Fabian Rodriguez

To what extent was your testing approach aligned to the software requirements? Support your claims with specific evidence.

I created a JUnit test to test the char length. The test ran successfully and found that the char length's length was greater than or equal to the number of characters per the requirements.

Below is block of code for the evidence:

@Test

public void testUpdate() {

TaskService cs = new TaskService();

Task test1 = new Task("Learn Java", "Learn the java programming language.");

Task test2 = new Task("Learn Python", "Learn the python programming language.");

Task test3 = new Task("Learn Rust", "Learn the rust programming language.");

cs.addTask(test1);

cs.addTask(test2);

cs.addTask(test3);

assertTrue(cs.updateTask(test1.getID(), "Learn Dart", "Learn the dart programming language."));

assertFalse(cs.updateTask(test2.getID(), "Learn C++", "Learn the C++ programming language.Learn the C++ programming language.Learn the C++ programming language."));

}

* Defend the overall quality of your JUnit tests for the contact service and task service. In other words, how do you know that your JUnit tests were effective on the basis of coverage percentage?

The quality of the test is good, although, it can be better if I had tested a helper function. The test coverage would have been over 90%.

* How did you ensure that your code was technically sound? Cite specific lines of code from your tests to illustrate.

To support testing and enforce the efficiency I created a function that takes a field and a max length that enforces the requirements and facilitates the testing. The function returns a Boolean value that helps us understand if a field is valid or not.

public boolean isValid(String field, int maxLength) {

return !(field.equals("") || field.length() > maxLength);

}