SE 4485: Software Engineering Projects

Fall 2024

Project Management Plan

|  |  |
| --- | --- |
| Group Number | 11 |
| Project Title | Build a Design-of-experiments Interactive Oracle LLM App |
| Sponsoring Company | The Fellows Consulting Group (FCG) |
| Sponsor(s) | Tom Hill |
| Students | 1. Alberto Escobar  2. Dara Moheimani  3. Lee Rafael Filomeno  4. Brandon Bailey  5. Blanca Berrios Henriquez  6. Saishrey Bhandare |

ABSTRACT

The project management plan outlines the development of an Ontology-Generation Large Language Model (LLM) for the Fellows Consulting Group; the project is designed to streamline requirements elicitation through automation. The project will consist of two major components: the development of a local LLM application using Ollama and Llama 3.2, and a WordPress based testing platform. Using an agile development approach, and an iterative life cycle model, the project aims to deliver a minimum viable product by early May 2025. The final system is expected to reduce the time and effort required for ontology generation. The plan provides comprehensive coverage of organizational structure, development methodology, resource requirements, and the development schedule for the project.

TABLE OF CONTENTS

# Introduction........................................................................................................................................

# 1.1 Purpose and Scope.....................................................................................................................

# 1.2 Product Overview......................................................................................................................

# Project Organization..........................................................................................................................

# Lifecycle Model Used........................................................................................................................

# Risk Analysis.....................................................................................................................................

# 4.1 Model Performance Risk.............................................................................................................

# 4.2 Integration Complexity................................................................................................................

# 4.3 Requirements Evolution..............................................................................................................

# Software and Hardware resource requirements.................................................................................

# Deliverables and Schedule.................................................................................................................

# Monitoring, Reporting, and Controlling Mechanisms.......................................................................

# 7.2 Sprint Tracking and Reporting....................................................................................................

# 7.3 Integration Quality Control..........................................................................................................

# 7.4 Project Documentation and Tracking..........................................................................................

# Professional Standards.......................................................................................................................

# Configuration Management Evidence...............................................................................................

# Engineering Standards and Multiple Constraints .............................................................................

# Additional References.......................................................................................................................

# LIST OF FIGURES

# Figure 1: Project Timeline and Dependencies.........................................................................................

# Figure 2: Activity network..........................................................................................................................

# LIST OF TABLES

# Table 1: Team Roles and Responsibilities................................................................................................

# Table 2: Project Risks and Mitigation Strategies.....................................................................................

# Table 3: Activity network key....................................................................................................................

# Table 4: Project Deliverables Schedule ...................................................................................................

INTRODUCTION

**Purpose and Scope**

The document’s main purpose is to define and communicate the approach that will be taken to complete the project. The document will serve as a reference for any questions regarding organizational structure, resource allocation, and timeline management. The document encompasses all aspects of project development starting from the planning stage all the way until the final version of the Ontology-Generation LLM and the demo website.

**Product Overview**

The Ontology Generation LLM application will address a fundamental challenge in software engineering, the creation of domain ontologies during the requirements elicitation phase. Currently, when a team wants to work on a project in an unknown domain a significant amount of manual effort must go into making an ontology. With our solution we hope to automate the ontology creation process and thus make it easier for engineers to work in new domains.

# PROJECT ORGANIZATION

The development team is structured into two focused groups to ensure efficient collaboration and clear accountability for project deliverables. Each team has distinct responsibilities, but they will maintain close coordination throughout the project lifecycle. Team member placement was carefully considered based on individual strengths and technical expertise. The structure allows for flexibility, enabling team members to assist across teams, when necessary, particularly during critical integration phases. This organizational approach will help reduce unnecessary overhead and allow the team to deliver a robust ontology generation LLM.

|  |  |
| --- | --- |
| **Website** | **LLM / LLM Integration** |
| Saishrey Bhandare | Alberto Escobar |
| Blanca Berrios Henriquez | Lee Rafael Filomeno |
| Brandon Bailey | Dara Moheimani |

**Table 1: Describes what sub-team each member is in**

# LIFECYCLE MODEL USED

The project will utilize an Agile iterative development lifecycle model. This model will support parallel development of both the WordPress web site and the Ontology-Generation LLM components. An iterative approach will allow for continuous integration, and adaptability to potential risks/requirements. The teams will work in two-week sprints, with each team maintaining autonomy over their development process.

At the end of each sprint teams will conduct integration testing to ensure compatibility between the website and LLM components. If issues do arise teams can coordinate additional testing as needed. Weekly meetings (outside of sponsor meetings) will be held where team members can raise concerns, discuss technical challenges, and align on integration requirements. Overall, with our lightweight model we are hoping to reduce unnecessary overhead while still ensuring regular communication and having regular integration points. The approach can also evolve based on team feedback and as project needs evolve.

# RISK ANALYSIS

# **Model Performance Risk** – The LLM may fail to generate sufficiently accurate ontologies, particularly during initial development. Mitigation for this will include maintaining evaluation metrics, and a stable version of the LLM in case rollbacks are needed for underperforming versions.

# **Integration Complexity –** The parallel development of the website and LLM may present integration challenges. Regular testing at sprint boundaries and clear interface specifications (data format, error handling procedures, etc.) will be kept between teams.

