AI Model for Detecting Abnormal Behavior

System Quality Assurance Plan (SQAP)

GROUP 1-C

Name	Position	Email	Phone No
Tran Quoc Dung	Software Development	dungtqswh00420@fpt.edu.vn	0943910306
Pham Hoang Duong	Data Research	duongphswh00843@fpt.edu.vn	0902162467
Nguyen Thai Son	Data Research	sonntswh00509@fpt.edu.vn	0902950403
Duong Quoc Trung	AI Researcher	trungdqswh00902@fpt.edu.vn	0962579876
Do Tuan Dat	Software Development	datdtswh00592@fpt.edu.vn	0865411803

COS40005, Computing Technology Project A, May-2024, 03/06/2024

Review history

Version	Date	Author	Comments
1.00	04/06/2024	All Authors	Created initial draft document.
1.10	11/06/2024	Dung, Duong, Son	Updating SQAP based on clients' feedback.
1.20	15/06/2024	Dung	Finalizing the Software Quality Assurance Plan.

Acronyms/Abbreviations

ASAP As Soon as Possible

DMS Data Management System

GUI Graphical User Interface

IEEE Institute of Electrical and Electronics Engineers

Ver. Version

SQAP Software Quality Assurance Plan

SRS. Software Requirement Systems

SVN Subversion

Table of Contents

Chapter 1: Introduction

1.1 Author List/Roles

Author	Student ID	Role Semester 1	Role Semester 2
Tran Quoc Dung	103803891	Team Leader/ Software Developer	Team Leader/ Testing Champion
Pham Hoang Duong	103843282	Data Researcher/ Documenter	Documentation Champion
Nguyen Thai Son	103806531	Data Researcher/ Documenter	Documentation Champion
Duong Quoc Trung	103843321	AI Researcher/ Code Champion	Code Champion
Do Tuan Dat	103804603	AI Researcher/ Code Champion	Code Champion
Mr. Khang		Mentor/ Client	N.A (No longer part of Team)

1.2 Purpose

This System Quality Assurance Plan (SQAP) is a key document for the project “AI Model for Detecting Abnormal Behavior”. The project aims to develop a robust artificial intelligence model capable of identifying and reporting behaviors that detected from defined norms, which is crucial in various applications such as security surveillance and healthcare monitoring .This document outlines the policies and procedures that members of Group 1-C will follow to achieve an overall high standard of quality for Project Eagle, a Data Management System (DMS) for the Defence Materiel Organization (DMO). All team members are expected to adhere to the processes outlined in this document.

Chapter 2: Reference Documents

- IEEE Standards: Including IEEE 730 for software quality assurance plans.
- ISO/IEC Standards: Such as ISO/IEC 25010 for software quality requirements and evaluation.
- Quality Assurance Institute (QAI): Best practices and guidelines for QA processes.
- Project Management Institute (PMI): PMBOK Guide for project management best practices.
- Software Engineering Institute (SEI): CMMI (Capability Maturity Model Integration) for process improvement.

Chapter 3: Management

3.1. Organization/Roles

3.1.1 Meeting Roles

- **Chair (Team Leader)**

- Responsibilities: The Chair leads the team, sets meeting agendas, ensures that the project stays on track, and that all team members are contributing effectively. They also act as the primary point of contact between the team and any stakeholders.

- Key Tasks: Facilitating meetings, delegating tasks, resolving conflicts, and maintaining overall project vision.

- **Client (Mr. Khang)**

- Responsibilities: The Client represents the interests of the end-users or stakeholders for whom the AI model is being developed. They provide requirements, feedback, and approval at various stages of the project.

- Key Tasks: Communicating requirements, providing feedback on deliverables, and ensuring the final product meets their needs.

- **Coding Champion - Software Developer**

- Responsibilities: The Coding Champion is responsible for overseeing the development of the AI model. They ensure coding standards are met and that the codebase remains clean and maintainable.

- Key Tasks: Writing code, reviewing code submissions, maintaining coding standards, and integrating various components of the AI model.

- **Documenter**

- Responsibilities: The Documenter oversees creating and maintaining all project documentation. This includes the SQAP, technical documentation, meeting minutes, and any other necessary reports.

