 25 Mar – 10 Apr (13 issues)
Complete sprint
SMPG-48 research which program is best to make the gui
SMPG-7 grabbing data from posted video
SMPG-47 verify user to allow data grab (INSTRUCTIONS)
SMPG-31 create database for video and metrics
SMPG-38 create graphs for metric comparsion
SMPG-52 change the text to speech module
SMPG-50 categorize the video types (ex/ horror, funny, fun)
SMPG-14 make a user interface
SMPG-32 show video on gui
SMPG-22 allow the user to upload video files to the user interface and work with the bot
SMPG-23 allow user to change voice with GUI
SMPG-24 allow user to change where they grab the reddit story (gui)
SMPG-53 Update readme added APIs
Create issue

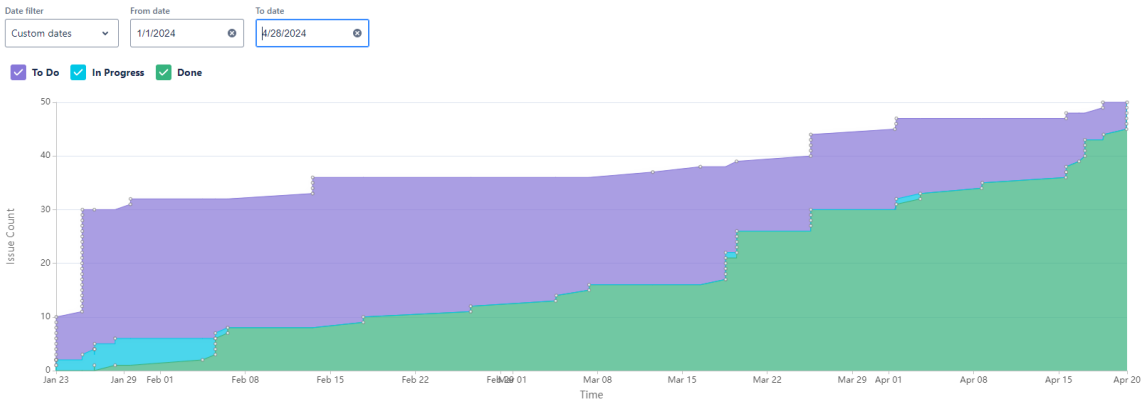

Scope changes log
View in issue navigator
Date
Key
Summary
Issue type
Epic
Details of scope change
Change in estimation
2024-03-25 SMPG-48 research which program is best to make the gui Task Issue added to sprint
2024-03-25 SMPG-49 gui prototypes Task Issue added to sprint
2024-03-25 SMPG-49 gui prototypes Task Issue removed from sprint
2024-03-25 SMPG-50 categorize the video types (ex/ horror, funny, fun) Task Issue added to sprint
2024-03-25 SMPG-52 change the text to speech module Task Issue added to sprint
2024-04-01 SMPG-53 Update readme added APIs Task Issue added to sprint


Incomplete issues
View in issue navigator
Key
Summary
Issue type
Epic
Status
Assignee
Story points
SMPG-32 show video on gui Task TO DO
SMPG-24 allow user to change where they grab the reddit story (gui) Task TO DO
SMPG-14 make a user interface Task TO DO
SMPG-22 allow the user to upload video files to the user interface and work with the bot Task TO DO
SMPG-23 allow user to change voice with GUI Task TO DO
SMPG-38 create graphs for metric comparsion Task TO DO RD
SMPG-48 research which program is best to make the gui Task TO DO HH
SMPG-50 categorize the video types (ex/ horror, funny, fun) Task TO DO MC
SMPG-52 change the text to speech module Task TO DO MA
SMPG-53 Update readme added APIs Task TO DO

