CSC 5524 : Quality, Metrics, Tests, Process

J Paul Gibson, D311

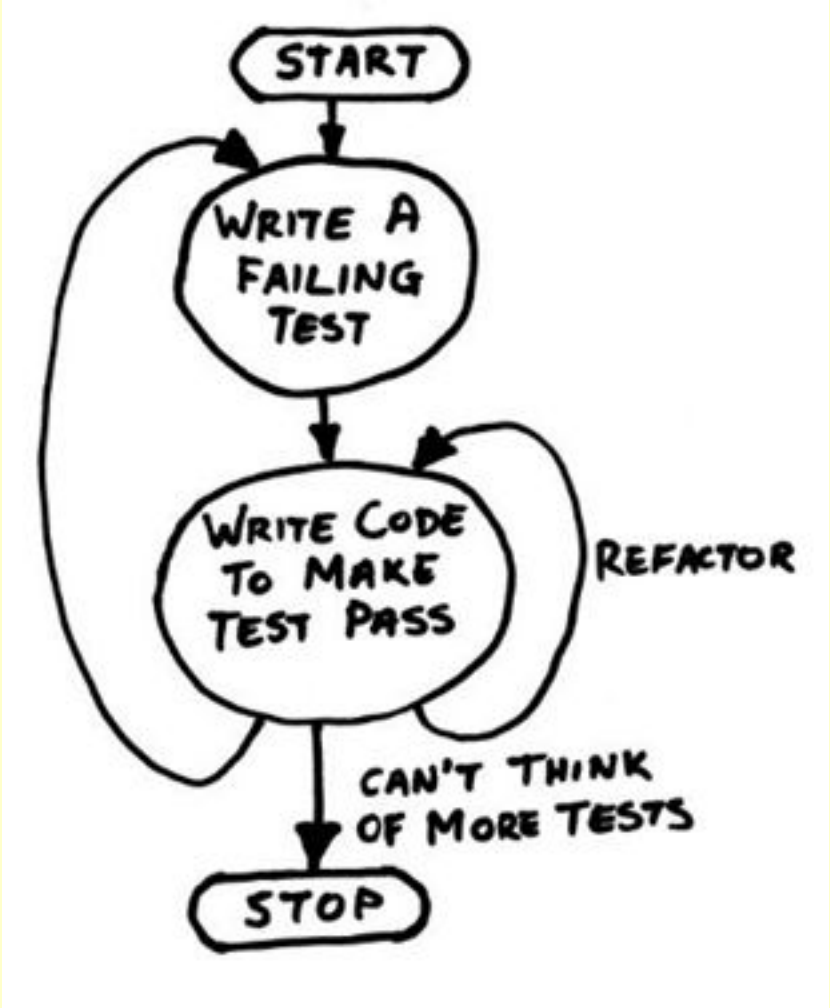
paul.gibson@int-edu.eu

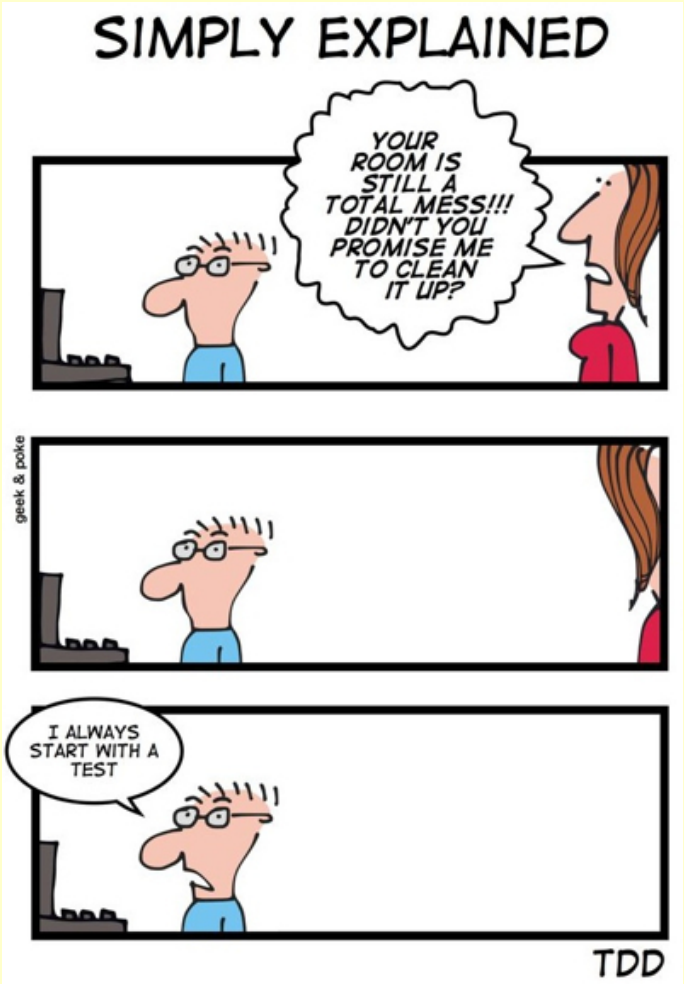
http://www-public.it-sudparis.eu/~gibson/Teaching/CSC5524/

