

Test Code Coverage

EECS 2311 - Software Development Project



Tuesday, February 6, 2018

When is testing done?

- Short answer: Never!
- A bit longer answer: When all features of the system have been tested with all possible inputs that could make a difference
- In practice, this is hard to determine
- Metrics such as code coverage can be used to give an idea of how sufficient the testing is

Statement Code Coverage

- Observe the system as it is running
- Keep track of how many of the statements in the code were executed at least once
- Divide by the total number of statements in the system
- A comprehensive test suite is important
- Typically, it is hard to get high coverage. Anything above 70% is pretty good for a large system

Problems with statement coverage

- A statement must be executed with different values for the relevant variables to be fully tested
- Loop bodies may need to be iterated many times to reveal issues
- Not all statements are equally important
- Only the true branch of an if statement may be executed but coverage may be 90% for the statement if the false branch is one tenth of the size

Other kinds of coverage

- Segment coverage
- Branch coverage
- Multi-condition coverage
- Dataflow coverage
- More in EECS 4313

Software engineering guideline

- Low code coverage indicates that more testing must be done
- High code coverage gives little information about the quality of the testing

Let's see a demo (EclEmma)...

Lab Task

- Install EclEmma and calculate the coverage of your testing for the Authoring app
- Demonstrate your ability to calculate coverage in the lab
- Your midterm and final submission must discuss test code coverage (Testing document)