Negative Testing

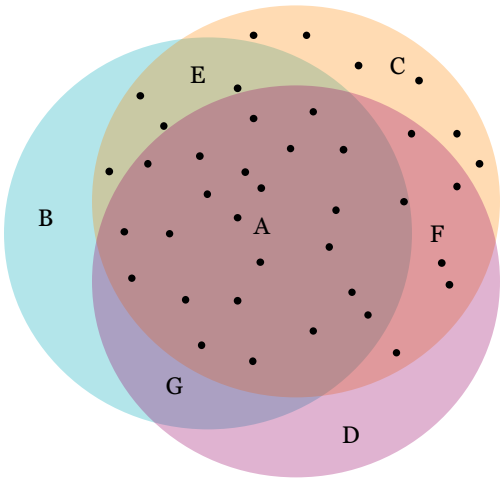
Adnan Causevic

In this Lecture

- Negative Testing
  - Fundamentals
  - Positive Test Bias
- Negative Testing Techniques
  - Outside specification
  - Invalid input?
  - Breaking the system

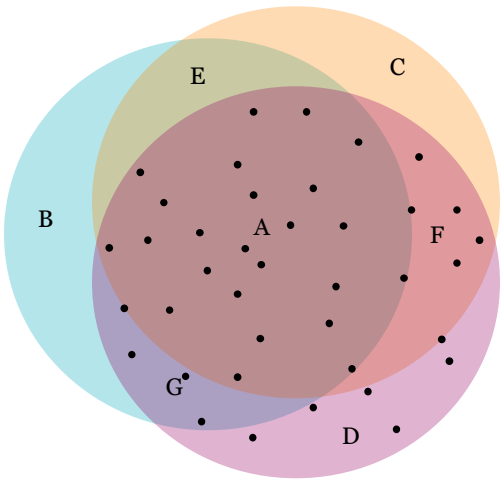
Specification-Based (Functional) Test Design

A: Desired, specified and implemented software behavior
B: Desired, but not specified or implemented software behavior
C: Specified, but not desired or implemented software behavior
D: Implemented, but not desired or specified software behavior
E: Desired and specified, but not implemented software behavior
F: Specified and implemented, but not desired software behavior
G: Desired and implemented, but not specified software behavior



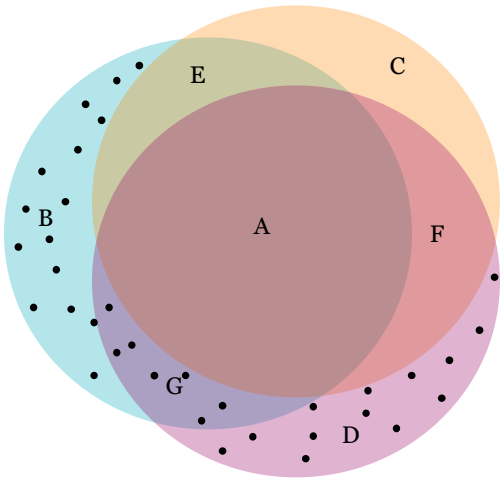
Implementation-Based (Structural) Test Design

A: Desired, specified and implemented software behavior
B: Desired, but not specified or implemented software behavior
C: Specified, but not desired or implemented software behavior
D: Implemented, but not desired or specified software behavior
E: Desired and specified, but not implemented software behavior
F: Specified and implemented, but not desired software behavior
G: Desired and implemented, but not specified software behavior



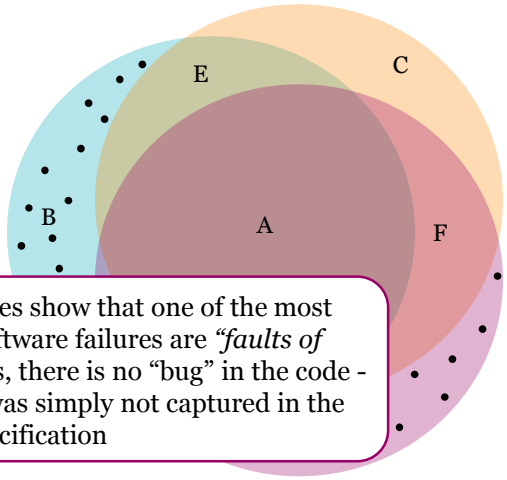
Negative Testing?

A: Desired, specified and implemented software behavior
B: Desired, but not specified or implemented software behavior
C: Specified, but not desired or implemented software behavior
D: Implemented, but not desired or specified software behavior
E: Desired and specified, but not implemented software behavior
F: Specified and implemented, but not desired software behavior
G: Desired and implemented, but not specified software behavior



Negative Testing?

