Software Testing 2022/3 Portfolio

Weiliang Zhao

https://github.com/Zhaoweiliang/Software-Testing-Code-and-file.git

January 28, 2023

1 Outline of the Software Being Tested

Understanding the spread of fire lies at the heart of fire preventing measures. I developed a model which can simulate the spread of fire and incorporates the effect of firefighters, firebreaks and different materials flammability. Our model considered a two-dimensional domain divided into a grid of square cells. These cells during the fire can be either flammable, burnt or burning with the probability of ignition of flammable states different for different materials. The state of any cell depends on the state configuration of the neighbouring cells. I used a mathematical model to examine this configuration state along with the wind effect, fire fighters' effect and the burning state's duration. To validate the model, I used it to simulate the great fire of London in 1666 and the model can reasonably accurately simulate fires in urban areas.

2 Learning Outcomes

1. **Analyze requirements to determine appropriate testing strategies** [25%]

   (a) **Range of requirements, functional requirements, measurable quality attributes, qualitative requirements**
   The software should have accurate fire behaviour prediction, integration capability, user-friendly interface, scalability, reliability, and robust security. Please refer to **Requirement.pdf** for specific details

   (b) **Level of requirements, system, integration, unit**
   For the system level requirements, my software needs to be 80% accurate when simulating the fire within 12 hours and 60% of accuracy within in 48 hours. For the unit level of requirements, The software will check the format of the input data and provide a notification of an error if the format is not correct.The software will restrict the quantity of input data to ensure that the data falls within the acceptable range for the program to function correctly. For integration requirements, the software can try to obtain fire and weather conditions through local weather centres.

   (c) **Identifying test approach for chosen attributes**
   The test approach I have chose for each attributes is written in the file **Test Approach.pdf**.

   (d) **Assess the appropriateness of your chosen testing approach** In this section, I analyzed several tests mentioned earlier, in the document **Appropriateness of Test Approach.pdf**, I analyzed whether these tests are appropriate tests to check the software's requirements. At the same time, I also discuss the weaknesses of each test approach.

2. **Design and implement comprehensive test plans with instrumented code** [default 15%]

   (a) **Construction of the test plan**
   In the test plan I, have discussed the two major overall requirements of the software Safety and Accuracy. Then explained the Priority, Prerequisites, process and risk based on the requirement. Furthermore, introduced several tasks to make sure the requirements are maintained. Please refer to **Test Plan.docx** for specific details.

(b) **Evaluation of the quality of the test plan** The detail for the strength and weaknesses of the chosen tests are discussed in the **Evaluation of the test plan.pdf**

(c) **Instrumentation of the code & Evaluation**

    i. Simulation time:
Adding code to measure the time it takes to run the simulation, which can be used to identify and optimize performance bottlenecks.

    ii. Area coverage:
Adding code to measure the percentage of the simulated area that is affected by the fire, which can be used to compare the simulation results with real-world data.

    iii. Agent behavior:
Adding code to track the behavior of agents (e.g. firefighters) during the simulation, which can be used to evaluate their effectiveness and optimize their behavior.

    iv. Fire spread rate:
Adding code to measure the rate at which the fire is spreading, which can be used to evaluate the effectiveness of the fire spread algorithm.

The above-mentioned Instrumentation greatly improves the efficiency of my unit testing of various modules in the software, such as the fire coverage parameter, which allows me to more efficiently compare the output results of the software with real data to determine whether the test has passed.

3. **Apply a wide variety of testing techniques and compute test coverage and yield according to a variety of criteria** [default 15%]

(a) **Range of techniques**
In the unit test section I use functional testing(Systematic Testing), Input/Output testing, Negative testing, User acceptance testing. Then in the validation testing I mainly use the comparison testing to test the accuracy. Furthermore, I also consider the compliance testing but due to the amount of effort that required to satisfied National Fire Protection Association (NFPA) standards is Incalculable I have completed it.

(b) **Evaluation criteria for the adequacy of the testing**

    i. Requirement 1 'Firefighter deployed':
The chosen testing methods should effectively identify any issues related to fire detection at the locations where firefighters are deployed. Utilizing unit test can effectively identify these basic and clear-cut problems. Additionally, conducting sensitivity tests will guarantee the safety of firefighters when deployed in any area of the diagram.

    ii. Requirement 2 & 3:
The tests employed are effective in identifying the fire state and burning status of each cell. Utilizing unittest allows for the detection of these relatively basic and straightforward issues efficiently. However, for requirement R2, testing at each iteration when the iteration count is high may negatively impact the performance of the software simulation. To maintain efficiency, it is recommended to select a limited number of tests for each simulation.

    iii. Accuracy:
To ensure the accuracy of the simulation, validation tests are used to verify that the software's results align with real-world outcomes. However, due to the diversity of data types and the limited number of test cases, additional test sets are necessary to fully guarantee the simulation's accuracy.

(c) **Results of testing & Evaluation of the results**
The Results and the evaluation of testing is discrible at **Test Result.pdf**

4. **Evaluate the limitations of a given testing process, using statistical methods where appropriate, and summarise outcomes** [default 25%]

(a) **Identifying gaps and omissions in the testing process** In the **test gap.pdf** I have identified several the gaps for each tests I used in LO3.

(b) **Identifying target coverage/performance levels for the different testing procedures**
The target coverage/performance levels for each test is discussed in this file **Test Coverage.pdf**

(c) **Discussing how the testing carried out compares with the target levels**
The target coverage/performance levels for each test is discussed in this file **Test Coverage.pdf** as well

(d) **Discussion of what would be necessary to achieve the target level**
In order to improve the unit tests, the testing code should be optimized to ensure that it outputs more detailed and informative results. For example, incorporating a mechanism to record the specific input values and iteration number at which a failure occurred, in addition to the current state of the simulation. For validation tests, more test data should be used and a more comprehensive approach should be taken towards considering edge cases and boundary conditions, in order to increase the robustness and reliability of the tests. For sensitivity tests, the method of data selection should be refined, using more representative and statistically significant data sets, and sufficient time should be allocated to ensure that the data used in the tests is comprehensive enough to cover a wide range of possible scenarios. Additionally, a more systematic approach should be taken towards testing variations in input parameters, such as varying the values within a certain range or testing specific combinations of input values.

5. **Conduct reviews, inspections, and design and implement automated testing processes** [default 20%]

(a) **Identify and apply review criteria to selected parts of the code and identify issues in the code**
In this report, I have evaluated the necessary considerations for implementing R1-R3 with regard to the reliability, usability, scalability, and readability of the software code. Please refer to the document "**Code Review.pdf**" for further details

(b) **Construct an appropriate CI pipeline for the software**
All the details about the CI pipeline is in the file **CI pipiline.pdf**

(c) **Automate some aspects of the testing**
All the details about the CI pipeline is in the file **CI pipiline.pdf**

(d) **Demonstrate the CI pipeline functions as expected**
All the details about the CI pipeline is in the file **CI pipiline.pdf**