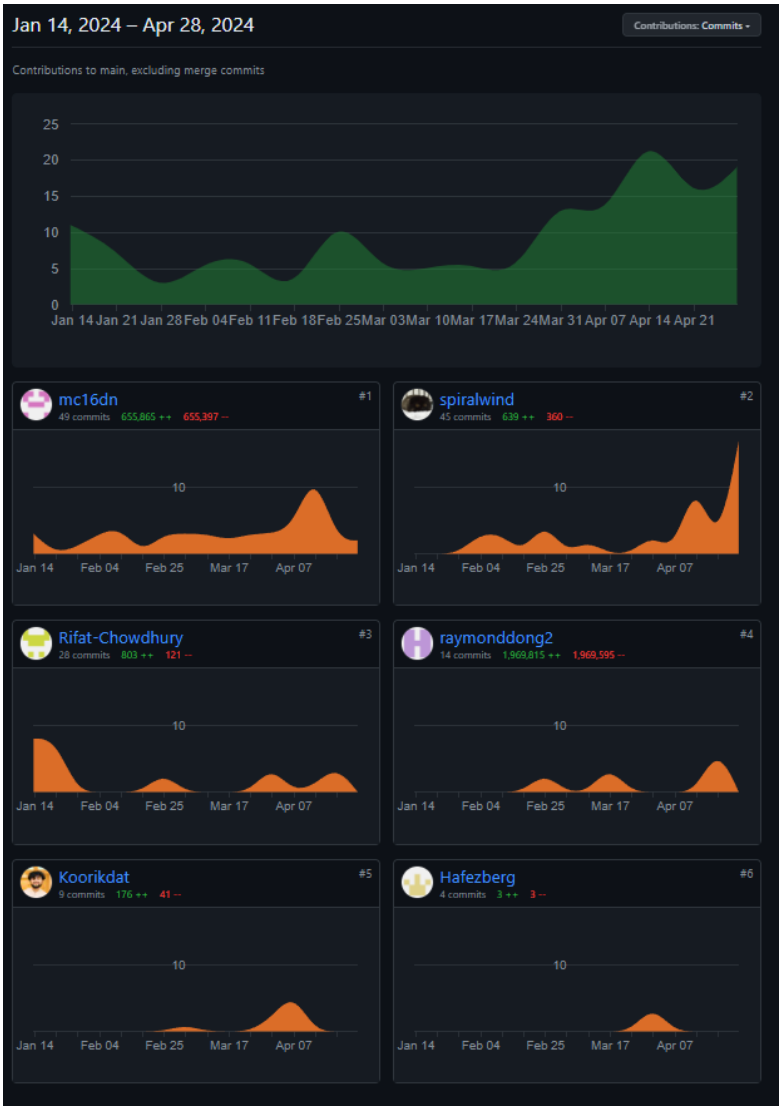

Completed issues
View in issue navigator
Key
Summary
Issue type
Epic
Status
Assignee
Story points
SMPG-31 create database for video and metrics Task DONE RD
SMPG-7 grabbing data from posted video Task DONE RD
SMPG-47 verify user to allow data grab (INSTRUCTIONS) Task DONE RD