TDD: The 'Matches' Problem

…/~gibson/Teaching/CSC5524/CSC5524-TDD-Problem.pdf

Implementing A Solution: Test Driven Development

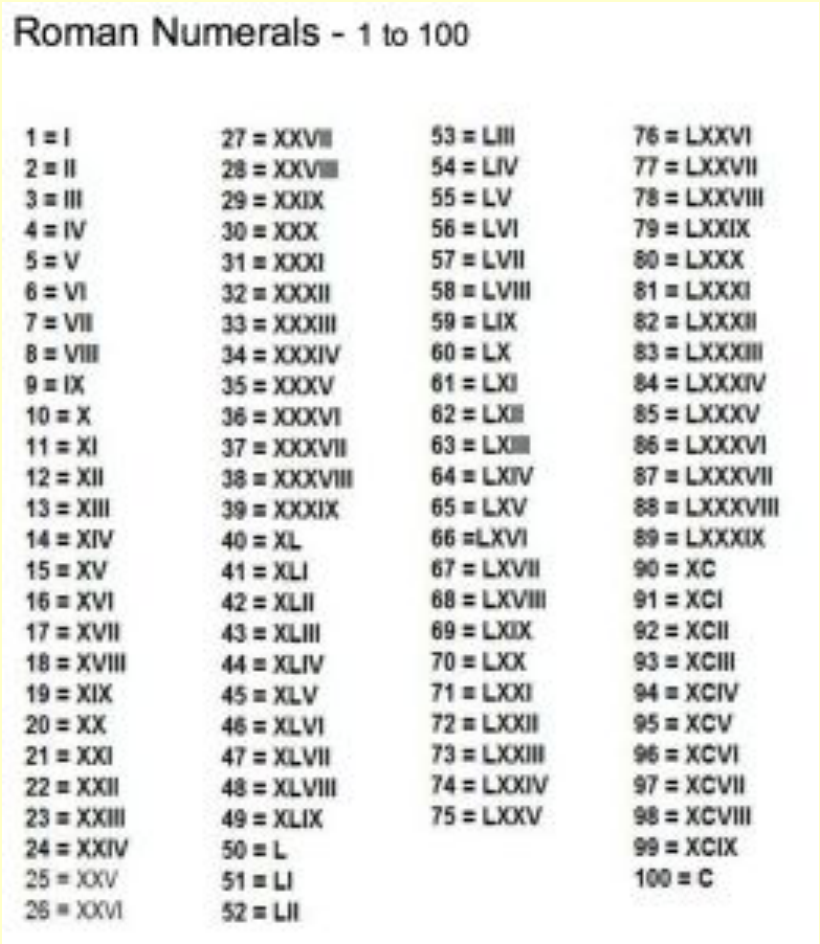




An Example of Test Driven Development

```
public static String arabicToRoman(int arabic)
```

https://remonsinnema.com/2011/12/05/practicing-tdd-using-the-roman-numerals-kata/



