Kas's Software Engineering Practice Test of Doom

Please note any answers in () Typically are not part of the answer rather are added context or a joke.

1 Testing

1. What is the cost of testing?
   At least half of your development budget will be spent on testing.

2. Is not testing a great way of saving money?
   No, infact it can actually cost more money. Debugging later and later into a project tends to cost more due to the 1-10-100 rule.

3. What are the four types of activities testing can be broken up into and what type of knowledge is required?
   Test design: Designing test values to satisfy coverage criteria or other engineering goals. Requires knowledge of discrete math, programming, and testing.
   Test automation: Embed test values into executable scripts. Requires knowledge of scripting.
   test execution: run tests on software and record the results. Requires little knowledge.
   Test evaluation: Evaluate results of testing and report to developers. Requires domain knowledge.

4. What is validation in testing?
   The process of evaluating software at the end of software development to esnure compliance with intended usage.

5. what is verification in testing?
   The process of determing whether the products of a given phase or software development process fulfills the requirement established during the previous phase.

6. What is a failure in testing.
   Any deviation of the observed behavior from the specified behavior.

7. what is an error in testing?
   The system in a state such that further processing by the system will cause a failure.

8. What is a fault in testing?
   The mechanical or algorithmic cause of an error. Also known as a bug.

9. what are some ways of dealing with errors? verification, Modular redundancy, Declaring a bug to be a feature, Patching, and testing.

10. What is testing? Finding inputs that cause the software to fail.

11. What is debugging? The process of finding a fault given a failure.

12. what is a test case? A set of test inputs, execution conditions, and expected results. Test inputs are the values that directly satisfy one test requirement.

13. What is black box testing?
    Focusing on I/O behavior. If we are able to predict the output for any given input. Typically impossible to generate all possible inputs.

14. What is the goal of black box testing?
    Reduce the number of test cases by equivalence partitioning. Divide input conditions into equivalnce classes and choose a representative for each class.
    equivalnce classes are determined through coverage and being disjoint.

15. What is white box testing?
    Testing with the goal of Thoroughness/coverage. Every statement in the component is executed at least once.

16. What are the types of White-box testing? Statement testing, Loop testing, Path Testing, Branch Testing.

17. What is static Analysis?
    Analysis done without running a program.

18. what are the types of static Analysis?
    Hand execution (reading the source code), Walk-Through (information presentation), Code inspection (Formal presentation), Automated tools for checking syntatic and semantic errors or a departure from coding standards.

19. What id dynamic Analysis?
    Testing done with running a program.

20. What are the types of dynamic analysis?
    Black-Box testing and White-box testing.

21. Is it possible to completely test any abritary system?
    No, you'd have to solve the halting problem and even if it was solved it'd cost a lot of time and money.

22. What are the 4 testing steps?
    Select what needs to be measured, Decide how the testing is done, develop test cases, and create a test oracle.

23. What is a test oracle?
    The set of predicted results for a set of test cases. It must be written down before the testing occurs.

24. What is unit testing?
    The testing of a unit. (Quite helpful I guess...)

25. What is a unit? A module or a small set of modules. A few examples, classes or interfaces.

26. Why bother with unit testing?
    It's practical and allows a divide-and-conquer approach. Splitting systems into units is very helpful and narrow down where bugs are. You don't want to chase down bugs in other units. Support regresion testing allows you to make changes to code and know if you broke something. Improves

confidence that changing one many things doesn't break everything (yay!).

27. How can one do unit testing?
Build systems in layers. Then test upwards. Start with classes that don't depend on others and countinue testing building on already tested classes.

28. Benefits of unit testing? Avoid having to write stubs. When testing a module it depends on verified reliable modules.

29. What are some unit testing Heurisitcs?
Create unit tests as soon as object design is completed. Develop the test cases. Cross-check the test cases to eliminate duplicats. Desk check your source code. Create a test harness. Execute the test cases. Compare the results of the test with your orcale.

30. What are the parts of test code?
The test fixture, test driver, and test oracle.

31. What is the test driver?
The class that runs the tests.

32. What is the test fixture?
Set of variables used in testing.

33. What is the intergration testing strategy?
The entire system is viewed as a collection of subsystems. Note: The order which subsystems are selected for testing and intergration determines the specific strategy.

