Assignment2

Part 1

PLS(Project Life Cycle)

1. Preparation.

- Clarifying the overall purpose of the project, the requirements and scope.

2. Planning

- Creating the project roadmap, setting timelines, resource allocation, and estimating costs.

3. Execution

- The actual development work based on SDLC is performed, including designing, coding, and integrating.

4. Monitoring and Control

- Continuous assessment of progress against the plan. Tracking the project's timeline, identify variances, and adjust if necessary.

5. Closure

- After the product is delivered, we will close the project by ensuring the project meets the quality standards and requirements.

SDLC(System Development Life Cycle)

1. Information Gathering

- Collecting user and system requirements to define the functionalities of the eye-track app.

2. Designing

- Translating requirements into technical specifications, designing the user interface and determining on the app architecture.

3. Development

- Developing the eye-tracking functionality, involving multi-threading operations, and data operation.

4. Testing

- Testing the app across different devices to ensure proper functionality.

5. Deployment

- Deploying the app on both Android and iOS platforms.

6. Maintenance

- After deployment, maintaining the app by providing updates and fixing issues.

Relationship Between PLC and SDLC:

In the Execution phase of the PLC, the SDLC is actively carried out. The SDLC fits into the PLC as the "how" of developing the mobile app, while the PLC defines the broader project management framework.

Part2

1. Review the 6 steps to developing an MOV on pp. 62-66. For each step, list:

(1) Identify the Desired Area of Impact

- What I already know: the potential desired area of impact could be user experience and technological innovation, specifically enhancing user interaction on mobile devices through eye-tracking technology.
- Questions to ask stakeholders:

A. What specific impact should the eye-tracking feature have on user engagement?

B. Should we consider increasing app accessibility?

(2) Identify the Desired Value of the Project

- What I already know: The desired area of impact for the eye-track mobile app development project is improving the accuracy of the machine learning model, thereby enhancing the user experience.
- Questions to ask stakeholders:

How should we measure the value of the eye-track app in terms of financial or user experience improvements?

(3) Develop an Appropriate Metric

- What I already know: Metrics could include accuracy of eye-tracking machine learning model, response time.
- Questions to ask stakeholders:

What specific metrics will best quantify the accuracy of machine learning model?

(4) Set a Time Frame for Achieving the MOV

- What I already know: The development cycle is likely spread over several months, and initial outcomes should be measurable by the quarter after the release.
- Questions to ask stakeholders:

What server do we need to deploy the application on, and what requirements does it need to meet?

(5) Verify and Get Agreement from Stakeholders

What I already know: Stakeholders need to agree on data transmission efficiency between the front-end and back-end, and the accuracy range that the machine learning model can achieve.

Questions to ask stakeholders:

Have we accurately captured the expected outcomes for the app?

(6) Summarize the MOV in a Clear, Concise Statement

- What I already know: The MOV will summarize how the app will To improve the efficiency of data transmission between the frontend and backend, and to enhance the accuracy of the machine learning model
- Questions to ask stakeholders:

Does this MOV statement align with stakeholder expectations, and does it clearly outline the project's goals?

2. Review steps 3-7 to developing a business case on pp. 68-74. Complete the following:

(1) Step 3: List of Alternative Solutions

Solution 1: Mobile Eye-Tracking Application (Capturing images)

Develop the proposed eye-tracking app that improves real-time tracking and mobile user interaction.

Solution 2: Use Existing Eye-Tracking Software

Integrate a pre-built eye-tracking SDK (Software Development Kit) into the mobile app instead of building the feature from scratch.

Solution 3: Mobile Eye-Tracking Application (Recording videos)

Develop the eye-tracking mobile that collecting data via recording videos.

(2) For step 5: List as many costs as you can think of involved in the Total cost of ownership (TCO) for your project

- Risk: Privacy Concerns with User Data Collection

  Identification: Collecting eye-tracking data and other personal information may raise privacy issues, potentially violating user privacy laws and regulations.

Assessment: If not handled properly, this could lead to legal consequences, user mistrust, and reduced adoption of the app.

Response: Ensure compliance with privacy regulations such as GDPR or CCPA. Provide transparent user consent options, and regularly audit data collection practices to protect user privacy.

- Risk: Insufficient User Base for Machine Learning Model Accuracy

Identification: A small number of users might not generate enough data for the machine learning model to be accurate and effective.

Assessment: This could result in poor eye-tracking performance, affecting the overall user experience and the app's success.

Response: Mitigate this risk by using data augmentation techniques to artificially expand the dataset. Additionally, collaborate with research institutions or other sources to acquire more diverse datasets to improve model accuracy.

- Risk: Potential License Fees for Cloud Deployment

Identification: Deploying the eye-tracking app on cloud platforms may involve unforeseen license fees for scaling, data storage, and computing resources.

Assessment: This could lead to budget overruns, increasing the overall cost of the project and reducing its profitability.

Response: To minimize this risk, conduct a thorough cost analysis of cloud services before deployment. Opt for flexible cloud providers that offer pay-as-you-go plans, and explore open-source or lower-cost alternatives to keep expenses within budget.

(3) Total Cost of Ownership (TCO)

- Development Costs:

Time spent on coding, testing, debugging, and optimizing the app.

- Cloud Hosting Costs:

Cloud service fees (e.g., AWS, Google Cloud, or Azure) for deploying and running the backend.

Data storage fees for storing user data, images, and model outputs.

Costs for scaling the application as the user base grows.

- Licensing Fees:

Potential costs for machine learning model tools or frameworks.

(4) Total Benefits of Ownership (TBO)

- Learning and Growth:

The team can gain valuable experience in machine learning, mobile app development, and cloud infrastructure, increasing their skills and capabilities for future projects.

- Potential Future Applications

The eye-tracking technology and machine learning models developed could be repurposed or extended for use in other applications or products, providing long-term value.

(5) For step 7: What do you think is the best way to compare and analyze the alternatives listed in step 3 and why? For example, you may choose to use something like table 3.3 on p. 74, the costs/benefits you brainstormed, the risks you listed, or some other criterion. Explain which one you would choose and why.

In this case, I would choose Solution 1: Mobile Eye-Tracking Application (Capturing Images) as the best option. The reasons:

- Customization for Machine Learning Model:

By developing the app in-house, we can control every aspect of the data collection process (such as resolution, frequency, and gaze coordinates), ensuring the data fits the exact requirements for the machine learning model.

- Familiarity with Codebase:

Building the eye-tracking feature from scratch allows the team to be more familiar with the underlying code. This makes it easier to maintain and optimize the system in the future, giving the team the flexibility to adapt the solution as needed.

- Data Security and Privacy:

Having full control over data collection also ensures compliance with privacy regulations, as we can enforce strict measures for handling user data.

- Reduced Data Transmission and Backend Workload:

Sending only photo data to the backend, instead of full video recordings, reduces the size and complexity of the data being transmitted. This minimizes bandwidth usage and ensures faster, more efficient data transfer between the frontend and backend.

The backend will also have a reduced workload because processing individual images is less computationally intensive than processing video streams.

- Comparison with Other Solutions:

Solution 2 (Using Existing SDK): While this might save development time, it lacks the flexibility and control over the data format and accuracy needed for the machine

learning model. Additionally, it introduces dependency on third-party providers, which could lead to higher costs and potential risks in terms of updates and customization.

Solution 3 (Recording Videos): While video data might offer richer information, it would significantly increase the complexity of data handling, bandwidth usage, and backend processing. This could negatively affect system performance and user experience, while also increasing storage and computation costs.