

Range of techniques

Weiliang Zhao

January 28, 2023

1 Accuracy

1.1 Unit Test

1. Functional testing:

This involves testing the software's functional requirements and checking if they meet the desired specifications. For example, if the simulation software is supposed to simulate fire spread in a certain area, a functional test would involve inputting the area's parameters and checking if the simulation returns the expected results.

2. Boundary testing:

This involves testing the software's input values at the minimum and maximum limits to check if the software can handle extreme cases. For example, if the software can handle the minimum and maximum limit of area, fuel load, and wind speed.

3. Negative testing:

This involves testing the software's behavior when it receives invalid or unexpected input values to check if it can handle errors and unexpected situations. For example, giving the software an area with negative values or a wind direction that does not exist in the real world, and checking if the software can handle it without crashing or giving unexpected results.

1.2 Validation Test

1. Comparative testing:

This involves comparing the simulation results with real-world data to check for consistency and accuracy. For example, comparing the simulation results of a particular scenario with data collected from a real-world fire in that scenario to check for consistency and accuracy.

2. Functional testing:

This involves testing the software's functional requirements and checking if they meet the desired specifications. For example, if the simulation software is supposed to simulate fire spread in a certain area, a functional test would involve inputting the area's parameters and checking if the simulation returns the expected results.

3. Compliance testing:

This involves testing the software to ensure that it complies with industry standards and regulations. For example, checking if the simulation software complies with the National Fire Protection Association's (NFPA) standards for fire spread simulations. However, due to the amount of effort that required to satisfied National Fire Protection Association (NFPA) standards is impossible to completed it with in 50h.

2 Performance Test

1. Volume testing:

This involves testing the software's performance under different data volumes. For example, running a simulation with different areas and fuel loads and checking if the software can handle the different data volumes without slowing down or crashing.

2. Endurance testing:

This involves testing the software's performance over a long period of time to check if it can handle sustained loads. For example, running a simulation for several hours and checking if the software's performance remains consistent over time.

3. Spike testing:

This involves testing the software's performance under sudden and drastic changes in load. For example, running a simulation with a high number of fires for a period of time, then abruptly increasing the number of fires and checking if the software can handle the spike in load without slowing down or crashing. However, this test is high cost of money, resources and time which means is not suitable to be made in this project.

3 Sensitivity Test

1. A/B testing:

This involves testing different versions of the software with different groups of users and comparing the results to see which version is more effective. For example, giving version A of the software to a group of users and version B to another group of users, and comparing the results to see which version is more user-friendly.

2. Beta testing:

This involves testing the software with a group of real users in a real-world environment to check for any bugs or usability issues. For example, giving the software to a group of fire department officials, and asking them to use it in real-world scenarios and reporting any bugs or usability issues they encounter.

3. User acceptance testing (UAT):

This involves testing the software with end-users to ensure that it meets their requirements and is fit for purpose. For example, giving the software to a group of fire department officials, and asking them to perform specific tasks and gather feedback on whether the software meets their needs and if they would use it in their work.