- Key Tasks: Writing and updating documentation, ensuring clarity and completeness, and keeping records organized.

3.1.2 Formal Review Meeting Roles

- **Client**

- Responsibilities: The Client evaluates the deliverables to ensure they meet the requirements and provides feedback from the perspective of the end-user or stakeholder.
- Key Tasks: Reviewing deliverables, providing feedback, and making decisions on approval.

- **Documenter**

- Responsibilities: The Documenter records all the findings, discussions, and decisions made during the review meeting.
- Key Tasks: Taking detailed notes, ensuring all points are captured accurately, and distributing minutes after the meeting.

- **Inspector**

- Responsibilities: The Inspector critically examines the deliverables for defects and adherence to specifications and standards.
- Key Tasks: Identifying issues, suggesting improvements, and verifying that corrections are implemented.

- **Author**

- Responsibilities: The Author presents the work to be reviewed and answers any questions regarding their work.
- Key Tasks: Explaining design choices, justifying approaches taken, and responding to feedback.

- **Reader**

- Responsibilities: The Reader represents a fresh set of eyes, often reading through the deliverables aloud to identify any discrepancies or areas that lack clarity.
- Key Tasks: Reading documents out loud, identifying unclear sections, and helping to interpret complex information.

3.1.3 Champion Roles

In our project, the Champion Roles are pivotal to maintaining high standards of quality across all facets of development. Team Leader, who commonly gathers teammates together, is responsible for ensuring the group works effectively together, with a view to accomplishing the aiming product of the project. The team leader also observes the work logs throughout the project's processes, with a view to guaranteeing the productivity of individuals. They also compose status reports where appropriate and attend team leader meetings.

The Documentation Champion ensures that all project documentation is accurate, up-to-date, and adheres to the best practices in technical writing. This role is crucial for maintaining clear and consistent communication within the team and with external stakeholders.

The Code Champion takes ownership of the codebase, enforcing coding standards, reviewing code for quality and efficiency, and mentoring team members to foster a culture of excellence in programming.

Lastly, the Testing Champion leads the charge in quality assurance by designing comprehensive test plans, executing tests rigorously, and advocating for continuous integration and deployment practices that ensure reliability and robustness of the AI model.

Together, these Champions form the backbone of our quality assurance strategy, each upholding their domain's integrity while collaborating closely to achieve a cohesive and high-quality software product.

3.1.4 Communication Roles

Client

The Mentor, also regarded as the Client in this project, is accounted as the single contact point between the team and the client. This permits coordination of all approaching and active correspondence with the client and conveyance to all colleagues.

Supervisor

The Supervisor permits the University and Supervisor to have a single point of contact for the team. In any case, other colleagues might contact the Supervisor for issues themselves. The role is typically filled by the Team Leader.

3.2. Tasks and Responsibilities

3.2.1. General Team Member Responsibilities

- Comply with established coding, documentation, and testing standards as outlined in the SQAP.
- Engage in all team meetings, discussions, and review sessions, providing constructive feedback and insights.
- Stay informed about the latest best practices in AI development and quality assurance to continuously improve personal skills and project outcomes.
- Work closely with other team members, sharing knowledge and assisting with problem-solving.
- Manage personal workload effectively, meet deadlines, and communicate about any challenges or delays.

3.2.2. Champions

Team Leader

- Setting up, running weekly team meetings.
- Booking labs weekly for meeting rooms.
- Contact with the mentor
- In charge of motivating and tracking team progress.
- Monitoring work logs and time spent on projects.
- Responsible for keeping up with managerial documentation.

Client

- This champion is straightforwardly answerable for all correspondences.
- Any incorporated adaptations that should be tried will be sent through the contact.
- Correspondence from colleagues should be handed-off to/from client on time through client contact.
- Any compiled versions that need to be tested will be sent via the liaison.

Documentation

- Keeping up with the document templates
- Maintaining quality and standards of documents.

Code

- In charge of quality control of code artifacts.
- Following code progress.
- Discuss progresses, address any troubles or concerns with the Mentor/ Client.
- Accountable for guaranteeing principles and best practices are met and followed, individually, during the improvement cycle.

