



Testing Terminology

Lecture Objectives:

- 1) Explain the difference between white box and black box testing.
- 2) Explain the difference between static testing and dynamic testing.
- 3) Define test requirement.
- 4) Define coverage criterion.
- 5) Define coverage and coverage level.
- 6) Define infeasibility.
- 7) Compare and contrast white and black box testing.
- 8) Compare and contrast top down and bottom up testing.
- 9) Compare and contrast static testing with dynamic testing.

What is wrong with the following terms?

- Complete testing
- Exhaustive Testing
- Full coverage

Synonymous ..
May not
be possible ..

Maybe meaningless
terms ..

Black Box versus White Box Testing



Black Box versus White Box

Testing



Black Box Testing and White

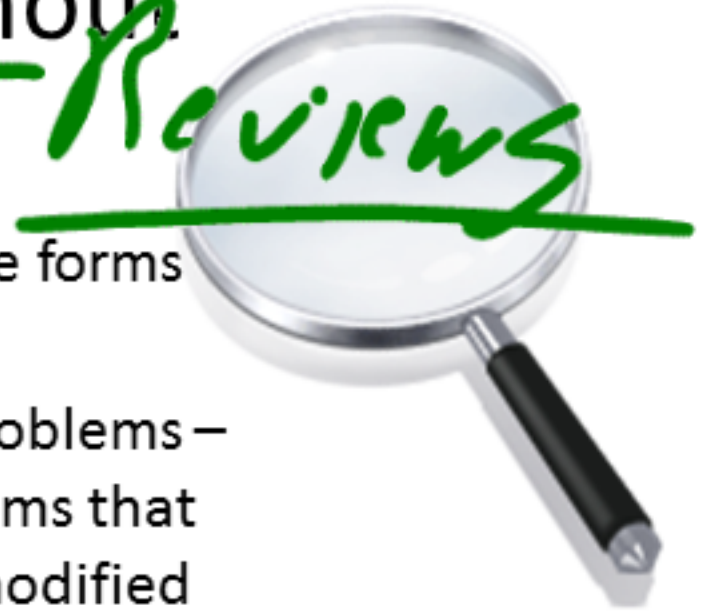
Box Testing

- Black box testing (also called functional testing) is testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions.
- *White-box testing is testing that takes into account the internal mechanism of a system or component (IEEE, 1990).*

for different criteria evaluating

- Static Testing : Testing without executing the program

- This include software inspections and some forms of analyses
- Very effective at finding certain kinds of problems – especially “potential” faults, that is, problems that could lead to faults when the program is modified



- Dynamic Testing : Testing by executing the program with real inputs

Actually running the program.



Top-Down and Bottom-Up

Testing
"other terms"

- Top-Down Testing : Test the main procedure, then go down through procedures it calls, and so on

Bottom-Up Testing : Test the leaves in the tree (procedures that make no calls), and move up to the root.

- Each procedure is not tested until all of its children have been tested



Start here
and work
up

Work
down



Test Design

100+ brands to discover



+

Coca-Cola
freestyle®

Definitions

- Test Requirement — *Meet something*
 - A test requirement is a specific element of a software artifact that a test case must satisfy or cover.
- Coverage Criterion — *map*
 - A rule or collection of rules that impose test requirements on a test set
- Coverage
 - Given a set of test requirements TR for a coverage criterion C, a test set satisfies C if and only if for every test requirement tr in TR, at least one test t in T exists such that it satisfies tr
- Coverage Level — *Measurement of test quality.*
 - Given a set of test requirements TR and a test set T, the coverage level is the ratio of the number of test requirements satisfied by T to the size of TR.
- Infeasible test requirements
 - test requirements that cannot be satisfied
 - No test case values exist that meet the test requirements
 - Dead code — *Bad / he f, d, or*
 - Detection of infeasible test requirements is formally undecidable for most test criteria

Test requirements

- TR1 = {base=Coke, base= Sprite, base=Fanta, base = Minute Maid Lemonade, base = Poweraid, base=Barq's Root Beer, base = Seagram's Ginger Ale}
- TR2={subtype=Regular, subtype=Zero, subtype=Caffeine Free}
- TR3={flavor=Regular, flavor = Orange, flavor = Lime, flavor=Raspberry, flavor=Cherry, flavor=Cherry Vanilla}

Main Flavors

Types

Bells and whistle flavors

Test Cases for Freestyle

- https://docs.google.com/document/d/1uFQRZYtwyyvW2wL6J2eVUcumg_OChHrDLE3vczadGyg/edit?usp=sharing

Life Savers



TR1={Flavor=Cherry, flavor=Raspberry, flavor=Watermelon,
flavor=orange, flavor=pineapple}

- TR2={Color=red, color=purple, color=green, color=orange,
color=yellow}

Cherry Red
Rasp purple
Wmelon green

ORANGE OR
Pineapple yellow

Criteria Subsumption

- Definition
 - A coverage criteria $C1$ subsumes $C2$ if and only if for every test set that satisfies criterion $C1$ is also satisfies $C2$.