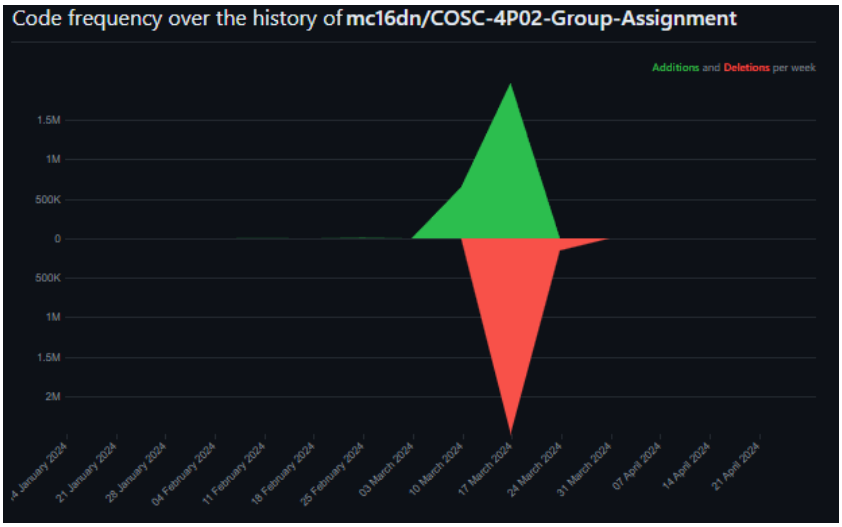

Sprint
start of GUI
Estimation field
Issue count
Date - March 25th, 2024 - April 10th, 2024
Completed work
Guideline
Work Scope
Today
Issue count
Mar 27
Mar 29
Mar 31
Apr 01
Apr 03
Apr 05
Apr 07
Apr 09
Date
Date
Event
Issue
Completed
Scope
Mon, Mar 25 2024, 9:15pm Sprint started
SMPG-31 create database for video and metrics
SMPG-32 show video on gui
SMPG-24 allow user to change where they grab the reddit story (gui)
SMPG-14 make a user interface
SMPG-22 allow the user to upload video files to the user interface and work with the bot
SMPG-23 allow user to change voice with GUI
SMPG-38 create graphs for metric comparsion
SMPG-7 grabbing data from posted video
SMPG-47 verify user to allow data grab (INSTRUCTIONS)
Mon, Mar 25 2024, 6:04pm Estimate updated SMPG-48 research which program is best to make the gui 9 → 10
Mon, Mar 25 2024, 6:04pm Added to sprint SMPG-48 research which program is best to make the gui 10
Mon, Mar 25 2024, 6:06pm Added to sprint SMPG-49 gui prototypes 10 → 11
Mon, Mar 25 2024, 6:09pm Removed from sprint SMPG-49 gui prototypes 11 → 10
Mon, Mar 25 2024, 6:10pm Estimate updated SMPG-50 categorize the video types (ex/ horror, funny, fun) 10
Mon, Mar 25 2024, 6:10pm Added to sprint SMPG-50 categorize the video types (ex/ horror, funny, fun) 10 → 11
Mon, Mar 25 2024, 6:22pm Issue completed SMPG-7 grabbing data from posted video 11
Mon, Mar 25 2024, 6:22pm Issue completed SMPG-47 verify user to allow data grab (INSTRUCTIONS) 11
Mon, Mar 25 2024, 6:22pm Issue completed SMPG-31 create database for video and metrics 11
Mon, Mar 25 2024, 6:24pm Estimate updated SMPG-52 change the text to speech module 11
Mon, Mar 25 2024, 6:25pm Added to sprint SMPG-52 change the text to speech module 11 → 12
Mon, Apr 01 2024, 4:20pm Added to sprint SMPG-53 Update readme added APIs 12 → 13
Mon, Apr 01 2024, 7:16pm Issue completed SMPG-48 research which program is best to make the gui 10

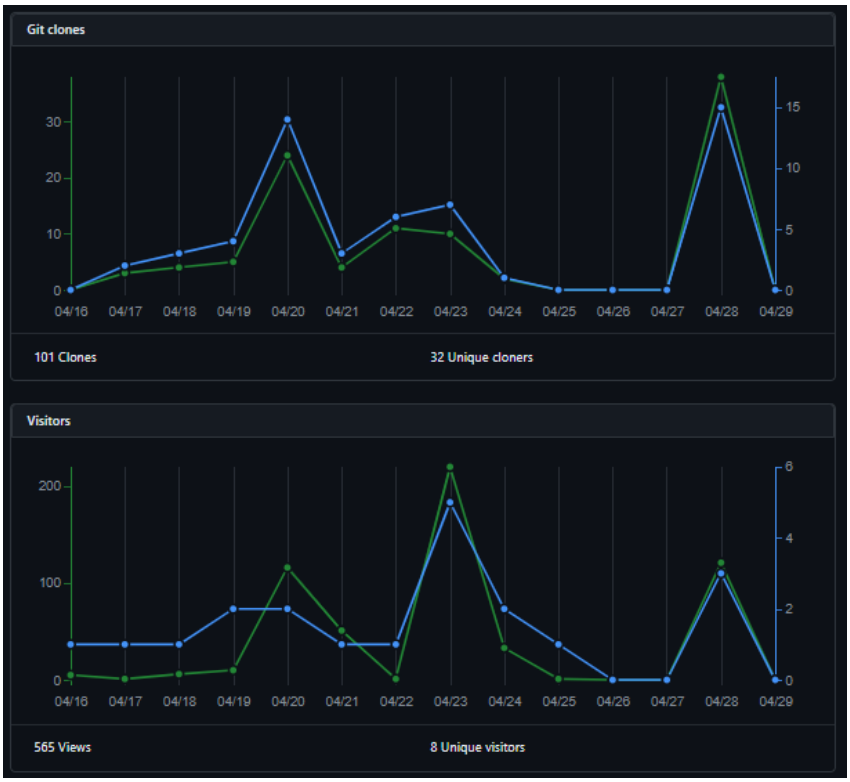
Sprint Burnup Report:



GitHub







Referring sites

Site	Views	Unique visitors
github.com	229	3
statics.teams.cdn.office.net	11	1
com.google.android.gm	5	1

Popular content

Content	Views	Unique visitors
mc16dn/COSC-4P02-Group-Assig...	64	4
COSC-4P02-Group-Assignment/RE...	54	3
COSC-4P02-Group-Assignment/D...	46	3
Editing COSC-4P02-Group-Assign...	23	2
COSC-4P02-Group-Assignment/vi...	18	3
COSC-4P02-Group-Assignment/M...	18	2
mc16dn/COSC-4P02-Group-Assig...	13	6
COSC-4P02-Group-Assignment/vi...	12	1
COSC-4P02-Group-Assignment/D...	10	1
Commits	10	1