Review 10: Week of Mar 31

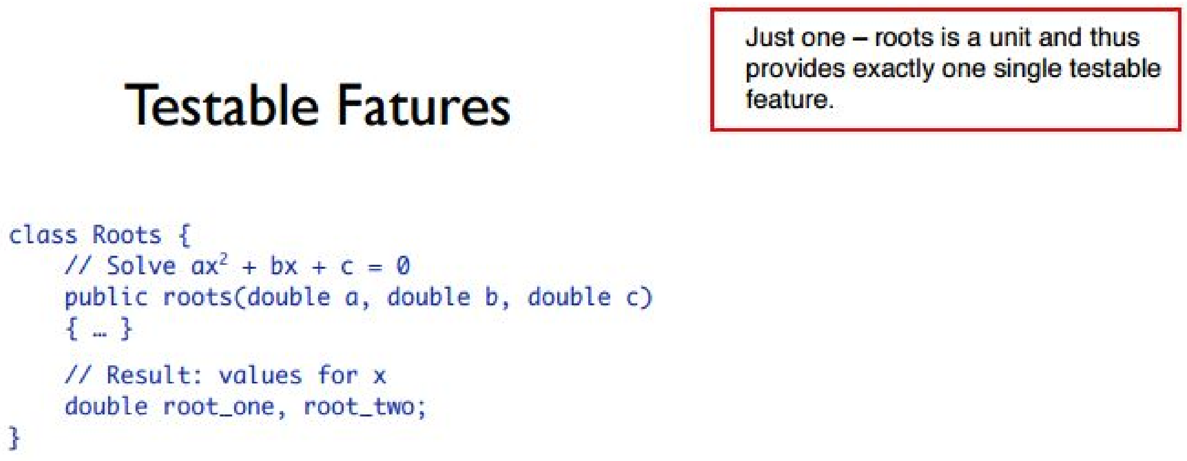
Testing

1. Define functional or black box testing.
   Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance.
2. Define structural or white box testing.
   White-box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality.
3. State three differences between functional and structural testing.
   1) Functional testing is based on specification, and structural testing is based on code;
   2) Functional Test covers as much specific behavior as possible, while structural testing covers as much implemented behavior as possible;
   3) Functional testing applies at all granularity levels, while structural testing only applies to unit and integration testing.
4. What is the state explosion problem? Explain in the context of testing. What strategy can be adopted to tackle this problem?
   Model checking tools face a combinatorial blow up of the state-space, commonly known as the state explosion problem, that must be addressed to solve most real-world problems.
   Strategy: functional testing (I think so, too)
5. What is random testing?
   Random testing is a black-box software testing technique where programs are tested by generating random, independent inputs. Results of the output are compared against software specifications to verify that the test output is pass or fail.
6. What are the advantages of random testing?
   - It is cheap to use: it does not need to be smart about the program under test.
   - It does not have any bias: unlike manual testing, it does not overlook bugs because there is misplaced trust in some code.
   - It is quick to find bug candidates: it typically takes a couple of minutes to perform a testing session.
   - If software is properly specified: it finds real bugs.

7. What are the disadvantages of random testing?

- It only finds basic bugs (f.ex. Null pointer dereferencing).
- It is only as precise as the specification and specifications are typically imprecise.
- It compares poorly with other techniques to find bugs (f.ex. static program analysis).

8. What are independent testable features? Give example from a multifunction calculator.
    1) Every single function becomes an independently testable feature.
    2) Example:



Testable Fatures

Just one – roots is a unit and thus provides exactly one single testable feature.

```
class Roots {
    // Solve ax² + bx + c = 0
    public roots(double a, double b, double c)
    { … }

    // Result: values for x
    double root_one, root_two;
}
```

9. What are representative values?

10. What are equivalent classes? How can we select representatives from equivalence classes?
    Divided based on the desired / specified functionality of the subject under test.

11. What are the heuristics for prioritizing tests?

Heuristics for testing

1. What are the heuristics for detecting bug in code? State three and explain in detail.
2. What is Gunes Koru's heuristic for detecting bugs by inspecting less code? Try to justify why the Koru method might work.
    smaller methods have disproportionately more methods that larger methods.
    Manul up: Sort code smallest to largest, Until you run out of time, read in that order.

3. What are different inspection methods used for detecting bug in code? State and explain each.
    1) Heavyweight inspections
       The process should have entry criteria that determine if the inspection process is ready to begin. This prevents unfinished work products from entering the inspection process.

The stages in the inspections process are: Planning, Overview meeting, Preparation, Inspection meeting, Rework and Follow-up. The Preparation, Inspection meeting and Rework stages might be iterated.
The process is ended by the moderator when it satisfies some predefined exit criteria. The term inspection refers to one of the most important elements of the entire process that surrounds the execution and successful completion of a software engineering project.

2) Lightweight inspections

- After inspecting M artifacts, generate G groups for the different kinds of defects seen
- Now, for the next N, each reviewer only looks items in one group
- Each reviewer works faster, in more depth

4. Define all pairs. Given limitations in human cognitive ability and finite testing budgets, justify why all-pairs might be an interesting test strategy.
    1) **all-pairs testing** or **pairwise testing** is a combinatorial method of software testing that, for *each pair* of input parameters to a system (typically, a software algorithm), tests all possible discrete combinations of those parameters.
    2) The most common bugs in a program are generally triggered by either a single input parameter or an interactions between pairs of parameters.Bugs involving interactions between three or more parameters are both progressively less common and also progressively more expensive to find---such testing has as its limit the testing of all possible inputs.Thus, a combinatorial technique for picking test cases like all-pairs testing is a useful cost-benefit compromise that enables a significant reduction in the number of test cases without drastically compromising functional coverage.
5. Show the error in the following output for all pairs of 5 booleans. Explain why it is an error and fix the error. (Note: in the following "0" means "don't care")

```
((2 2 1 1 1)
(2 1 2 2 2)
 (1 2 2 1 2)
 (1 1 1 2 1)
 (0 2 2 2 2) wrong, should be 1.
 (0 1 1 1 2))
```