ASSIGNMENT 2: NETWORK MODELS—PROJECT MANAGEMENT

Alex McCorriston
MSDS 460: Decision Analytics
Northwestern University
Dr. Thomas Miller
April 21, 2024

1. Problem Setup

The initial step in this assignment was completing an Excel spreadsheet detailing the best-case, expected, and worst-case completion times for fifteen tasks necessary for developing a restaurant recommender system. The spreadsheet also includes information about required roles and their hourly costs. Roles include project manager, frontend developer, backend developer, data scientist, data engineer, and database administrator. Access to the completed spreadsheet is available in the GitHub repository under "Complete_Project_Plan.xlsx" (McCorriston 2024).

To create my task completion time estimates, I referenced a Forbes article about software development time estimation, which outlines three main stages: requirements and design, planning, architecture, and development, and software testing. According to the article, the first stage typically lasts between two to four weeks, the second stage between three to six months, and the third stage between three to six weeks (Kytainyk 2022).

Using this information, I sought assistance from Google Gemini, a large language model (LLM), to produce reasonable time estimates. My conversation with Gemini is stored in the GitHub repository as "Gemini_Conversation.txt" (McCorriston 2024). Given my limited experience in project management and software development, leveraging AI capabilities seemed like an effective approach for coming up with reasonable task completion estimates. Although Gemini's estimates carry inherent uncertainty, they aligned with the timeframes outlined in the Forbes article, which helped me validate their credibility. I also requested Gemini's estimates on the required number of roles for each task, which it provided.

For cost estimates, I found a list of the 2024 average annual salaries for the relevant roles in the United States (Built In 2024). I converted these salaries to hourly wages using a calculator, assuming 40 hours per work week and 52 work weeks per year (CalcXML n.d.). It is important

to note that there is uncertainty around these estimates due to a lack of information on data collection and analysis methods provided by the creators of this list.

Finally, a directed diagram for the project, created using draw.io is included below (Draw.io n.d.). This diagram shows that tasks A and B have no predecessors, so these can be completed in parallel. Additionally, once A is completed, tasks C and D1 can be completed in parallel. Once D1 is completed, D2 and D3 can be completed in parallel. Once D4 is completed, D5 and D6 can be completed in parallel. Once D8 is completed, F and G can be completed in parallel.

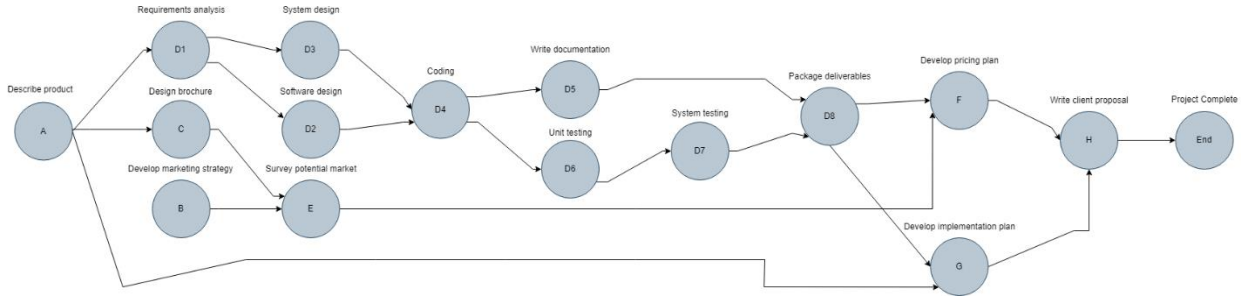

Figure 1. Project Directed Diagram.

2. Model Specification

The following setup is the standard form of the linear programming (LP) problem:

Decision Variables:

- S_i = start time for task i
- E_i = end time for task i

Objective Function:

- Minimize
  - $T = \Sigma E_i$
  - $T = \Sigma H_i \times R_j$ (where R_j represents the hourly rate for role j)

Subject to:

- $E_i = S_i + H_i$ for all tasks i (where H_i represents the best-case, expected, or worst-case time estimate for task i)
- $S_i > E_j$ for all tasks i, j where i is a successor of j

and

- $S_i, E_i > 0$ for all tasks i (non-negativity constraints)

By setting up the problem in this manner, two objectives are simultaneously addressed: minimizing time and minimizing cost. However, given the simplifying assumption that all contributors charge the same hourly rate, a minimum total time solution will inherently minimize total cost as well. Therefore, I will use a minimum-time objective for my problem setup.

3. Programming

To implement the LP problem, I used Python's PuLP package. The program code and output can be found in the GitHub repository under "Critical_Path_Analysis_Code.py" and "Critical_Path_Analysis_Solution.txt" respectively (McCorriston 2024). There are three different LP problems and solutions to represent the three different scenarios of best-case, expected, and worst-case.

4. Solution

With a minimum-time objective, the best-case scenario minimum total project time is 151 hours. The expected minimum total project time is 350 hours. Finally, the worst-case scenario minimum total project time is 700 hours. Regardless of the scenario, the critical path remains the same, consisting of tasks A → D1 → D2 → D4 → D6 → D7 → D8 → G → H. This is the critical path because it reflects the longest sequence of dependent tasks that must be completed to

execute a project. To help visualize the solutions, Gantt charts, created via Excel, are displayed below (Stratvert 2022).

