

Project Plan - Restaurant Recommendation System

This project is designed to develop a project plan for proposed software development for a group of restaurant owners in Marlborough, Massachusetts. It is described as a "consumer-focused recommendation system" that will consist of the following technologies:

- Alpine.js and Tailwind (Frontend)
- GraphQL (API)
- Go web and database server (Backend)
- Either PostgreSQL, EdgeDB, or PocketBase (Database)
- Cloud hosting (AWS, Azure, or GCP)

The software will ingest Yelp reviews of the participating restaurants. Restaurants will be updated monthly and Yelp reviews will be updated daily.

Part 1 - Problem Setup

Using a provided template for the project plan with required steps, for each step, I have filled out the required team members and the number of hours. I have also provided the hourly rate for each team member role and thus am able to calculate the budget based on these estimates. The project plan also specifies best case scenario, expected scenario and worst case scenario.

The project plan is provided in the data folder.

Uncertainties

- We have not been given a budget range by the client(s) so we are determining the budget based on a reasonable expectation for quality and budget
- This proposal does not assume any scope change and is based on the assumption that the client has a clear vision of the project. If the client changes the scope, the budget will need to be adjusted accordingly.
- We will need to discuss the ongoing costs of hosting and maintenance with the client to determine the ongoing costs of the project post-launch.

Additional Notes

- Several steps in the project can be done in parallel, which will help reduce the overall time to completion. These would be:
 - Developing pricing plan and developing implementation plan
 - Writing documentation and unit testing will be done in parallel with development
 - Developing marketing strategy and designing the brochure will be done in collaboration

As we will see below, this overall time to completion will be determined through critical path analysis using the values of each step.

Step 2 - Model Specification

An assumption provided is that all contributors to the project charge the same hourly rate. This means that less time for all members decreases the time by the same amount. This a minimization problem regarding cost but also a critical path analysis that will be determined by task dependency and overlap.

Normally we would have decision variables for individual team members but based on the instructions we will assume that all team members are the same hourly rate and there are no constraints on their availability. The decision variables are the number of hours for each task and when they are scheduled. This way we can determine an optimal schedule.

The important parts of determining the critical path are the latest start time and latest end time of each task as well as the tasks they are dependent on.

Time Constraints

The constraints are that the start time of each tasks has to be equal to or greater than 0, the sum of all of the task durations has to be equal time to the total project time.

Dependency Constraints

The dependency constraints are that the start time of a task is greater than or equal to the end time of the task it is dependent on.

In mathematically terms this would be:

- Let S_i be the start time of task i
- Let D_i be the duration of task i
- Let T be the total project time
- Let S_j be the start time of task j
- Let D_j be the duration of task j
- Let j be the task that task i is dependent on

The constraints are as follows:

1. Project completion time constraints: $T \geq S_i + D_i$
2. Precedence constraints: $S_i \geq S_j + D_j$ for all j

Step 3 - Programming

The code to both implement a critical path analysis solution to the linear programming problem and the output of the results to Gantt charts are contained within the Jupyter notebook in the Github repository for this project. I use PuLP to solve the linear programming problem and matplotlib to create the Gantt charts.

Step 4 - Solution

The solutions for each scenario are as follows:

Best-Case:

Total project completion time: 122 hours

Total project cost: 10440 dollars

Expected:

Total project completion time: 184 hours

Total project cost: 15720 dollars

Worst-Case:

Total project completion time: 246 hours

Total project cost: 21120 dollars

In the above code, I have run the solution for all three scenarios (best-case, expected, and worst-case). The code calculates the earliest start (ES) and finish (EF) times, latest start (LS) and finish (LF) times, slack, and critical path for each task. It also calculates the total project completion time and cost. The results are displayed for each scenario.

The critical path is the sequence of tasks that determines the total project completion time. Since certain tasks are dependent on other tasks, there are several paths through the project. For example, you can see that Task B (Develop marketing strategy) only has Task E relying on it, so theoretically it can have a much later start time than the other tasks (hence, it's ES value of 0 and its LS value of 144 in the expected scenario, for example).

To be more specific about criticality, the critical path items, as viewed in the Gantt charts are tasks that, if they are delayed or go over time, would delay the full completion of the process.

Step 5 - Overview

I have quite a bit of experience building software for clients (it has been my job for the past twelve years) so this process is quite familiar to me.

In my experience, clients almost always undervalue the effort it takes to build software and, due to this, not only underestimate budget but also. Additionally, items out of scope they feel are trivial tend to create conflict and eat into our teams overall budget.

I would likely present the worst-case scenario budget with an additional 15% added. I would make sure I'm making at least a 25% margin on what I pay the contractors. This way I have padding where I hopefully can deliver the project under the worst-case scenario and also have padding in my margin for any unexpected costs if things go way out of hand.

One of the major reasons for the caution is that there are a few critical parts of the project plan template that I feel are missing. The first is the UI/UX design phase. This would drive the requirements analysis and is often where the client is the most critical because they have a visual representation of what they are envisioning. This, in my experience, tends to be the most difficult part of any process and, if done correctly, sets the development team up for success or

failure. Although 'software design' is included in the template, I find this to be more about the architecture of the software and not the user experience.

The other part that is missing is the client testing and iteration. In order to gain approval from the client (and thus get paid in full), it is important that the client has enough time with the product to give meaningful notes and that the development team has enough time to iterate on those notes.