SE 4485: Software Engineering Project

Fall 2024

Project Management Plan

|  |  |
| --- | --- |
| Group Number | 3 |
| Project Title | Knowledge Management Assistant (Team B) |
| Sponsoring Company | The Fellows Consulting Group (FCG) |
| Sponsor(s) | Jeff Buchmiller |
| Students | 1. Blythe Williams  2. Nidhi Prakuzhy  3. Roj Pawig  4. Ashley Primrose  5. Humayl Sheryar  6. Dalton Brua |

ABSTRACT

This document gives a detailed outline of a comprehensive plan for our senior project focused on developing a knowledge management assistant application. Which will include sections on project organization, lifecycle models, risk analysis, resource requirements, deliverables, and monitoring mechanisms.

The goal for this project is to create software that efficiently searches and manages information. Within this document, it is stated that our team will adopt an agile lifecycle model to accommodate continuous improvement. It is also stated in risk analysis, the multiple issues that could arise during the progression of this project as well as mitigation strategies to counteract each risk.

Along with the specifications of the hardware and software requirements, including the operating systems, and etc., this document also has a layout of a schedule of deliverables with the hours and responsibilities listed for each one.

There is also a section that references monitoring and reporting mechanisms to help emphasize continuous communication and time management, which will help encourage progress. As well as professional standards and guidelines to ensure high-quality outcomes.

Overall, this document provides a structured and detailed approach to project management, risk mitigation, ensuring successful development, and deployment of the knowledge management assistant application.

Table of Contents

[LIST OF FIGURES 2](#_Toc1545806417)

[LIST OF TABLES 2](#_Toc1644004437)

[PROJECT ORGANIZATION ~ Humayl 2](#_Toc1083813788)

[LIFECYCLE MODEL USED 2](#_Toc326283982)

[RISK ANALYSIS ~ Blythe 2](#_Toc1178417478)

[SOFTWARE AND HARDWARE RESOURCE REQUIREMENTS ~ Ashley 2](#_Toc1671022119)

[DELIVERABLES AND SCHEDULE ~ Dalton 2](#_Toc329615343)

[The deliverables are as follows: 2](#_Toc942431881)

[MONITORING, REPORTING, AND CONTROLLING MECHANISMS ~ Roj 2](#_Toc1415980396)

[PROFESSIONAL STANDARDS ~ Nidhi 2](#_Toc27319867)

[EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT 2](#_Toc1407048511)

INTRODUCTION

* This document gives an in-depth overview of our senior project to create a knowledge management assistant application. The layout of this document will include the organization, models, requirements, scheduling, standards, etc. This document will provide our sponsor with information as we progress in our project, so that the company itself will use these details to continue where we left off once this project concludes.
* The purpose of this document is to provide guidance and direction as well as communication and providing risk management. The scope of this document is to also provide goals, objectives, scheduling, management, roles, responsibilities, and closure.
* A brief overview of the knowledge management assistant application is our software will search keywords in each search request in a timely manner. Then, the software will manage the responses to maximize the reuse of information it has previously collected. This will be executed through a web-based UI, with the possibility of Apple IOS and Android UI as well.
* Given this introduction, the following structure of the document will give you insight into our project organization, life cycle models we used, risk analysis, resource requirements for both hardware and software, the deliverables and schedule, monitoring/reporting mechanisms, and lastly, our standards.

# PROJECT ORGANIZATION

* describe the way in which the development team is organized, the people involved, and their roles on the project
* include the rationale
* The development team will consist of a Project Manager, Front End developers, Back End Developers, and QA based off interests and experiences of the group members.
* Sponsor- Jeff, Team Leader- Blythe, Group Members- Nidhi, Dalton, Roj, Ashley, Humayl

