Intro to Testing

JUNE 30, 2021

What is Testing?

Within the realm of software:

- What does it mean to perform testing?
- What does testing entail?
- What kinds of testing are there?
- How do you test your work?
- How important is it to test and why?

Levels of Testing

1. Unit Testing
   - Testing of individual components
2. Integration Testing
   - Testing that components work together
3. System Testing
   - Testing the entire system
4. Acceptance Testing
   - Testing done typically by the client

- Example?

Test Code Coverage

- What does code coverage mean? What kinds are there? Why is it important?
- Commonly used coverage criteria:

1. **Function coverage** – Has each function in the program been called?
2. **Statement coverage** – Has each statement in the program been executed?
3. **Branch coverage** – Has each branch of each control structure (such as *if* and *case* statements) been executed? For example, given an *if* statement, have both the true and false branches been executed?
4. **Condition coverage** (or predicate coverage) – Has each Boolean sub-expression evaluated both to true and false?

    e.g. if ((x > 0) && (y > 0))

Source: https://en.wikipedia.org/wiki/Code_coverage

Testing Paths

Happy path is a scenario where all provided input is as expected and no error conditions occur.

Unhappy path is any scenario in which error conditions are encountered, such as:

- User provides text input when numbers are expected
- The specified file cannot be found or is locked
- The network is unavailable
- The database doesn’t exist or is corrupted
- The user has insufficient permissions
- etc…

Automated vs Manual Testing

Automated tests are easier to re-run but take longer to write up-front

Manual tests are easier to do the first time but take more effort than automated tests in subsequent runs

Realistically, you should automate as much testing as possible, but there will almost always be some amount of manual testing (particularly at the Acceptance Testing level)

Writing Tests

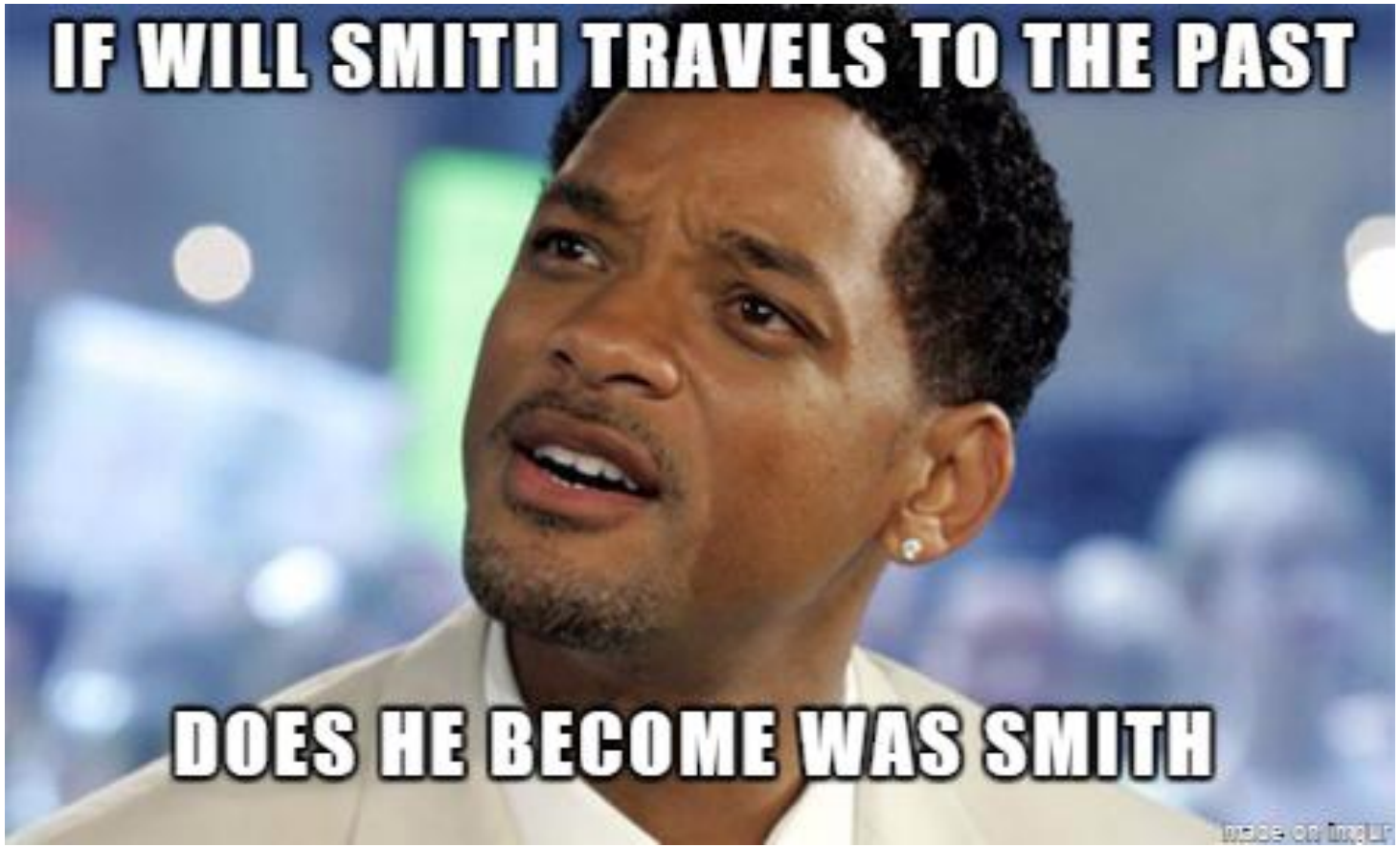
1. Who is responsible for writing tests?
   - It can depend on the organization, but it is good practice for you (the developer) to write tests for your code.
   - There are some organizations who have dedicated test engineers whose entire job is to create tests. This role can sometimes be called a QA or Quality Assurance engineer (though generally QAs have less experience than test engineers).
   - Sample QA job description: https://jobs.intel.com/ShowJob/Id/2895220/Software-QA-Engineer

2. When should tests be written?
   - What students typically do → Write tests last minute before turning in the assignment or not at all.
   - A better approach is to write tests for any new code before committing it to the repository.
   - Another approach is called Test Driven Development (TDD) → Write the tests during the design phase based on the requirements.

Today's Activity

- Modularize your code into functions that can be unit tested.
  - Consider dividing your code in a way that makes sense, but also so that each module has a discrete, testable purpose.
- You will **not** be writing unit tests in this activity; that comes later today. For now, you are preparing to write unit tests by making your code more testable.
- You will write comments in your code that identify the kinds of unit tests you will write later today.

- You will go into your TA's breakout room and work on the activity.
- Feel free to work with others in your room and ask questions of the group.
- The TA is there to assist you. If you want to ask me a question, come back to the main zoom room.



Questions?