Testing

- Appointing running of tests
- Building test archives
- Announcing results of tests to the group, maybe providing advanced requirements

Chapter 4: Documentation

4.1. Software Documents

4.1.1. SQAP

The Software Quality Assurance Plan (SQAP) outlines the processes, standards, and responsibilities required to ensure the quality of the software developed in this project. The goal of this document is to build AI models to identify abnormal behavior in predefined data sets.

4.1.2. SRS

The behavior of the proposed DMS inferred from the customer's needs will be described in the software requirements specification to be created. Although it will be modified to fit our mission, SRS will be based on the IEEE 830 standard.
A general outline of the document is as follows:

1. Introduction

(a) Purpose - Outline of the SRS:

This Software Requirements Specification (SRS) aims to provide guidelines for creating an artificial intelligence (AI) model that can identify anomalous activity in designated datasets. The purpose of this document is to give a thorough explanation of the features, capabilities, and limitations of the suggested system.

(b) Scope - What the DMS is, what it will do and its application:

This project's scope includes creating an AI model that can identify abnormal behavior in a variety of datasets. The system will include preprocessing, testing, model training, and data collection. This system will be mainly used by operations teams, management, and data analysts to properly monitor and respond to anomalous activity.

(c) Definitions, acronyms, and abbreviations

- AI: Artificial Intelligence
- DMS: Data Management System
- ML: Machine Learning

(d) References

…

(e) Overview

This article is organized to give an overview of the project and its background before going into great detail on the general specifications and functionalities of the system. It addresses the functions, user attributes, limitations, presumptions, dependencies, and particular requirements of the product.

2. Overall description

(a) Product perspective - How the product works with other tools, i.e dataflow

The AI model is a stand-alone element that works in tandem with current data analysis software to improve its capacity to identify abnormal behaviors. The system's functions include evaluating incoming data, recognizing odd trends, and offering useful insights. The data flow includes gathering data, preparing it, using an AI model to analyze it, and producing output.

(b) Product functions

- Collect information using surveys to detect abnormal behavior.
- Create and train AI models to recognize abnormal behaviors using ML algorithms.
- Check data to find and report abnormal behavior.
- Provide results to the relevant team (e.g. Team 1-A)

(c) User characteristics - Who will use the product, what training will they get

- Data Analysts: Use the AI model for data analysis. Training required in data interpretation and model usage
- Operations Team: Monitor outputs to respond to detected anomalies. Training required in operational protocols for anomaly response

(d) Constraints

- Limited computational resources for model training and data processing
- Compliance with data privacy and security regulations
- Minimal false positives and false negatives to ensure operational reliability

(e) Assumptions and dependencies

- Assumptions:
  - ❖ There is enough data to train the AI model
  - ❖ Users access the necessary hardware and software tools
- Dependence:
  - ❖ Dependence on existing data sources and data collection methods
  - ❖ Integrate with existing data analytics tools and platforms

3. Specific requirements

- Collect and preprocess data from surveys and other sources
- Train the AI model to detect abnormal behaviors using the specified data set
- Analyze incoming data to identify unusual behaviors
- Generate output in a specific format for further analysis by Team 1-A
- Provides a user interface for data entry, model configuration, and output display
- Integrate with existing data analytics tools via specified APIs
- Comply with all relevant data security regulations
- Deploy data encryption when storing and transmitting to ensure data security

4.1.3. Project Plan

The project aims to develop an AI model capable of detecting abnormal behavior in specified datasets. This AI model will leverage advanced machine learning techniques to analyze data and identify unusual patterns, thereby providing valuable insights for various applications.

