**BAHRIA UNIVERSITY, (Karachi Campus)**



*Department of Software Engineering*

**Assignment 2 - Spring 2024**

# COURSE TITLE: Intr. to Software Engineering COURSE CODE: SEN-210

Class: **BSE-II (A)** Time Allowed:  **2 Week.**

Course Instructor: **ENGR. RAHEMEEN** Max. Marks: **7 marks**

Submission Date: **19-05-2024**

**Note:**

* This assignment should be submitted in to group (only 2 members)
* Strictly follow the given deadline.
* Assignment must be submitted on LMS.
* Must be in PDF file only.

Abdullah (02-131222-099)

Sophisticated flight control system

Project Plan:

1. **Requirements Analysis**

For the requirements analysis phase, we'll conduct a thorough examination of both functional and non-functional aspects of the flight control system. This includes:

* **Functional Requirements:**
* Real-time data processing for sensor inputs (GPS, altitude, temperature, speed etc.).
* Precise navigation algorithms for autonomous flight and mission planning.
* Fault tolerance mechanisms to handle system failures gracefully.
* Integration with sensors (cameras, lidar, etc.) and actuators (motors, servos, etc.).
* Communication protocols for remote control and data transmission.
* **Non-functional Requirements:**
* Performance constraints (response time, accuracy, reliability).
* Safety critical aspects adhering to aviation regulations (FAA, EASA, etc.).
* Scalability for future upgrades and enhancements.
* User-friendly interface for operators and maintenance personnel.

1. **System Architecture Design**

The proposed high-level system architecture will include:

* **Flight Control Unit (FCU):** Central processing unit for data processing, navigation algorithms, and control logic.
* **Communication Module**: Handles communication between ground control station and UAV.
* S**ensor Interface**: Interfaces with onboard sensors for data acquisition.
* **Actuator Interface**: Interfaces with actuators for control commands.
* **Data Storage and Logging**: Stores flight data for analysis and logging.
* **User Interface**: Provides a graphical interface for operators.

Interfaces between these components will use industry-standard protocols like CAN bus for internal communication and Wi-Fi/RF for external communication.

1. **Software Development Methodology**:

We propose an Agile software development methodology for this project due to its iterative nature and adaptability to changing requirements. The development process will include:

* **Sprint Planning:** Define sprint goals and tasks based on prioritized features.
* **Daily Standups**: Short daily meetings to discuss progress and challenges.
* **Iterative Development:** Build, test, and iterate on software components in short cycles.
* **Continuous Integration/Continuous Deployment (CI/CD):** Automate testing and deployment processes for faster feedback loops.
* **Version Control:** Use Git for version control and collaboration among team members.

1. **Risk Management**

Identify potential risks such as:

* Technology dependencies on third-party libraries or hardware.
* Regulatory compliance issues with aviation standards.
* Integration complexities with different subsystems.

Risk mitigation strategies include:

* Conducting thorough feasibility studies and technology assessments.
* Regular communication with regulatory bodies to ensure compliance.
* Prototyping and testing integration early in the development cycle.

1. **Testing and Validation**:

Define a comprehensive testing strategy covering:

* Unit Testing: Test individual software components.
* Integration Testing: Test interactions between software modules.
* System Testing: Test the entire system in simulated and real-world environments.
* Performance Testing: Evaluate response times and system scalability.
* Safety Testing: Verify fault tolerance mechanisms and emergency procedures.

Criteria for acceptance include meeting performance metrics, regulatory compliance, and reliability benchmarks.

1. **Documentation and Maintenance**

Outline a plan for:

* Comprehensive documentation including system architecture, user manuals, and API references.
* Regular maintenance and updates based on user feedback and evolving requirements.
* Training sessions for operators and maintenance personnel.

References:

<https://termaviation.com/what-is-afcs-in-aviation/#google_vignette>

<https://www.aircraftsystemstech.com/p/automatic-flight-control-system-afcs.html>