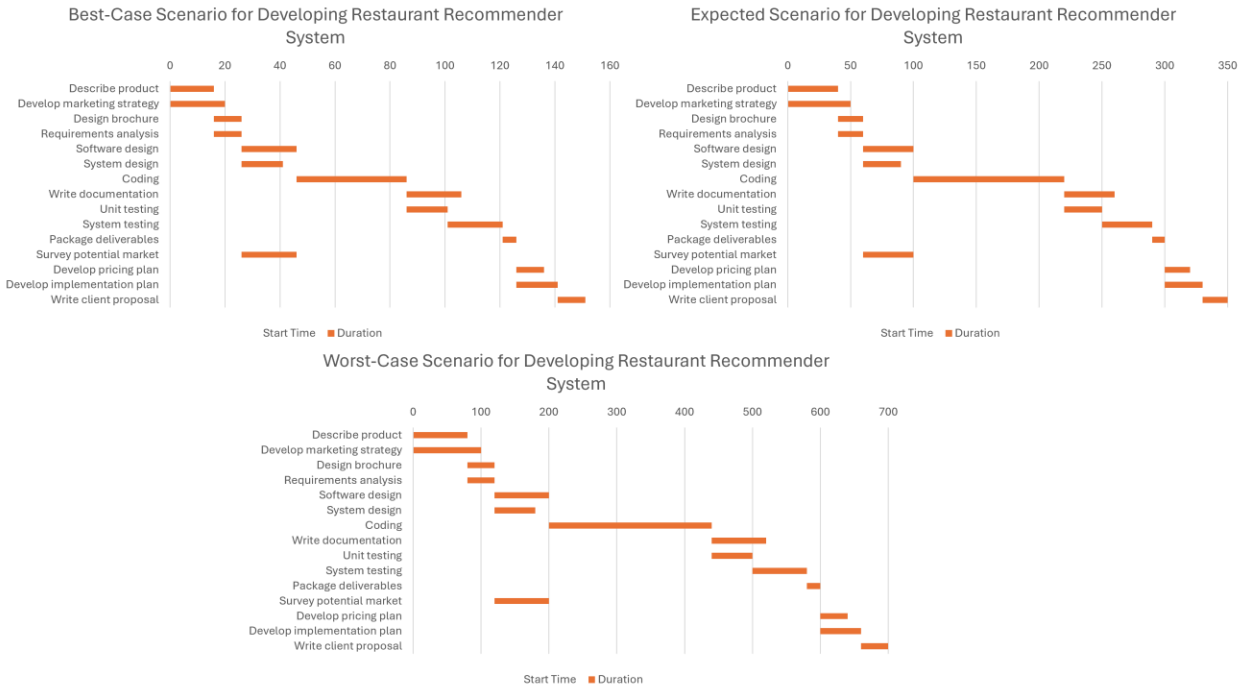

Figure 2. Gantt Charts for Best-Case Scenario (Left), Expected Scenario (Right), and Worst-Case Scenario (Bottom).

5. Overview (Mock Discussion with Prospective Client)

The project aims to develop an advanced restaurant recommender system tailored for the Marlborough, Massachusetts area, utilizing specific technology stacks for the frontend, backend, and database components. Alpine.js and Tailwind will power the frontend, while the backend will feature a GraphQL API and a Go web and database server. I will oversee the project from project management and data scientist roles. The project's complexity also requires assembling a skilled software development team, including frontend and backend developers, data engineers, and database administrators.

Pricing for the project depends on completion time, ranging from 151 hours in the best-case scenario to 350 hours in the expected scenario to 700 hours in the worst-case scenario. Consequently, costs vary accordingly, with projections of $45,704.42 for the best-case, $102,116.80 for the expected, and $204,233.60 for the worst-case scenarios. These calculated cost estimates can be found in the project GitHub repository under "Critical_Path_Analysis_Code.py" and "Total_Cost_Summary.txt" (McCorriston 2024). Delivery of the product prototype is estimated to span from a few weeks to several months, considering best-case to worst-case task competition time estimates. To mitigate potential delays and ensure timely completion, adding independent contractors may be beneficial. Their addition to the team could expedite delivery, especially if they fill the roles required for tasks along the project's critical path. This strategic allocation of resources will prioritize efforts where they are most needed, minimizing the risk of delays and ensuring project completion according to schedule. Thank you for your consideration of this project proposal.

References

Built In. 2024. “US Tech Salaries 2024 | Built In.” https://builtin.com/salaries.

CalcXML. n.d. “Convert my salary to an equivalent hourly wage.” Accessed on April 20, 2024. https://www.calcxml.com/calculators/convert-salary-to-hourly.

Draw.io. n.d. “draw.io - Free Flowchart Maker and Diagrams Online.” https://app.diagrams.net/.

Google. n.d. Accessed April 20, 2024. https://gemini.google.com/app.

Kytainyk, Vlad. 2022. “Software Development Time Estimation: How Long Should It Take to Develop a Product?” Forbes. https://www.forbes.com/sites/forbesbusinesscouncil/2022/12/02/software-development-time-estimation-how-long-should-it-take-to-develop-a-product/?sh=661dccb176ce.

McCorriston, Alex. 2024. "MSDS-460-Assignment-2-Project-Management." GitHub Repository. Accessed April 20, 2024. https://github.com/alexmc808/MSDS-460-Assignment-2-Project-Management.

Stratvert, Kevin. 2023. “How to Make Gantt Chart in Excel.” Youtube video, 7:43. https://www.youtube.com/watch?v=zC22yPmc6Kw.