4.1.4. Module Plan

The module plan will be developed according to the requirements analysis phase of each iteration in the software life cycle, including:

- Design solution:
    - Survey design: Create a comprehensive survey to collect data on unusual behaviors.
    - Data collection: Implement survey distribution methods and collect responses.
    - Data cleaning: Develop algorithms to clean collected data, removing any inconsistencies or errors.
    - Data preprocessing: Apply preprocessing techniques to prepare data for model training, including normalization, feature extraction, and transformation.
- Work division:
    - Survey design: Tran Quoc Dung + Duong Quoc Trung
    - Data collection: Pham Hoang Duong + Nguyen Thai Son
    - Data cleaning: Duong Quoc Trung + Do Tuan Dat
    - Data preprocessing: Tran Quoc Dung + Do Tuan Dat
- Timeline of artifacts and testing:
    - Survey design: June 16, 2024 - June 30, 2024
    - Data collection: June 30, 2024 - July 15, 2024
    - Data cleaning: July 15, 2024 - July 18, 2024
    - Data preprocessing: July 18, 2024 - July 25, 2024
    - Check and confirm: July 25, 2024 - July 30, 2024

4.1.5. Self-assessment reports

A self-assessment report is to be completed by each team member each as per the unit outline that will provide evidence of work completed and self-reflection. It will document knowledge and experience that has been gained during the process.

This document should contain the following sections:

1. **Summary**

Provide an overview of the work completed during the reporting period. Detail the tasks and activities completed.

2. **Mistakes Made**

Identify any mistakes or challenges encountered and describe how they were addressed or resolved.

3. **Knowledge Gained**

Describe the knowledge and experience gained throughout the process.

4. **Evidence**

Provide evidence of the work completed, such as code snippets, test results, documentation, or screenshots.

4.1.6. Audit Report

Whenever an audit is conducted, a document must be produced to show the results. The report should identify any deviations from the procedures outlined in the SQAP, confirm whether the procedures were followed, and describe any corrective actions taken. Audits can be performed both internally and externally.

1. **Audit result**

Summary of audit results, including key findings and observations.

2. **Follow the procedures**

Details whether the procedures outlined in the SQAP were followed. Highlight any deviations and their reasons.

3. **Corrective action**

List any corrective actions taken to address non-compliance or deviations in the process. Include steps to prevent future incidents and ensure compliance.

These documents will help ensure the project stays on track, maintains high quality standards, and provides transparency and accountability throughout the development process.

4.2. Management Documents

4.2.1. Meeting Agendas

- Introduction and review of previous minutes (5 minutes):
    - Team leader
- Action and next steps for the project (10 minutes):
    - Review action items from previous meeting
- Project review (15 minutes):
    - Team member 1: Status
    - Team member 2: Progress
- Discussion topics (20 minutes):
    - Team member 3 + 4: Issues, updates of the project
- Q&A and open discussion (20 minutes):
    - Open floor for any additional questions or topics

4.2.2. Meeting Minutes

- Date and time the meeting happened
- Names of attendees, as well as absent participants
- Acceptance of, or amendments made to, the previous meeting's minutes
- Decisions made regarding each item on the agenda, such as:
    - Activities undertaken or agreed upon
    - Next steps
    - Outcomes of elections
    - Motions accepted or rejected
    - New business
    - Date and time of the next meeting

Chapter 5: Standards, practices, conventions and metrics

5.1. Purpose

This section serves as the cornerstone for ensuring that all aspects of the AI model development are performed to the highest quality standards. This section outlines the specific standards that will be adhered to during the project lifecycle, the best practices that will guide our work, the conventions that will ensure consistency and clarity across the codebase and documentation, and the metrics that will be used to measure and evaluate the quality of our outputs. By defining these elements clearly, we aim to create a shared understanding among team members and stakeholders of what quality means for our project and how it will be achieved.

5.2. Standards

The following standards will be used as the basis for quality control in this project. Standards will be investigated to guarantee that they are being met.

- **Coding Standard**

- Follow a consistent code formatting style to enhance readability and maintainability.
- Use clear and descriptive names for variables, functions, classes, and other identifiers.
- Write meaningful comments and maintain up-to-date documentation within the code to explain complex logic or decisions.
- Utilize version control systems like Git for tracking changes, collaborating on code, and managing releases.
- Implement robust error handling and logging practices to ensure the system's stability and ease of debugging.
- Write unit tests for all new code and maintain a high level of test coverage across the codebase.
- Adhere to security best practices to protect data and prevent vulnerabilities.
- Optimize code for performance, considering factors like algorithmic complexity and resource usage

- **Documentation Formatting Standard**

- Consistent Structure: All documents should follow a consistent structure with clear headings, subheadings, and numbered sections for easy navigation.
- Adhere to a chosen style guide, such as the Microsoft Writing Style Guide or APA, for consistency in language, grammar, and punctuation.
- Use standardized templates for common document types to ensure uniformity across project documentation.
- Clearly mark the version number and date on all documents and maintain a version history for significant changes.
- Ensure that documentation is accessible, with alternative text for images and proper contrast ratios for text and backgrounds.
- Use clear, concise language that can be easily understood by all stakeholders, avoiding jargon where possible.
- Implement a review process for all documentation to catch errors and ensure quality before distribution.