# LIFECYCLE MODEL USED

* describe the lifecycle model used
* include the rationale
* AGILE

# RISK ANALYSIS ~

* **Project risks:**
  + Performance
    - There is the risk that the program of the application will run into performance degradation. Meaning the possibility of the software processing slower response times or having difficulty taking in a multitude of requests of different sizes.
    - Likelihood: There is a moderate chance of this risk occurring.
* Risk reduction: To avoid having performance issues, we can create a strategy to handle all possible scenarios that might affect the runtime of the program.
* Mitigation: We can optimize our data.
  + - Rationale: Design the program with scalability in mind.
  + Timeline and Constraints
    - There is the risk that the project will face delays due to unforeseen challenges or resource limitations. Whether we run out of time to complete a finished project within the timeline or limited access to necessary tools.
    - Likelihood: There is a moderate chance of this risk occurring.
* Risk reduction: Our team can break the project down into manageable tasks throughout each meeting, along with consulting with our sponsor with further questions. In doing so, we can slowly progress with our project to come to a finished product within the timeline.
* Mitigation: Using agile methodologies and regularly assessing our timeline. Keeping the scope realistic.
  + - Rationale: Regularly assess the project scope and schedule.
  + User engagement
    - There is the risk that the design does not meet the user’s expectations or the chance that the application itself is difficult to use for new users.
    - Likelihood: There is a low to moderate chance of this risk occurring.
* Risk reduction: Our team will take priority in creating this application with an appealing and easy to use UI. Considering any scenarios that a future user might have issues with and creating a solution throughout the planning process.
* Mitigation: Conducting early user testing as well as gathering feedback to ensure a more intuitive design.
* Rationale: Engaging our sponsor in the early stages of software development. As well as testing the application to align with the user’s expectations and needs.
* Team skill gaps
* There is a risk that some of the members on the team may lack or not have much experience in the skills needed to develop the project.
* Likelihood: There is a low to moderate chance of this risk occurring.
* Mitigation: Invest time in practicing software related skills for the project outside of class if a member of the team feels that they need it.
* Rationale: Building skills needed will help ensure that our team can handle the complex aspects of this project.
* Inadequate Testing
* There is a risk that we will run into failures identifying bugs due to not testing accurately.
* Likelihood: There is a moderate to high chance of this risk occurring.
* Mitigation: We can implement rigorous manual testing, as well as ensuring that we test at every phase of our project.
* Rationale: Testing will help to reduce the risk of creating a final product with bugs or performance issues.
* Hardware/Software Incompatibility
* There is a risk that there could be compatibility issues with different software versions or certain devices.
* Likelihood: There is a low to moderate chance of this risk occurring.
* Mitigation: We can test the application across different devices and platforms to ensure that the application is useable across different versions of software or devices.
* Rationale: Ensuring compatibility and avoiding the chance of alienating users.
* Inadequate Documentation
* There is the risk that one of the members or us as a group could have poor internal or external documentation. Which could lead to user confusion when it comes to the application.
* Likelihood: There is a low to moderate chance of this risk occurring.
* Mitigation: We can keep thorough internal documentation and possibly create user-friendly guides or FAQs
* Rationale: Proper documentation will help our team stay organized and assist the users in understanding our application.
* Scope Creep
* There is the risk that our project scope is continually increasing beyond its original plan.
* Likelihood: There is a high chance of this risk occurring.
* Mitigation: We need to clearly define the scope and obtain the stakeholder agreement. Also, continually go over the scope and prioritize features that fall within said scope.
* Rationale: Managing the scope from the start will prevent our application from becoming overwhelming and it will help our team meet deadlines.
* Feature Overload
* There is the risk that our team will get too ambitious and add too many features which could overwhelm the users.
* Likelihood: There is a moderate chance of this risk occurring.
* Mitigation: We can prioritize the must-need features.
* Rationale: Focusing on the core functions of the application will make it easier to use and more focused on the goal of the application and user needs.
* User data loss
* There is the risk that our application may accidentally lose stored information, either to user error or technical malfunction.
* Likelihood: There is a low to moderate chance of this risk occurring.
* Mitigation: We can implement a data recovery option or a backup system.
* Rationale: Data reliability is something that will ensure user trust and continued use.

# SOFTWARE AND HARDWARE RESOURCE REQUIREMENTS

Software requirements:

* Operating System
  + Windows 10/11 or macOS 10.15 and above.
* Development Environment
  + Integrated Development Environment (IDE) - Visual Studio Code
* Programming Languages
  + Python 3.8 or later (tentative)
  + JavaScript (Node.js 14.0 or later)
* Web Frameworks and Libraries
  + Flask or Django?
  + ReactJS
* Database
  + PostgreSQL
* Cloud Platform?
  + AWS or Google Cloud
* Version Control
  + Git
* Additional tools
  + Docker

Hardware Requirements:

* Development Workstations
* Server Hardware (if local testing is required)
* Networking Equipment
* Peripheral Devices

