GONZAGA UNIVERSITY
School of Engineering and Applied Science
Center for Engineering Design and Entrepreneurship

Wildfire Resource Simulator

Project Risks, Release, Management, and Maintenance Plan

Plan Sections 04-07

Release:
Draft v0.1

PROJECT PLAN DRAFT STAGE DOCUMENT
October 8th, 2023

The D Squad

Kevin Dang, Claire Yegian, Drew Bogdan, Vincent Do, Kole Davis

4 Project Risks

1. **API Outage(s)**
   a. If one of the APIs we use goes down, it could disrupt simulation data collection and cause certain project features to stop working.
   b. We can store the API data we collect in our own separate database and have our simulators pull from that system instead of directly pulling from the APIs.
   c. We can monitor API uptime and outage reports while developing project features.
   d. One or multiple API outages occurring would trigger our need to mitigate this risk.
   e. If one or multiple APIs we are using go down, our system will cease API data requests and instead use the info already pulled for the duration that the API(s) are down.
2. **Database Host Outage**
   a. If we have not backed up the data we have collected, we could lose our data and our simulations would be unable to make database connections and run.
   b. We can create local backups of the database to prevent data loss.
   c. We can monitor our database machine from time to time and keep an eye on outage reports if they are available.
   d. Our database host either being faulty or going down completely would trigger our need to rely on local backups.
   e. If our database host goes down, we will have data backups in place that we can feed into our simulators as a substitute.
3. **Unity Integration Version Control Errors**
   a. Caution must be exercised when using GitHub in tandem with the Unity game engine, as Unity is not natively compatible with GitHub and could break the project if not properly incorporated.
   b. We can put in place the proper environment structure Unity needs to be compatible with GitHub, as well as maintain a standard Unity environment version and gitignore file among the team. In addition, we can declare a branch we will use to build and run the project. We can also revert repository states if the situation calls for it as a last resort.
   c. We can monitor each other’s commits and pull requests to the repository and what Unity files will be changed in those commits.
   d. A loose commit could result in version control errors that we would need to address via reverting the repository branch(es) to a previous working state.
   e. If errors such as these are encountered, we can work as a team to fix the issue or revert the repository.

5 Initial Product Release Plan

5.1 Major Milestones

Our major milestones will be the completion of our Project Plan, amassing a workable amount of data, building a functional map, constructing a machine learning model to predict on our data, and finally integrating the ML model and the dynamic map. These milestones encompass our three major features: a database, an interactive map, and a ML model. We ordered them in this way because we need the plan to begin development, the data to begin working with the ML model, and both the ML model and the map prepared in order to integrate the two. The map and the ML model will be developed in parallel because they are fairly independent of each other until we begin integration.

Table 3: Major Milestones

Milestone	*Description*	*Target Completion Date*
Complete and Present Project Plan	We will have a fully revised and approved plan outlining our project, goals, and rough task timeline.	October 18th & 19th
Workable Amount of Cleaned Data	Cleaned data encompassing all 50 states for planes and fires with the minimum necessary metrics and ready to begin development with the ML model (not necessarily fully collected, but enough to start working with).	End of January
Functional Dynamic Map	A map displaying the US with the ability to represent fires, planes, and their changing locations over time.	Middle of February
Working Machine Learning Model	A ML model using our collected data to make predictions about fire outcomes (does not need to be our final, fully refined model, just something we can work with).	End of February
Integrated ML Model with Map	Two simulations, one with historical data and one with predictive data, both displayed dynamically on a timescale.	Beginning of April

5.2 Initial Sprint Releases

The following is our Sprint Release Plan for our project beginning after our final Project Plan is submitted and going until the final presentation of our project. Please note that sprints towards the end of the year are less precise than those beginning in a few weeks, and our plan may be adjusted as the project progresses. We will work on code commenting and basic documentation as we develop but will use our final sprint and the weeks before our presentation in May to clean up and add higher-level information to that documentation. We will also use that time for usability testing and deployment.

Table 4: Sprint Release Plan

Sprint Date	*Sprint Goal*	*Backlog*	*What we will demo*
4th Week in Oct to 1st week in Nov	Design and Create Database Graphics Start	Working on Establishing Graphics and Data Collection	The Database
2nd Week in Nov to 3rd Week in Nov	Start to Gather Data Basic Window	Fire Data	Data in database and window

4th Week in Nov to 1st Week in Dec	Continue to Gather Data Basic Map	Flight Data	Full Data in database and window with map.
4th Week in Jan to 5th Week in Jan	Clean and Refine Data Basic Map Functions	Working on Establishing Graphics & Fire Stats Display	Clean data in database and window with map and functions.
1st Week in Feb to 2nd Week in Feb	Begin Building and Training Model Start Map Integration	Working on Location Data & Basic Simulation	Machine Learning Model and beginning of map and integration
3rd Week in Feb to 4th Week in Feb	Refine Model Dynamic Map	Location Data	Integrated Map
1st Week in March to 3rd Week in March (includes Spring Break)	Working Model Complete Begin Data Integration into Map	Basic Simulation for Better Attack Vector	Advanced model for prediction and integrated map
4th Week in March to 1st Week in April	Refine model for large scale predictions Refine Map integration	Estimate difference in Size Based on Better Attack Vector and working on Fire Timeline	Large scale predictions and display on the map Somewhat combined integration
2nd Week in April to 3rd Week in April	Full integration of map and machine learning model into ML Useability testing, deployment and cleaning up documentation	Establishing Graphics Complete Fire Timeline Complete	Fully integrated Map and machine learning model
Final Push	Not a sprint, but a bit of time for tying up loose ends		Our final product

6 Maintenance Considerations

The initial build of this project will not require much maintenance overall. This is a static build of the program and will not be a live service, but future goals for this project include a live service dashboard. Because of this, there will be a large portion of future maintenance considerations to understand.

Due to this service not being a live dashboard for our project scope, there will not be much maintenance on the data end. The reason for this is because our plan originally will be to gather all the data in many API calls and store the information in our own database. Once we have this built, the maintenance would just be to keep the database up and running with consistent backups. However, on future additions to this project, important maintenance would need to be taken on to make sure the data coming in from the APIs is clean and merges well with the already existing database.

Other maintenance that would need to be considered for the future goals of the project would be keeping the map and details up to date and working well with the new data as it comes in live. This also includes the machine learning model that needs to be kept in check and working. With new data being added into the database, the ML model would need to be checked and retrained to understand the new data that would then be present for it to learn on.

7 Project Management Considerations

- Meeting Information
    - The team will be working in 2-week sprints to complete project deliverables, providing daily asynchronous stand-ups in the team Discord server.
    - Team meetings are held at least weekly on Mondays in BCSE 006.
    - Other weekly meetings:
        - Meetings with our sponsor are set on Wednesdays at 4:30pm PST.
        - Meetings with our advisor are set on Mondays at 6:00pm PST.
    - At the end of every sprint, we plan to update our advisor, DAB member, and sponsors with emails of the work that was done. If necessary, meetings can be held when implementations are not clear to the respective parties.
- Work Splitting
    - We conducted a work preferences survey for the team to determine what areas each member wanted to work on. The current work splits are as follows:
        - Front-End Development will involve working with designing the interface of the simulations.
            - Kole Davis
        - Back-End Development will involve data collection, data cleaning, and database management between the client and the server.
            - Vincent Do
            - Drew Bogdan
        - Full-Stack Development will involve both front- and back-end development of the simulators.
            - Kevin Dang
            - Claire Yegian
- Additional Tools
    - None as of now.