- **Filename/Location standards**

File Naming Conventions:

- Keeping file names short but concise, quite comprehensive
- Leverage context to avoid redundant, long names

Folder Structure:

- Project Root: The main folder containing the entire machine learning project
- Data: Storing relevant data files, along with the datasets
- Code: Keep your model training and evaluation code here
- Models: Storing trained model files
- Docs: Documentation, including model cards and data documentation
- Results: Storing evaluation results, logs

- **Document Releases**

When the document is submitted to the university, or to the clients, it should be converted into pdf format, which is a static format. Moreover, the document version should be updated, along with the document itself.

5.3. Practices

The accompanying practices will be utilized as the reason for quality control in this project. Practices will be evaluated throughout the project life cycles, to guarantee that they are being followed suitably.

Communication Practices

- **Client**

- Beginning by defining a clear vision and strategy for adopting AI, to understand their goals, expectations and desired outcomes
- Keep the client informed about the project progress, milestones, and any adjustments
- Be transparent about limitations, risks and uncertainties.
- Encourage the client to provide feedback throughout the project

- **Team**

- Learn the AI-specific vocabulary to communicate effectively with data scientists, engineers and other team members
- Define the roles clearly
- Organize regular meetings, brainstorming sessions
- Document decisions, assumptions and progress.
- Utilize Waterfall methodology for iterative development.

- **Supervisor**

- Keep your supervisor informed about the project status, challenges and achievements.
- Discuss any risks or issues that can affect the project
- Ask to know about resource requirements
- Ensuring alignment between the supervisor's expectations and project outcomes

Meetings

- Team meetings will be held twice a week, on Tuesday and Friday, for about 30-60 minutes based on the amount of work.
- All team members are required to be present, in capability.
- A meeting outside of the weekly team meeting does not need all members present; notes are required.
- Notes are compulsory in meetings.

Worklogs

- Team Leader monitors Worklogs.
- Team Leader monitors and maintains the project hours summary sheets.

Coding practices

General guidelines

- Select coding standards relevant to the domain (AI)
- Write easy-to-understand code
- Consider whether the code will be open-source or proprietary
- Including headers or docstrings at the beginning of each module or file
- Use descriptive variable names to avoid ambiguity
- Document the code with comments explaining complex logics
- Once a week meeting (15 minutes) to report progress/difficulties in development, could be conducted after weekly team meeting. The aim is to ensure problems are known early and progress are understood by the whole team.

Guidelines on Component Design

- Developing AI Models:
- Gather relevant data and ensure annotations
- Clean, transform and prepare data for model training
- Train the model using labeled data

- AI-Driven Design Workflows:
- Use reusable AI models to extract relevant features from data
- Select the most informative features for the product
- Train the AI model using the selected features

Chapter 6: Reviews and Audits

6.1. Purpose

This part of the SQAP is to review and validate the final deliverables and verify the team process in regards of the project requirements and standards

Validation helps to see if the requirements and standards match the needs and wants of the clients. This is done with internal and external review

Verification instead helps to see if the product being built matches the requirements specified successfully above. This is done with internal and external review

The standards, procedures and practices can be found in chapter 5, Standards and Practices.

6.2. Review/Audit list

6.2.1. Reviews

Reviews are held during all phases of the project's life cycles:

- **Review Processes Code**

Code quality in regular review is ensured in below tasks. In instances where there's errors or low-quality codes, they will be collected and assigned as a new task for the original coder.

1. Peer review: Everyone in the team will check whether the code is ready for merging into main branch by below standards:
    - Coding Standard
    - Whether results are satisfied
    - Verified against requirement specifications
2. Client review: Twice a week, all work done to the code in the process will be sent to the client for testing based on metrics below:
    - Time finished
    - Verified against requirement specifications
    - Whether the tasks is done correctly

- **Meeting**

Meeting quality will primarily be maintained through audits of the correct process, but all meeting related documents will also be reviewed for quality.

