Jonas Seidl

Summary MIT Testing

1) Validation

Validation is a term that encompasses testing, as well as formal reasoning and code review. Formal reasoning is about making proofs that a program works, usually done for smaller, mission critical parts of an application. Code review is about another person looking over your code to find bugs and testing will be explained in more detail here. Testing is an inherently hard problem, because it is near impossible to test all cases, this means that you must choose test cases very carefully. You also need the right mindset, you cannot treat it as something fragile, you mostly cannot break programs irreparably, so hit it as hard as you can.

2) Test-first Programming

This is a method to fix bugs right as they are implemented. You write test cases based on a specification of the program and write the code after that and test it immediately. This should find most bugs right when they are created.

3) Partitioning

This concept for choosing test cases is basically separating input values in groups that should have the same behavior. You must find these groups. A basic example for a numeric value would be <0, 0, >0 and Null. These partitions oftentimes have the same behavior, but this is only a basic example and you need to find the right partitions for your specific program. When you have multiple inputs, it is recommended that you use a full cartesian product when possible. Simply put, this means that you test the program with every partition-combination possible. Continuing with the example, let us say we have 2 numeric input values, each partitioned into the 4 partitions. We would then need to test 4x4=16 combinations, so <0 and <0, <0 and 0, …

4) Documenting Testing Strategy

For each function you should document the strategy you used for testing the program (in comments for example). This would include any partitions you made and any other relevant information.

5) Coverage

Coverage means what places in the code the test cases reached. There are 3 fundamental types:

a. **Statement Coverage**
   As the name suggests, this only checks the percentage of statements run by the test.

b. **Branch coverage**
   This checks if every if and while statements are taken in every possible direction.

c. **Path coverage**
   This checks if every possible path in the program has been taken.

6) Unit Testing and Integration Testing

a. **Unit Testing**
   Unit Testing means testing each method/class (=module) in isolation. You give it an input and you except an output.

b. **Integration Testing**
   Integration Testing means testing multiple modules together, i.e. giving the output value form one module to another and seeing what happens.