

# Software Testing 2022

S1833951 li fang

January 26 2022

## 1 Analyze requirements to determine appropriate testing strategies

### 1.1 Range of requirements, functional requirements, measurable quality attributes, qualitative requirements

#### Stakeholders:

- Government for using it measuring data regularly
- Company used for navigation system in their delivery drone
- Academy used for academy research
- Personal usage for entertainment

#### Requirements:

- Predicted route should meet the distance requirement - accuracy  
The route distance should be under constrain, as it should find the minimal distance to complete the delivery.
- Time for finding such route - performance  
The total time for the software to find the predicted route and actual route.
- Always meet the distance requirement - robustness  
The software does always find the route that meet the requirement for different travel plan
- The predicted plan is able to be conducted - accuracy  
As the drone does move in a certain manner, the software should be able to transfer the predicted route to a route that matches the moving pattern of the drone, after the transformation the route should also meet the requirement.
- The route meets the requirement of safety - reliability  
The route should not lead the drone in to any building or let it moving out the restricted area.
- Unauthorized approach should not be able to change the route – security  
Unauthorized approach should not be able to change the determined final route, or changing the algorithm at any point.
- The software should be able to undertake large number of inputs – scalability  
As this allows us to test the performance, also allowing the software to host large number of requests.

### 1.2 Level of requirements, system, integration, unit.

#### System level:

- The software does check if the predicted pass meets the safety requirement and choose to accept and add it to the route or return a false to find another route.
- The software check if the input point list is in correct format, if not return an error message.
- Make sure the amount of input does not exceed the system limit (amount of route need to be computed)

#### **Integration:**

- The software should use the predicted path as guide, to generate a actual path based on the drones limitation and feature, such as the drone does have battery limit, can only move in one direction for a fix distance
- The Drone should properly receive the data required from the webserver and other online resources.

#### **Unit:**

- The drone should have function to find the next nearest point based on the distance calculated.
- The drone should have algorithm predict if the following move will hit the building and choose alternative path

### **1.3 Identifying test approach for chosen attributes.**

- **Accuracy:**  
Unit tests can be used to improve the accuracy of the predicted route, including verifying that the safety requirements for each move are met and that points are not being revisited. This can be accomplished by comparing the calculated route to the building area and restricted areas to ensure that the route does not intersect with them, and checking if a point is already included in the route. Integration tests can be used to verify that the route includes all necessary points and starts and ends at the correct locations. This can be done by comparing the starting and ending points of the route to the expected points, and checking if all delivery addresses are included in the route.
- **Performance:**  
Use statistical testing to check the time it takes to generate a predicted route, and an actual route. Find the average time to generate such route. Use performance testing to see if the actual route is meeting the requirement of the given test list. This would be done by comparing the distance of each output route to the constrains, and measure the percentage of the route meeting the requirement(distance).

### **1.4 Assess the appropriateness of your chosen testing approach.**

- The choice of unit test for improving accuracy is good as the software does gain the route by looking at all the next possible point and building up the route step by step while make sure the next point meet the requirements, and unit test is good at testing small function or variable and friendly to changes, so I can use unit test while I develop the software, also it can test small function separately such as the part to check the

drone not hitting any building or the part checking the drone is not visiting a visited point.

Since there are a lot of constraints in choosing the next point, I can have unit tests for each constraint and assemble the result of the test in the end. This does not include any external or global dependencies, or require checking the overall structure of the software, so it does avoid the problem of the unit test such as limited scope or coverage, and maximize its advantages as a test method.

- The choice of system for accuracy as it can test the integrated result, the route. This could not be covered by the unit test, so we need integration tests to check the output is overall satisfy the requirement, such as safety requirement is met for the overall route or all the points have been delivered. The integration test combined with unit tests we can ensure the overall function could be tested in both micro aspect and overall aspect.
- Performance testing is crucial and should be used as it can see if the software is capable of providing route under all circumstances, and I can improve the software based on the result. Such as the software does not have good performance when the route does contain path from point C to G, I could try to look at the specific case and improve it accordingly. Since the route does contain a lot of points and will be tested for a lot of point combinations to form a route, performance test will be helpful improving the overall quality of the software.