A meeting presentation reporting of what has been done will be briefly discussed and made available before the meeting to show to the clients

Any feedback collected from the meeting will be noted and use as part of the changes made

- **Audits**

Audits are a critical component of our quality assurance process, providing an independent examination of project activities and deliverables to ensure compliance with established standards and procedures. In this project, audits should be held regularly during all phases of the project's life cycle to ensure processes put in place are being adhered to.

- Code Audits: Regularly scheduled code reviews will be conducted by an independent auditor to assess adherence to coding standards, security practices, and overall code quality.

- Documentation Audits: An external reviewer will periodically examine project documentation for accuracy, completeness, and compliance with documentation standards.

- Testing Audits: Testing procedures and results will be audited to verify that all tests have been executed according to the test plan and that they provide sufficient coverage.

- Process Audits: Our development processes, including requirements gathering, design, implementation, and deployment, will be audited to ensure they are being followed correctly and efficiently.

Chapter 7: Testing

The group is to proceed as many test cases as possible on the deliveries prior to providing the product to the client. Unit tests can be used yet won't make up a huge piece of the testing. They can be utilized for affirming mathematical information created by the program has been controlled accurately.

The prototype will also undergo some usability and function testing by the team. Most often, this will entail attempting to corrupt or crash the software. The tester is required to record all mistakes, alerts, and general remarks regarding the GUI's usability as part of this testing. A list of performance metrics that need to be checked and tracked during the testing procedure should also be recorded by the tester.

Once the team has made their best effort to break the program, handle all exceptions, provide usable error messages and are satisfied with the release, it is to be given to the client for feedback. The release should be accompanied with feedback sheets for the client to complete and return to the team so that issues can be repeated, and ultimately, resolved. The client will also be expected to provide an acceptable performance level for the metrics used.

7.1 Requirement

- Develop a comprehensive test plan that outlines the testing strategy, scope, resources and deliverables.
- Create test cases based on the requirements that cover all functionalities, including positive, negative, and edge cases.
- Implement automated testing where feasible to increase efficiency and repeatability of tests.
- Conduct performance testing to ensure the AI model operates within acceptable parameters under various conditions.
- Perform security testing to identify and mitigate potential vulnerabilities within the AI model.
- Regression Testing: Regularly execute regression tests to ensure new changes do not adversely affect existing functionalities.

7.2 Use case generation

Use cases will be implemented by the testing champion, with the observation of the clients. Test cases will permit an essential comprehension for the group, in any case the client will oversee conveying the requirements, purposes of the product to guarantee the group can develop the product appropriately.

7.3 Installation and User Documentation Generation

- Provide step-by-step documents, include system requirements, setup instructions, and troubleshooting tips.
- Create a user manual that describes how to use the AI model, including an overview of features, user interface navigation, and common use cases.
- Compile a list of frequently asked questions to assist users in resolving common issues quickly (FAQs)
- Offer a troubleshooting guide that addresses potential problems users may encounter and provides solutions.
- Establish a feedback mechanism for users to report issues or suggest improvements, which will be reviewed as part of ongoing quality assurance.

Chapter 8: Problem Reporting and Corrective Action

8.1 Personnel

All problems related to personnel within the AI model project team are to be directly reported to the Team Leader, Tran Quoc Dung. This includes issues such as project timeline, task creation, and any other personnel-related difficulties that may arise during the project.

