

#### Gaps and omissions in the testing process:

- A lack of time. Since we want to get the product to production as soon as possible, we had to trade off extensive testing. Time to market is prioritized over the high dependability hence the completeness quality process is often ignored.
- The lack of data that reflects the real world. The synthetic data used for testing was created based on the tester's assumption of how the orders would be like and the how big the demand would be. Since there has not been any previous work done which is similar to this project, we are unable to use any data that reflects the real world to test this project.
- Although the flightpath of the drone was tested and shown to visit the locations it needs to visit and it avoids no fly zones, the programs capability to withstand external forces like strong winds were not accounted for or tested. This is because of the lack of gps sensors on the drone which prevents it from knowing if it has been thrown off course.
- There is no way of checking the condition of the load carried by the drone due to the lack of sensors. We only test the orders to make sure only 4 items are ordered maximum but never do we check if the drone is able to handle the load(different food may have different weight) or the condition of the items (eg if the food was damaged).

#### Targeted coverage/ performance levels:

- For the efficiency of the software, the target is for it to have a runtime of less than 60 seconds which is ample time. However, this is just an initial target. In future releases it would make more sense for it to have a runtime of 30 seconds or less
- We would also expect there to be more test cases to test those lines of code that were not covered by the current tests. This would hopefully in turn give a 100 % line, class and branch coverage.
- The percentage of money gained and percentage of completed orders would also be expected to be at least 90%, which I would consider the minimum percentage to be considered a functional and optimal system in the service industry.
- One of the functional requirements was for the delivery time to be short. This was not explicitly tested in the test plan. Instead the average delivery time was tested and compared between different heuristics. Now that the heuristics are decided upon, we can target for the delivery time to be at a maximum of 10 minutes which is a decent time given the size of the area the drone is confined to.
- A qualitative requirement that we have not set a target level for is that the flight of drone should be safe for the public. Safety is not a measurable aspect but we can ensure this by targeting the average number of crashes by the drone per month to be minimum and set at 5 or lower. This would be the same for the average number of times the drone crosses a no-fly-zone.

#### How do we achieve the target levels:

- As I mentioned above, the main factor was the lack of time due to the prioritization in time to market. With more time, more extensive testing can be done and other adequacy criterias can be considered.
- If we could get better test data or data that is more tailored to real world experiences then the program can conduct more effective testing, hence eventually reaching those target levels when use in the real world.

- The installation of sensors on the drone would also allow us to eventually reach those target results. Some of those targets are impossible to measure now such as the drone crashing or flying into no-fly-zones due to strong winds. Hence with these sensors these targets would eventually be achieved.
- The lack of time prevented me from using external testing frameworks and tool such as openClover and jacoco. With more time and by using these tools and frameworks, more defects in the code will be detected and rectified which would eventually lead to the program being better optimized, running better and yielding better results hence eventually meeting those targets.