34. What is system testing?
Ensure that the complete system compiles with the functional and nonfunctional requirements. Functional testing, Perforamnce testing, Acceptance testing, and installation testing are all different types of system testing.

35. What is the impact of requirements on system testing?
The more explicit the measurements the easier they are to test. Quality of use cases determines ease of functional testing. Quality of subsystem determines the ease of structure testing. Quality of nonfunctional requirements and constraints determine the ease of perforamnce tests.

36. What are test requirements?
Specific thigns that must be satisfied or covered during testing.

37. What are test criterion?
A collection of rules and a process that defines test requirements.

38. What is another way of describing coverage in testing. (Hint think of test sets).
Given a set of test requirements X for coverage criterion C, a test set T satisfies C coverage iff for every test requirement in X there is at least one test in T it is satisfied by a test requirement.

39. What is criteria subsumption? A test criterion C1 subsums C2 iff every set of test cases that satisfies criterion C1 also satisfies C2. (basically subsets)

40. what are some criteria structures?
    Graphs, Logical expresions, Input domain characterization, syntatic structures.

41. What is predicate coveage in Logical expresions?
    Each predicate must be True or False.

42. What is closure Coverage in Logical expresions?
    Each clause much be true or false.

43. Combinatorial coverage in Logical expresions?
    Various combinations of clauses.

44. What is active caluse coverage?
    Each caluse must determine the predicate's result.

45. What is input domain characterization?
    They describe the input domain of the software. Identify inputs, parameters, or other categorization. Partition each input into finite stes of representative classes. Choose combinations of values.

46. What are syntatic strcutres?
    Based on a grammar or other syntactic definition.

47. What is mutation testing and what is it an example of?
    When you introduce small changes tot he program and find tests that cause the mutant programs to fail. Failure is determined by a different output from the original program. Check the output of useful tests on the original program.

48. What is Model-Based testing.
    Derives tests from a model that describes some aspects of the system under a test. Model describes part of the behavior.

49. What is a coveage graph?
    The most commonly used structure for testing. Made up of graphs where tests are intended to cover the graph in some way.

50. What is a graph? A nonempty set Z of nodes. A set nonempty set X of initial nodes. A nonempty set of final nodes Y. A set E of edges where each edge connects one node to another.

51. What is a path in a graph?
    A sequence of nodes.

52. What is the length in a graph?
    The number of edges.

53. what is a subpath?
    A subsequence of nodes in p is a subpath of p.

54. What is reach in graphs?
    subgraph that can be reached from a node n.

55. What is a test path?
    A path that starts at an initial node and ends at a final node. Test paths represent execution of test cases.

56. what is an SESE graph?
    All test paths start at a single node and end at another node. Single-entry and single-exit.

57. What is visiting?
    A test path p visist a node if a node n is in p.

58. What is touring?
    A test path p tours a subpath q if q is a subpath of p.

59. what is Node coveage?
    Requires that each node and edge in a graph be executed. Test set T satisfies node coveage on a graph G iff for every syntatically reachable node n in N, there is some path in path(T) such that p visits n. (simple version Test requirements contain each reachable node in G.)

60. What is edge coverage?
    A slightly stronger system than node coverage. Test requirements contain each reachable path of length up to 1, inclusive, in G.
    Length up to 1 allows graphs with one and no edges.

61. When are Node coveage and edge coverage difference?
    When there is an edge and another subpath between a pair of nodes.
    Node coverage TR = $\{0, 1, 2\}$.
    Test path = $[0, 1, 2]$.
    Edge coverage TR = $\{0, 1, 2\}$.
    Test path = $[0, 1, 2], [0, 2]$.

62. What is edge pair coveage?
    Requires pairs of edges or subpaths of length 2. Test requirements contain each reachable path of length up to 2, inclusive, in G.

63. What is complete path coverage and where does it fail?
    Test requirements contain all paths in G. It fails whenever the program introduces a loop. Infiniately many TR paths arrise unfornately (might be solvable with the infinite monkey theorem assuming you have infinite monkeys...).