# **Requirements Evolution** – As the team gains a better understanding of the system’s capabilities requirements may evolve. The agile approach being taken will allow for adjustments to be made between sprints.

|  |  |  |  |
| --- | --- | --- | --- |
| RISK | LIKELIHOOD | IMPACT | MITIGATION |
| **Model Performance Risk** | Medium | High | Monitor evaluation metrics and maintain a stable version of the LLM. |
| **Integration Complexity** | High | High | Regular integration testing held at sprint boundaries. |
| **Requirements Evolution** | High | Medium | Taking an agile approach to the project will help deal with changing requirements. |

# **Table 2: Describes the possible risks and different mitigation strategies**

# SOFTWARE AND HARDWARE RESOURCE REQUIREMENTS

**Hardware Requirements**

Workstation – 16 GB RAM (32 GB preferred for LLM development team), 500 GB of

storage, NVIDIA GPU with 8GB of VRAM preferred for LLM development team.

**Software Requirements**

**LLM Development** – Ollama framework, Pytorch and Transformers libraries.

**WordPress Development** – MySQL 5.7+, Local WordPress environment.

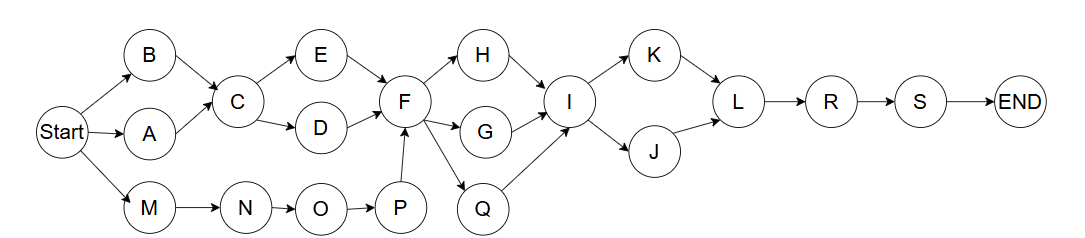
GitHub will be used for version control for both development teams.

# DELIVERABLES AND SCHEDULE

# Currently the program itself (the LLM and Website) is expected to be done in 4 2-week sprints, and each sprint will incrementally add functionality. The end of the first sprint will lead to a basic website linked with an LLM. The 2nd sprint will aim to deliver basic ontology generation and a more polished website. The 3rd sprint will deliver an LLM with more advanced ontology generation and the final version of the website. The 4th, and final sprint, will involve working on the MSI package and completing testing on the WordPress site. Activities are divided equally among the groups, and documentation (deliverables for the class) will be split equally among members.

|  |  |
| --- | --- |
| * Setup Ollama & llama 3.2 | K. MSI package |
| * Setup website with basic functionalities | L. 4th integration point |
| * 1st integration point | M. Project Management Plan |
| * Stanford NLP analysis and design | N. Requirements Documentation |
| * Basic ontology generation | O. Architecture Documentation |
| * 2nd integration point | P. Detailed Design Documentation |
| * Demo interface | Q. Test Plan |
| * Enhanced Generation | R. Final Project Presentation Slides and Demo |
| * 3rd integration point | S. Final Project Report |
| J. Testing Infrastructure |  |

**Table 3: Key for the Network Diagram Below**

**Figure 2: Network diagram representing project activities and their dependencies**

|  |  |
| --- | --- |
| **Deliverable** | **Due Date** |
| Project Management Plan | February 7 |
| Requirements Documentation | February 21 |
| Architecture Documentation | March 21 |
| Detailed Design Documentation | April 4 |
| Test Plan | April 18 |
| Final Project Presentation Slides and Demo | May 3 |
| Final Project Report | May 9 |

**Table 4:** Outlines the specific documentation due dates

# MONITORING, REPORTING, AND CONTROLLING MECHANISMS

# **Sprint Tracking and Reporting**

# After each two-week sprint a sprint review meeting will be held where both teams present their progress reports. The reports will document completed user stories, any challenges that were faced, and metrics based on the component. For the website team, this would include completion of certain functionalities and integration milestones. For the LLM team, this would include model performance metrics and training progress.

# **Integration Quality Control**

# Regular integration testing will be held at sprint boundaries where reports detailing test coverage, performance metrics, and compatibility issues will be addressed.

# **Project Documentation and Tracking**

# Technical Documentation, such as api specs, will be maintained under the projects GitHub repository. The repository will also house the centralized STATUS.md file which will track milestone completions, upcoming deadlines, and any other relevant information. This will help ensure that all members, and the sponsor, are able to follow along with the team's progress.

# PROFESSIONAL STANDARDS

On the first occurrence of unacceptable behavior, determine the circumstances involved, resolve the problem, and document the event in the meeting minutes.

On a second occurrence, notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem.

On a third occurrence, again notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem. At this point, the team will have the *option* of removing the team member. If removed, then the team member receives a pro-rated grade based on the number of weeks they have participated in the group.

Examples of unacceptable behavior may include not delivering on time, delivering poor quality work, missing team meetings, being unprepared for team meetings, disrespectful or rude behavior, etc. Reasons such as “too busy” or “I forgot”, or “my dog ate my design model” are unacceptable.

Valid reasons that must be considered include those listed for obtaining an incomplete standing in a course (illness, death in the family, travel for business or academic reasons, etc.)

# EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT

# GitHub will be the chosen CM tool for our group. The repository has already been set up along with a /docs folder that will contain all documentation related to the project.

ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS

Standards:

* IEEE Std 1058-1998 for Software Project Management Plans
* PMBOK® Guide: Project Management Body of Knowledge
* IEEE Std 12207: Software Life Cycle Processes

Constraints:

* Professional formatting with consistent headings
* Defined technical terms and abbreviations
* Proper citations and references
* Version control requirements
* Adherence to document templates

ADDITIONAL REFERENCES

[1] IEEE Standard for Software Project Management Plans, IEEE Std 1058-1998, 1998.

[2] Project Management Institute, "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)", 7th Edition, 2021.

[3] IEEE Standard for Software Life Cycle Processes, IEEE Std 12207-2017, 2017.