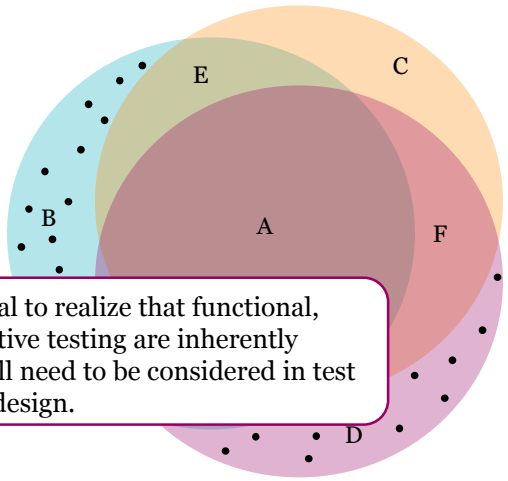
A: Desired, specified and implemented software behavior
B: Desired, but not specified or implemented software behavior
C: Specified, but not desired or implemented software behavior
D: Implemented, but not desired or specified software behavior
E: Desired and specified, but not
imp
F: S
but
G: I
not



Note that several studies show that one of the most common causes of software failures are *"faults of omission"*. In other words, there is no "bug" in the code - the desired behaviour was simply not captured in the specification

Negative Testing?

A: Desired, specified and implemented software behavior
B: Desired, but not specified or implemented software behavior
C: Specified, but not desired or implemented software behavior
D: Implemented, but not desired or specified software behavior
E: Desired and specified, but not imp
F: S
but
G: I
not s



Note also that it is vital to realize that functional, structural and negative testing are inherently **complementary** and all need to be considered in test design.

An Initial Experiment

(1, 2, 3) (1, 2, 3) (1, 2, 3) (1, 2, 3) (1, 2, 3) (1, 2, 3) (1, 2, 3) (1, 2, 3)
(6, 4, 2) (6, 4, 2) (6, 4, 2) (6, 4, 2) (6, 4, 2) (6, 4, 2) (6, 4, 2) (6, 4, 2)
(9, 12, 4) (9, 12, 4) (9, 12, 4) (9, 12, 4) (9, 12, 4) (9, 12, 4) (9, 12, 4) (9, 12, 4)
(0, 1, 16) (0, 1, 16) (0, 1, 16) (0, 1, 16) (0, 1, 16) (0, 1, 16) (0, 1, 16) (0, 1, 16)
(13, 4, 1) (13, 4, 1) (13, 4, 1) (13, 4, 1) (13, 4, 1) (13, 4, 1) (13, 4, 1) (13, 4, 1)
(1, 2, 3) (1, 2, 3)
(6, 4, 2) (6, 4, 2)
(9, 12, 4) (9, 12, 4)
(0, 1, 16) (0, 1, 16)
(13, 4, 1) (13, 4, 1)

I need three volunteers

(1, 2, 3) (1, 2, 3) (1, 2, 3) (1, 2, 3) (1, 2, 3) (1, 2, 3) (1, 2, 3) (1, 2, 3)
(6, 4, 2) (6, 4, 2) (6, 4, 2) (6, 4, 2) (6, 4, 2) (6, 4, 2) (6, 4, 2) (6, 4, 2)
(9, 12, 4) (9, 12, 4) (9, 12, 4) (9, 12, 4) (9, 12, 4) (9, 12, 4) (9, 12, 4) (9, 12, 4)
(0, 1, 16) (0, 1, 16) (0, 1, 16) (0, 1, 16) (0, 1, 16) (0, 1, 16) (0, 1, 16) (0, 1, 16)
(13, 4, 1) (13, 4, 1) (13, 4, 1) (13, 4, 1) (13, 4, 1) (13, 4, 1) (13, 4, 1) (13, 4, 1)

An Initial Experiment

- **Experiment Background:** You are asked to identify a rule applying to triplets of integers.
  - Given any triplet (x, y, z), the triplet can fit the rule or not.
  - The triplet (2, 4, 6) fits the rule.

- **Experiment Procedure:**
  1. You hypothesize a suggestion for what the rule could be and write it down (don't say it out loud)
  2. For five rounds, do the following
     A. Has your hypothesized rule changed? If so, write the new hypothesized rule down
     B. Test your rule by providing a triplet. I will let you know if the triplet fits my rule or not
  3. State your proposal for the rule

An Initial Experiment

- **Experiment Background:** You are asked to identify a rule applying to triplets of integers.
  - Given any triplet (x, y, z), the triplet can fit the rule or not.
  - The triplet (2, 4, 6) fits the rule.

Actual rule?

1. You hypothesize a suggestion for what the rule could be and write it down (don't say it out loud)
2. For five rounds, do the following
   A. Has your hypothesized rule changed? If so, write the new hypothesized rule down
   B. Test your rule by providing a triplet. I will let you know if the triplet fits my rule or not
3. State your proposal for the rule