8.2 Work

8.2.1 Project major timeline

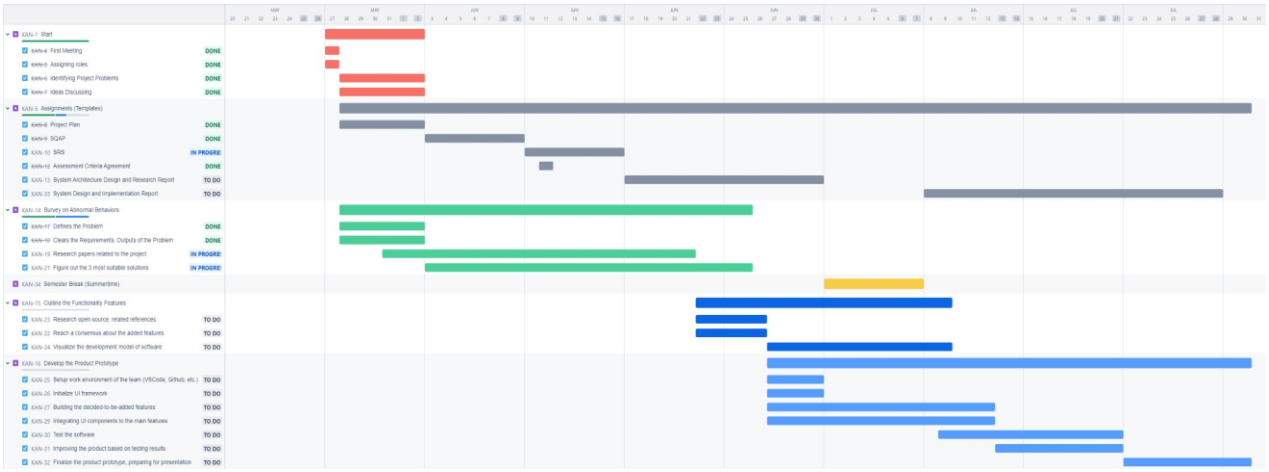

Figure 8.2.1: Project A Timeline (Group 1-C)

8.2.2 Task creation

Team Leader will assign tasks based on the current progression of the project (which stage is current on) and based on the feedback responses from the clients after every meeting. Team members in the project can assign tasks themselves if needed for the current situation of the project (incoming bugs, issues or suggestion for improvements).

8.2.3 Task assignment

Task that needs changes are assumed to be addressed by the original assignee, unless anything is changed by the Team Leader who will be the one notifying said changes

8.2.4 Task life

Task once assigned will have its due dates based on the main Project Plan but it's always up to changes after feedback with the clients. During the review process, if there's more feedback appropriate for the task by the clients, the task will not be marked as complete until no more are provided.

8.2.5 Issue Categories

All categories can be updated to suit the clients feedback and the project needs, these are the most up to date:

- Audit - Internal: Internal audits to ensure compliance with standards and practices.
- Audit - External: External audits by stakeholders or third parties.
- Coding - Prototype: Tasks related to developing and refining the prototype.
- Documentation - PP: Documentation related to the Project Plan.
- Documentation - SQAP: Documentation related to the Software Quality Assurance Plan.
- Documentation - SRS: Documentation related to the Software Requirements Specification.
- Lecture: Tasks related to attending or preparing lectures.
- Meeting - Client: Client meetings for feedback and updates.
- Meeting - Weekly: Weekly team meetings to discuss progress and issues.
- Presentation Preparation: Preparing presentations for stakeholders or internal reviews.
- Research - Coding: Research tasks related to coding and development.
- Research - Documentation: Research tasks related to documentation and best practices.
- Review - Internal: Internal reviews to assess progress and quality.
- Review - External: Reviews by external parties to gather feedback and ensure compliance.

Chapter 9: Tools and methodologies

9.1 Tools

- Microsoft Teams: Used for online meetings with the Mentor/ Client
- Zalo: Used for interacting with the Supervisor, Mentor & Team
- Google Drive: Containing documents, references
- Version Control: Git - for source code management and collaboration.
- Integrated Development Environment (IDE): Visual Studio Code - for writing and debugging code.
- Testing Frameworks: PyTest (for Python) - for writing and executing tests.
- Performance Monitoring: New Relic - for monitoring application performance in real-time.