Rationale

# DELIVERABLES AND SCHEDULE

# The deliverables are as follows:

* **September 6, Project Management Plan**
  + This deliverable is expected to take ~8 hours to complete, split up between the 6 members of the group.
    - Each member of the group was assigned one or two points on the project management document, and each member should spend 1-2 hours working on this document.
  + The activities of this deliverable include:
    - Documenting the plans for the management of our project
    - Getting the sponsor’s feedback and changing any aspect that seems unreasonable
* **September 20, Requirements Documentation**
  + This deliverable is expected to take ~20 hours to complete, split up between the 6 members of the group
    - Most of the time in this phase will be used to talk with the sponsor to determine what the project should look like, and the overarching ideas about the project.
    - The rest of the time will be writing it all down, gathering and combining the ideas, and producing a professional and readable requirements document.
  + The activities of this deliverable include:
    - Talking with the sponsor and determining the exact specifications for the application
    - Determining which platform the application will operate on.
    - Creating prototype user-interfaces.
    - Documenting all requirements composed for the project
* **October 18, Architecture Documentation**
  + This deliverable is expected to take ~30 hours to complete, split up between the 6 members of the group
    - Architecture is a very technical portion of the project and is often difficult to create efficiently. Many aspects of architecture require a very good understanding of the project and the many tools and platforms the project may use. Therefore, this portion of the project will require research to be done, as well as setting up the environments for the impending parts of the project.
    - Assuming we can split up most of the work between the members, each student should be expected to spend 4-6 hours researching and documenting their contributions.
  + The activities of this deliverable include:
    - Researching platform environments
    - Setting up environments for the planned platform(s)
    - Setting up version control and coding environments
    - Determining language(s), database(s), and protocol(s) that will be used
    - Documenting all architecture specifications
* **November 1, Detailed Design Documentation**
  + This deliverable is expected to take ~20 hours to complete, split up between the 6 members of the group
    - The detailed design documentation will outline the entire project and the way in which we will approach creating it. This should be extremely well documented and as such will require different diagrams and explanations.
    - We will assign one or more diagrams to each member, and we will allocate 3-4 hours for each member to make their diagrams and explanations.
  + The activities for this deliverable include:
    - Creating system-wide vs individual component diagrams
    - Creating class diagrams
    - Creating sequence and use-case diagrams
    - Writing documents describing each diagram and its use
* **November 15, Test Plan**
  + This deliverable is expected to take ~15 hours to complete, split up between the 6 members of the group
    - The test plan must be a very detailed document that takes a lot of time to design. This is because we must think of all the ways we can break our project and try to fix those problems. Because of this, we have allocated 2-3 hours for each member to think about the project and come up with tests for all the functionality of the project.
    - The last few hours will be for a member to collect all the tests and test ideas and compile them into one document which we can use as our test plan.
  + The activities of this deliverable include:
    - Designing tests
    - Documenting all tests and test ideas
* **November 30, Final Project Presentation**
  + This deliverable is expected to take ~10 hours to complete, split up between the 6 members of the group
    - The final project presentation is very important to the project’s success. The project may work perfectly and to all specifications, however if we cannot explain and demo the project well, no one will know.
    - Therefore, each member will work on a specific part of the presentation, then every member will check the others’ parts of the presentation to ensure it is of the highest quality. We will allocate each member 1-2 hours to ensure their slides are accurate and check the group’s work.
  + The activities of this deliverable include:
    - Preparing presentation slides
    - Preparing project demo
    - Practicing the presentation and demo
    - Gathering feedback from the sponsor on our presentation
* **December 2, Final Project Report**
  + This deliverable is expected to take ~20 hours to complete, split up between the 6 members of the group
    - The final project report is the most important document of the project. This means our group must ensure that this document is of a professional standard.
    - The work associated with this portion of the project includes mainly gathering information, however we need to collect all documents, code, and diagrams to be outlined in the last report. This will take a lot of time to collect and organize, so each member will be given around 4 hours to gather any information they can find.
  + The activities of this deliverable include:
    - Gathering and collecting all information used and created during the project
    - Organizing information gathered into one cohesive and understandable document.

