Introduction to Quality Assurance

Divyansh Rajesh Jain

UCDAVIS

What is Quality Assurance?

- Verify that a software is up to a quality standard
  - “Correctness”
  - “Efficient”
  - “Usable”
- What other things are required to ensure quality?

Traditional Approaches to QA

- Manual Testing → Human effort to test software
- Requirement Tracing → Map feature requirements to actual software features
- What is the problem with these?

State Of Art Approach to QA

- Automated Testing for “correctness” and requirement tracing
  - How do you guarantee “correctness”
- What are the pros/cons of automated testing?

Automated Testing Paradigm

Workflow:

- Write Test Cases test for all features and edge cases in the code
- Claim: If I pass all test cases, I have achieved “correctness” and fulfilled requirements

Example Code

```cpp
#include <iostream>
#include <vector>
#include <cstdlib>
#include <ctime>

std::vector<int> foo(int index, int val) {
    std::vector<int> buffer(10, 42);
    int rareTrigger = std::rand() % 100000;

    if (rareTrigger == 99999) {
        buffer[15] = 100;
    } else {
        buffer[index] = val;
    }
    return buffer;
}

int main() {
    std::srand(std::time(0)); // Seed for
rand
    vector<int> answer = foo(8, 100);
    //More computations on the result of answer
    return 0;
}
```

UCDAVIS

Automated Testing Paradigm?

Workflow:

- Write Test Cases test for all features and edge cases in the code
- Claim: If I pass all test cases **with 100% coverage**, I have achieved “correctness” and fulfilled requirements

Different Types of Testing

- Unit Testing → Test specific features in isolation
  - Ex: Testing each function in isolation
- Integration Testing → Test features together
  - Ex: Testing dependent functions together
- Scaling Testing → Test features with high load

Overheads to Automated QA

- Software tests software
  - Engineering team (expensive) to test software
- Who tests the tester?
  - What if there is a bug in the testing software?
- Can you guarantee “quality”?

Thank You!

Next Time: Unit Testing w/ Hands on Activity

UCDAVIS