64. Specified Path coverage is?
    Test requirements contain a set S of test paths, where S is supplied as a parameter.

65. The following questions use the following graph.
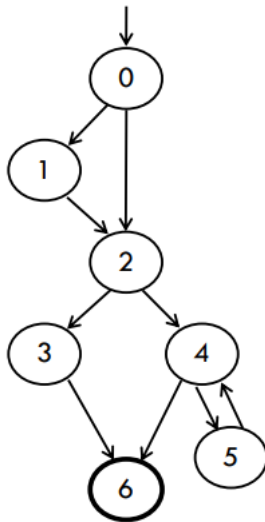
66. What are the test paths of the following image?

Node Coverage

TR = { 0, 1, 2, 3, 4, 5, 6 }

Test Paths: [0, 1, 2, 3, 6] [0, 1, 2, 4, 5, 4, 6]

67. What are the test paths of the following image?

Edge Coverage

TR = { (0,1), (0,2), (1,2), (2,3), (2,4), (3,6), (4,5), (4,6), (5,4) }

Test Paths: [0, 1, 2, 3, 6] [0, 2, 4, 5, 4, 6]

68. What are the test paths of the following image?

Edge-Pair Coverage

TR = { [0,1,2], [0,2,3], [0,2,4], [1,2,3], [1,2,4], [2,3,6], [2,4,5], [2,4,6], [4,5,4], [5,4,5], [5,4,6] }

Test Paths: [0, 1, 2, 3, 6] [0, 1, 2, 4, 6] [0, 2, 3, 6] [0, 2, 4, 5, 4, 5, 4, 6]

69. What are the test paths for Complete Path Coverage?
infinitely many paths exist unfornately... As long as a path reaches the end it'd be considered

70. What is a simple path?
If a given path from two nodes no node appears more than once except maybe the first and last nodes are the same.
There are no internal loops, include all other subpaths, a simple path.

6

71. What is a prime path?
A simple path that does not appear as a proper subpath of any other simple path.

72. What is prime path coverage?
A simple, elegant, and finite criterion that requires loops to be executed as well as skipped. Test requirements contain each prime path in G.
Will tour all paths of any lenghth. Subsumming node, edge, and edge-pair coverage.

73. Refering to the previous graph figure what are some prime paths?
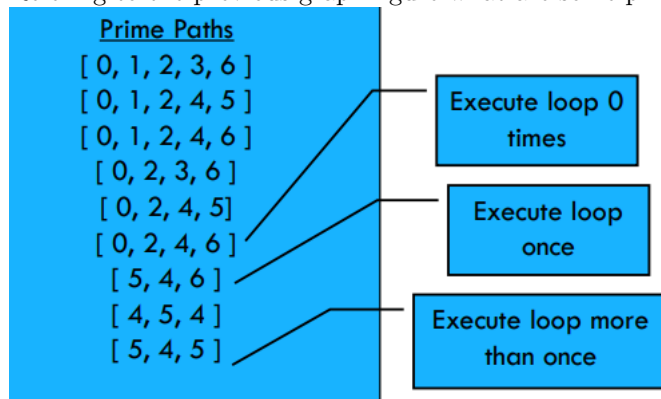
74. What is data flow criteria.
A way to ensure that values are computed and used correctly.

75. What are the parts of Data flow criteria?
Definition (def) the location where a value for a variable is stored into memory. Use a location where a variable's value is accessed. def (n) or def (e) the set of variables defined by node n or edge e. use (n) or use (e) the set of variables used by node n or edge e.

76. What is a DU pair?
A pair of locations (x, y) such that a variable v is defined at x and used at y.

77. What is Reach in a DU pair?
If there is a def-clear path from X to Y with respect to v, the def of v at x reaches the use at y.

78. what is Def-Clear?
A path with respect to variable v if v is not given another value on any of the nodes/edges in the path. No redefinitions.

79. What is a du-path? A simple subpath that is def-clear with respect to v from def to use.

80. What is graph coverage for source code?
The most common application is to the graph source. Made of graphs usually the control flow graph, Node coverage execute every statement, edge coverage execute every brach, loops looping structures, and data flow coverage augment the control flow graph.

81. What is a control flow graph.

    Modles all executions of a method by describing control structures. Made of nodes statements or sequences of statements, Edges transfers of control, and basic block a sequence of statements such that if the first statement, all statements will be (branchless). Typically annotated with extra information.