Test-Planning Document

Purpose:

This document outlines the planned analysis and testing activities for a selected subset of system requirements for the ILP Medical Drone Service. This document aims to construct a coherent, risk-aware test plan that integrates test planning, instrumentation, and scaffolding within a defined lifecycle. This plan focuses on a select few representative requirements rather than exhaustive coverage of the full requirement set. The plan makes use of the principles of validation and verification, early defect detection, scaffolding and instrumentation and lifecycle-aware planning and risk analysis. A Test-Driven Development (TDD) incremental lifecycle is chosen.

Selected Requirements:

Three requirements were selected to ensure coverage across different requirement types, criticality levels, and system abstraction layers:

SR1 (Safety Requirement 1): No-Fly Zone Compliance
Type: Safety
Level: System/Integration
Priority/criticality: High

CR1 (Correctness Requirement 1): Positioning Accuracy
Type: Functional correctness
Level: Unit/Integration
Priority/criticality: High

CR6 – Path Validity (Complete Validation)
Type: Functional correctness (system integrity)
Level: Integration/System
Priority/Criticality: High

PR1 (Performance Requirement 1): Pathfinding Response Time
Type: Performance
Level: System
Criticality: High

These requirements are representative of broader safety, correctness, and performance concerns in the system, defined for deep analysis and testing. Due to limited resources, testing effort is prioritised towards these high-risk requirements rather than trying to complete exhaustive testing for all requirements.

Priority and Pre-requisites

At this stage of the test planning process, it is essential to assess each requirement to determine the appropriate allocation of analysis and testing effort. This assessment considers the criticality, priority, and potential need for scaffolding and instrumentation.

SR1: No-Fly Zone Compliance (Safety-Critical)

- Priority and Criticality: High: As a safety-critical requirement, SR1 has direct regulatory implications. Violating set regulations and no-fly zones could lead to dangerous system behaviour
- Analysis and Testing Considerations:
    - Verification: Static code inspection is required to ensure that the no-fly zone logic is implemented accurately
    - Dynamic Testing: Integration-level tests that focus on boundary cases where flight paths approach or cross no-fly zones. Representative inputs are selected from defined input groups to achieve good coverage without testing every possible combination
    - Early Detection: Early verification is prioritised so that defects are identified quickly and less effort is required in later stages of testing
- Scheduled Tasks:
    - Conduct code inspections of no-fly zone enforcement logic
    - Develop automated input generators for boundary and transition testing
    - Implement dynamic tests to verify compliance under representative scenarios

CR1: Positioning Accuracy (Correctness)

- Priority and Criticality: High: Accurate positioning is critical to safe and reliable operation of the navigation system
- Analysis and Testing Considerations:
    - Unit Verification: Deterministic tests using synthetic coordinates with known expected outputs
    - Integration Verification: Testing interactions between positioning, navigation, and control code
    - Validation: Limited scenario-based validation to confirm that positioning accuracy meets the specified operational expectations
    - Iterative Testing: Performing repeated tests early, following a test-driven development approach, helps quickly identify and locate faults while ensuring the correctness of the system
- Scheduled Tasks:
    - Develop and execute unit tests for positioning computations
    - Perform integration tests for navigation and control interactions
    - Instrument and log positional data for analysis during simulated operational scenarios

CR6 - Path Validity (Complete Validation)

CR6 ensures that all generated paths are globally valid, meaning they:

- Start and end at valid locations
- Do not violate no-fly zones
- Remain within operational boundaries
- Do not exceed maximum move or path length constraints
- Are continuous and reachable

- Priority and Criticality: High. CR6 integrates multiple safety and correctness constraints and ensures that isolated correctness results in valid system-level behaviour.
- Analysis and Testing Considerations:
    - Integration Testing: Validation that pathfinding, safety checks, and configuration limits interact correctly
    - Negative Testing: Generation of invalid paths (e.g. excessive length, unreachable destinations) to ensure rejection
    - System-Level Testing: End-to-end mission scenarios validating full path generation under realistic constraints
- Scheduled Tasks:
    - Implement integration tests combining pathfinding, no-fly zone enforcement, and configuration limits
    - Generate invalid and edge-case paths to verify correct rejection behaviour
    - Validate complete path correctness during full simulated delivery scenarios

PR1: Pathfinding Response Time (Performance)
- Priority and Criticality: High: Efficient pathfinding is essential for responsive navigation and mission execution.
- Analysis and Testing Considerations:
    - System-Level Verification: Performance tests making use of synthetic workloads to measure response times under controlled conditions.
    - Validation: Monitoring and logging response times during realistic simulated missions to ensure operational relevance
    - Preparatory Work: As PR1 can only be evaluated meaningfully at the system level, it is necessary to develop scaffolding and instrumentation in advance to perform assessment of system performance at a later stage
- Scheduled Tasks:
    - Construct a simulation environment to execute performance tests
    - Implement logging and instrumentation to capture response times
    - Generate synthetic workload scenarios to simulate pathfinding under varying conditions

Scaffolding and Instrumentation:
Even though the drone system is currently a software simulation rather than real hardware, the project will still require scaffolding and instrumentation to support effective analysis and testing.