# MONITORING, REPORTING, AND CONTROLLING MECHANISMS

* Monitoring, reporting, and controlling are essential processes that must be carried out for the successful execution of any project. These processes include tracking and assessing the project’s progress, identifying any areas where adjustments to the settled plan and schedule are needed, and implementing the proposed changes as needed. Accordingly, corresponding reports on these important processes will be made to ensure satisfactory results, reduce project risks, and stay on scope.
* Status Reports**:** Continuous communication among project team members is expected as well as engagement with the sponsor, requiring members to provide status reports on current project tasks and/or performance. Monitoring the project through status reports provides necessary insight into how the team and stakeholders are satisfied with the quality and performance while also identifying any problem areas that may require more attention. Status reports are expected by team members orally or written through online communication during sponsor meetings and near important deadlines (See Deliverables and Schedule). Status reports must include one’s current standing on their assigned task and should also contain any other concerns. Although it is a critical process, status reports can and will be typically informal (due to time constraints and other responsibilities) as it is focused on members being transparent about their personal and project concerns, allowing the team to understand each other and be proactive on any issues that arise.
* Change Management: In tandem with status reports, managing necessary changes to the project is a natural and expected process that all project members are responsible for due to the nature of how projects are executed. Therefore, it is significant to have an appropriate procedure to manage change. As per the course syllabus, the ability to use a configuration management system and develop CM processes is expected for students to learn and be able to execute during this course. Configuration management tools are essential for tracking deliverables and synchronizing any changes made by multiple team members. CM tools that will be used include cloud storage (OneDrive/Google Drive), word processing software (Word/Google Docs), and code hosting platforms (GitHub). These tools allow the project team to see different versions when changes to a document are made, providing an important control measure for reviewing changes and comparing differences between two different versions. Accordingly, when making changes it is up to the members’ discretion to do it before discussing with the team as some changes are miniscule in terms of importance, e.g. fixing grammar mistakes. *However*, project members are expected to discuss and propose any **significant** changes with other team members, possibly requiring feedback from the sponsor. Changes made to the project not only include document and code changes but also possible scheduling changes that may need to be arranged for certain team members. Similar to status reports, formal change management reports are not required due to the primary focus on executing project deliverables and objectives, where formal change reports could induce time constraints and increase project risks. Since CM tools will be used, changes will be tracked, easily identifying the member who made a change to the document or source code. Simple comments can be made to further communicate minute changes to other team members. Moreover, when proposing more significant changes, members must make note of the change and discuss amongst the team, which may result in an impromptu meeting. It is of utmost importance that major changes are also communicated to the sponsor through email or other forms of communication. As previously underlined, communication in any shape or form is essential for the success of the project, making change management a more streamlined process that would not give rise to any more risks. Overall, management reports regarding changes are not expected *formally*, but members are expected to discuss, communicate, and be transparent about changes made/proposed to any aspect of the project.
* Time Management: Time management is an encompassing project control that covers not only general time management but also schedule management and resource management, important measures that ensures satisfactory project performance To ensure the project remains on track and on scope, it is essential to produce a project schedule report at the start of the project (See Deliverables and Schedule), which would be used as an outline to manage the team’s time and schedule. This report helps members identify possible risks and delays early, allowing for corrective actions to be taken. Additionally, it is important that weekly schedule reports are made to ensure everyone is on the same page, minimizing any possible miscommunication and misunderstanding about what tasks must be finished and what deliverable is due. Again, these reports are not required to be formal documents prepared by the team but must simply be conveyed through accepted forms of communication. Unitedly, resource management is necessary for appropriate time management. Managing and monitoring project resources, otherwise known as team members, are supervised by the project leader under the guidance of the project sponsor. The team, as a whole, is responsible for managing who is assigned what task and when it should be done. During weekly meetings, resource management should always be discussed, referring to the assignment of sub deliverables to specific team members. Reports regarding resource management are not required, but like other forms of project controls, they must be appropriately discussed by the team during sponsor meetings or through online communication. Provided that this course is about 16 weeks (about 3 and a half months), time management is ***crucial*** for the success of the project. It all comes down to competent communication, managing each members’ time appropiately.

# PROFESSIONAL STANDARDS

* Students are expected to act with a high-level of responsibility and academic honesty, demonstrating a high standard of individual honor in scholastic work as per the University’s policies. Regular class participation and engagement in group activities is expected, as explained in the course syllabus.
* refer to Appendix A for more details

# EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT

# A screenshot of a computer Description automatically generated

ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS

* IEEE Std 1058-1998: Software Project Management Plans
* PMBOK® Guide: Project Management Body of Knowledge
* IEEE Std 12207: Software Life Cycle Processes
* IEEE Std 15939: Measurement Process
* ISO/IEC/IEEE Std 29148-2018: Systems and Software Engineering
* Life Cycle Processes
* Requirements Engineering

ADDITIONAL REFERENCES

* Larson, E. and Gray, C., 2014. Project Management: The Managerial Process. McGraw Hill
* Humphrey, W.S. and Thomas, W.R., 2010. Reflections on Management: How to Manage Your Software Projects, Your Teams, Your Boss, and Yourself. Pearson Education

**Appendix A.**

The following provides a professional standards guideline for the teams. This guideline may be tailored. The professional standards must be agreed upon by each member in the team.

Guideline:

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.)