9.2 Design Methodology

- Iterative Waterfall Life Cycles
- Lifecycles provide many iterations when suitable which works for the client's meetup for feedback and progress
- Each stage has a main due date within the team in the Project Plan to have time verifying with clients for updates
- The iterative model focuses heavily on feedback-based iteration design which will involve the clients in most aspects of the project, providing transparency and correctness to the product
- The model has multiple stages that are also common in other SDLC but coupled with frequents changes due to feedback:
    - Planning
    - Requirement specification
    - Design
    - Implementation and testing

This modified version of the model is seen as below:

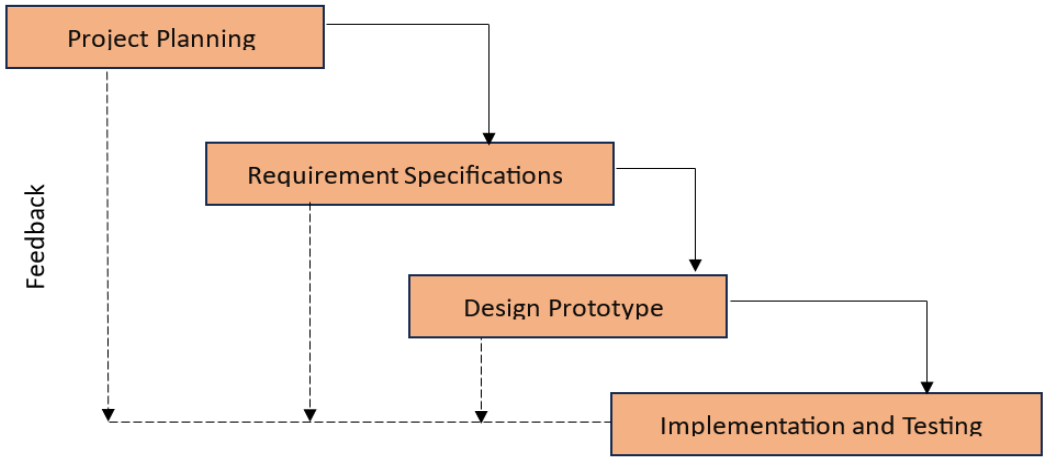

Figure 9.2.1: Iterative Waterfall Life Cycles

Chapter 10: Risk Management

10.1 Purpose

Risk management helps to ensure the quality of the deliverables to align with the client's expectation and requirements as much as possible under severe circumstances

10.2 Categorization

For this project three major categories of risks have been identified:

1. Risks with respect to the work done.
2. Risks with respect to the management.
3. Risks with respect to the clients.

Each risk will be addressed with an associated chance of happening, its impact on the project from low, moderate to high and the most practical action will be suggested to be taken before or after the incident happened to minimize the severity of it and potentially some way to prevent it from happening

10.3 Risks with respect to the work done

- Design doesn't match requirements
  - Probability: Moderate
  - Impact: High, the final product isn't suitable to offer to the clients as it doesn't fit their proposition
  - Preventative Action: Requirements need to be verified during the project often to correctly identify the needs of the clients
- Software can't be run on client system
  - Probability: Moderate.
  - Impact: High, as the product can't be shown to the clients
  - Preventative Measures: Occasional testing with the client to verify deploying ability of product

10.4 Risks with respect to the management

- Sudden absence of team leader
  - Probability: Moderate
  - Impact: Moderate, depending on what point in time the project currently is at
  - Actions taken: If important, organize an immediate meeting to talk about what needs to be done next with everybody else
  - Preventative Measures: Everybody should be made clear about the project since the start and the core work to be done

- Team members don't cooperate or communicate
  - Probability: Moderate
  - Impact: High, the team lacks human resources to continue the project normally, leading to others having to take on more responsibilities
  - Actions taken: Commute with the client to add extension to the project or find a replacement for that team member

- Team members can't finish work due to personal reasons
  - Probability: Moderate
  - Impact: High, as they are responsible for a part of the project
  - Actions taken: Redistribute and reassign the work to other members to complete it before deadlines
  - Preventative Measures: Start the work as early as possible to prevent issues happening by the deadline

10.5 Risks with respect to the clients

- Clients are busy and can't join meetings

- Probability: Moderate
- Impact: Moderate, as early on detection of errors for design and requirements might be crucial
- Actions Taken: Ask for a follow-up meeting if possible
- Preventative Measures: Clear plan discussed with the clients early on into the project

- Project information is not kept a secret within the team

- Moderate
- Impact: High, as this is a breach of confidentiality and losing trust with the clients
- Preventative Measures: NDA signature of all team members early into the project