Software Testing 2022 portfolio

S1833951 li fang

January 26 2022

Description of the software

The software aims to give a route that the drone is capable to measure the air quality of a list of points within a maximum total distance. There are constrain as the drone should not hit any building, and should not go outside the area of delivery. My software does achieve this by calculating a predicted test plan first, and secondly transform the predicted plan to actual route for the drone to conduct actual movement. So, the software is roughly in two parts, the part calculating the predicted route and the part control the drone to move according to the predicted route. And there are classes design for taking the reading, carried the move, and read the webserver data online.

1. Analyze requirements to determine appropriate testing strategies

There are a lot of requirements, and there are a lot of constrains when it comes to the drone, such as the battery, path number, angle of travel, not hitting any building etc. Among them the capability of route planning is the most important. So, in the route planning section, it can be divide to two sections, accuracy and performance, accuracy means it make sure the basic function is well carried out and no severe mistake should happened. Performance means it should carry the task with reasonable efficiency, since the software should be able to generate a lot of paths at the same time. So, I have chosen system test and unit test as the main way of testing, since the software does have a lot of small function contained, unit test should be useful to test the individual of them, and have a system test to test the sub-section or overall performance is acceptable.

2. Design and implement comprehensive test plans with instrumented code.

R1: the drone should remain in the restricted area always

- A safety requirement, as the software should make sure drone always stay in a rectangular restricted area, any path crossing the area is consider as serious failure.
- The input will be the start and end point, include the restricted zone's coordinate, and the output should be a single Boolean value shows the path does not exceed the restricted zone.
- The partition and visibility principle have been take in consider here, check supportive document for specific explanation.

Test plan:

- Mainly using generated synthetic data and unit test here. A system test might be needed as there are rectangular area and building, the test might check if requirement is meet for both of the area.
- Weakness is that it may hardly cover all the situation as generated data.

R2: the drone should always find the next nearest point when planning the visit

- The very basic function for any route-finding software, a performance requirement but crucial one.
- The input will be the starting point and the list of pint need to visit, and out put should be the nearest point.
- Principle of partition and visibility is used here.

Test plan:

- Generate a unit test for the distance calculation function along, and a system test for the overall requirement. The synthetic data is used too, to stimulate the real world scenario as much as it could.
- Risk is system test can hardly tell where did the bug came from, and generated data can hardly cover all the real world situation.

No instrumentation is planned here, as the project is not test driven, for more specific reason see the supportive document.

3. Apply a wide variety of testing techniques and compute test coverage and yield according to a variety of criteria.

R1:

- Mutation test, function test and other test technic is used here, ensure the general functionality of the function.
  The technic does have draw back, as it checks the behavior of the function, so can hardly find where did the function go wrong, and generated data can hardly cover all situation. Also, the security is not covered here too.

R2:

- Mutation test, systematic test and other technic are used here.
  Still, generated data can hardly cover all situation, and security is left behind for these test technics.

The overall outcome is good, when test is passed the final outcome is meeting the expectation, and when the test if failed, there is certainly something wrong in the code, this is proved by mutation test. Scaffolding is used and been convenient during the phase of developing. For more detail, please check the supportive document of this section.

4. Evaluate the limitations of a given testing process, using statistical methods where appropriate, and summaries outcomes.

There gap between the expected result and the actual result.

Target:

- Unit test are supposed to reveal the bug in the code, and do not pass whenever the function is not working or part of the variable is mix-up. As its feature, it is not expected to tell anything outside this function.
- System test are expected to fail when any of the function in this sub-branch contains bug or not functioning. It is not expected to report the exact place of failure.

Outcome & evaluation:

- For unit test in R1, it is still hard to locate the problem even it is failed and been designed case by case. The main reason is the function use method that is too compact, it is required to replace this method so test can be design to see the place of bug.
- For R2, the system test does perfectly of what is expected, fail every time when doing mutation test. But the unit test has similar problem, the unit test is design upon behavior check, so can hardly know the exact place of code go wrong. Rewrite the test would be needed

5. Conduct reviews and inspections and design and implement automated testing processes.

Code:

- There is readability problem, maintainability problem and other problems for both the requirement code. For specific check the supportive document

CI pipeline:

- There are many things that should be take in consider when construct a CI pipeline. Include source code management, automated build, code analysis and so on
- The test should be automated are unit test, the one best suit for automation. And integration test, performance test and security test.
- The functionality of CI pipeline should demonstrate by its capability of measuring code coverage, checking history of build, monitoring the states of system and other crucial points.