SR1: No-Fly Zone Compliance
- Inspection of the code logic to ensure that the drone never enters no-fly zones does not require scaffolding.
- For exhaustive testing:
    - Scaffolding: Input generators will create synthetic flight paths and obstacle configurations, including edge cases near no-fly zones. The simulator will feed these scenarios to the navigation system

    - Automated tests will verify compliance by checking that all generated paths respect no-fly zones
    - Building this scaffolding is an early task, as it enables repeated safety testing
- Instrumentation: Safety check assertions are added to automatically detect no-fly zone violations, with logging to record any violations and the drone's position relative to restricted areas

CR1: Positioning Accuracy
- Unit/Integration Testing:
    - Scaffolding: The navigation module will be tested with synthetic coordinates to measure positioning accuracy
    - Instrumentation: Logging of predicted vs actual positions and error calculations will allow analysis of accuracy metrics
- Integration Testing: The navigation and pathfinding modules will be combined in the simulator to ensure accuracy is maintained across the system

CR6 – Path Validity
- Scaffolding: End-to-end simulation scenarios generating complete delivery paths.
- Instrumentation: Logging of full path sequences, move counts, boundary checks, and constraint violations.

PR1: Pathfinding Response Time
- System-Level Performance Testing:
    - Scaffolding: The full simulation environment will be required, including dynamic obstacles and mission scenarios to test pathfinding under realistic conditions
    - Instrumentation: Performance counters will measure the time taken by the pathfinding algorithm to respond to dynamic changes
    - Simulated scenarios will be reused to systematically stress-test the system, and repeated runs will verify that response times meet requirements

After identifying scaffolding and instrumentation needs, these tasks can be scheduled within the lifecycle. Early implementation of simulators, input generators, and logging ensures that safety, correctness, and performance requirements can all be tested effectively. The use of instrumentation provides the necessary data to verify compliance, assess accuracy, and evaluate response times.

Process and Risk
The tasks identified must be scheduled within the project lifecycle to ensure systematic, risk-aware testing of the ILP Medical Drone Service simulation. The lifecycle assumes an incremental, Test-Driven Development (TDD)-inspired approach, which allows early verification and iterative validation, while facilitating prompt detection of defects

Process Considerations:
- **Early Scaffolding Setup:** Simulators, input generators, and logging tools for all requirements should be built at the start of the project. This makes it possible to run

repeated and automated tests throughout development and allows safety and correctness checks before the full system is integrated

- **Incremental Testing:** Unit and integration tests for CR1 (Positioning Accuracy) should be run as soon as synthetic coordinates and positioning logic are ready. Testing early and often helps catch errors sooner and makes it easier to locate and fix problems
- **System-Level Testing:** Tests for PR1 (Pathfinding Response Time) and full SR1 (No-Fly Zone Compliance) require the integrated simulation with realistic scenarios. These tests are performed after the individual modules have been verified, ensuring accurate results and preventing lower-level errors from hiding
- **Parallel Tasks:** Some preparation tasks, such as generating synthetic data, setting up simulators, and implementing logging, can be done alongside early module verification. This helps make efficient use of time and resources

Risk Considerations

Several risks are inherent in this test plan and need to be addressed to ensure reliable evaluation of the drone system:

1. **Synthetic/Generated Data Representativeness**
    - **Risk:** The simulated flight paths or mission scenarios might not accurately reflect real-world conditions. This could give a false sense of confidence in the system's behaviour.
    - **Mitigation:** Create a wide variety of scenarios, including edge cases for no-fly zones, obstacles, and high-density pathfinding situations. Regularly refine the scenarios based on system performance during testing
2. **Instrumentation Overhead**
    - **Risk:** Excessive logging or performance monitoring could affect the system's behaviour, particularly when measuring response times for PR1.
    - **Mitigation:** Keep instrumentation lightweight, constantly check its overhead, and use selective logging or sampling during performance-critical tests
3. **Resource Constraints**
    - **Risk:** Limited developer time or computational resources could delay the creation of scaffolding or the execution of tests
    - **Mitigation:** Focus on high-priority requirements (SR1, CR1, PR1) first. Schedule the construction of reusable scaffolding early and maintain a task backlog to allow parallel execution where possible
4. **Defects Propagating**
    - **Risk:** Errors in early-stage modules could carry over into system-level testing, making debugging more difficult.
    - **Mitigation:** Follow a test-driven development approach with frequent unit and integration testing to catch defects as early as possible

By implementing early scaffolding, parallelisable tasks, and carefully designed instrumentation, this plan reduces risks associated with safety, correctness, and performance. Regular review of test results and ongoing refinement of synthetic scenarios ensures that confidence in the drone system's behaviour grows alongside its development.