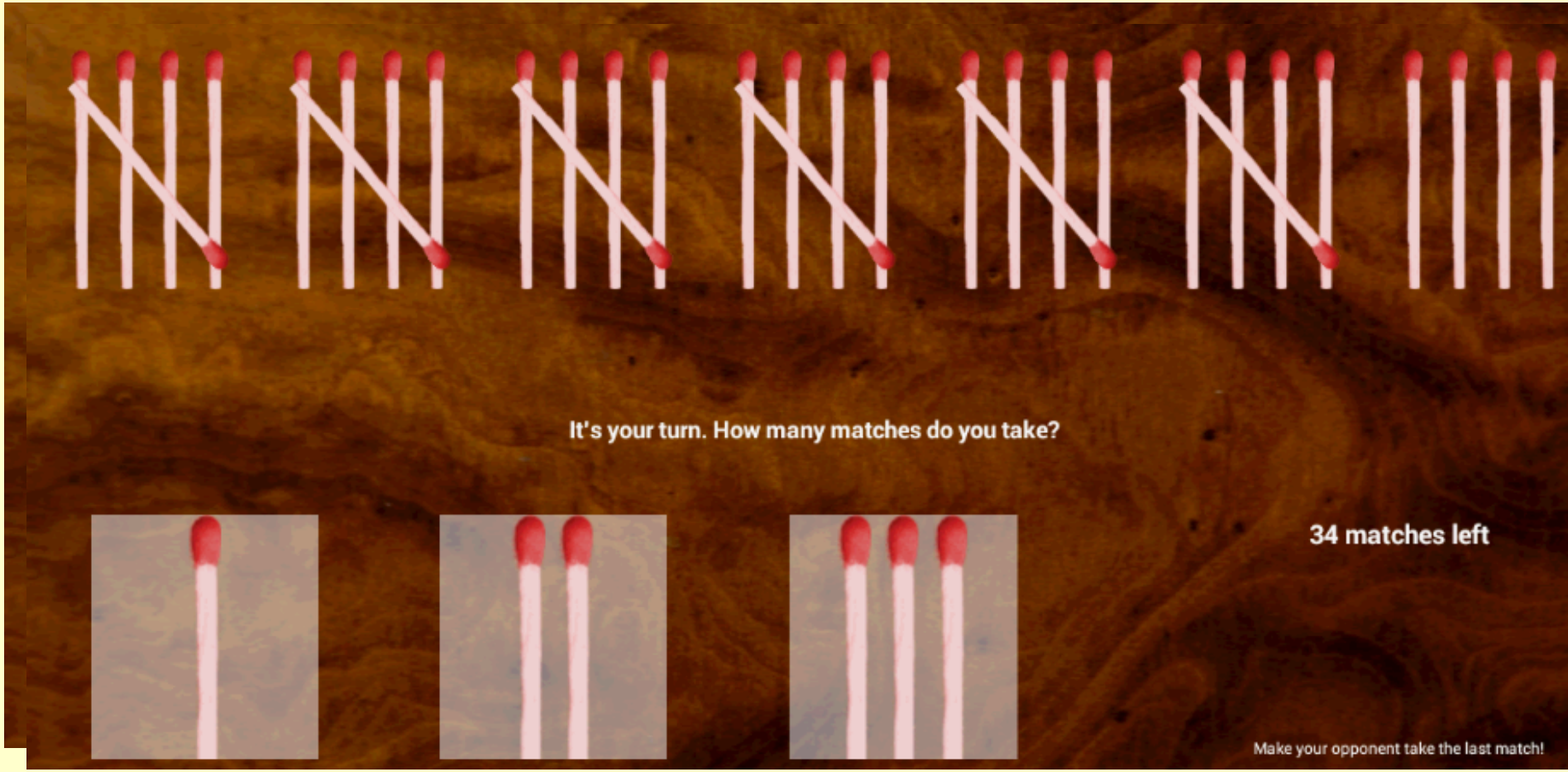
Roman Numerals - 1 to 100

1 = I	27 = XXVII	53 = LIII	76 = LXXVI
2 = II	28 = XXVIII	54 = LIV	77 = LXXVII
3 = III	29 = XXIX	55 = LV	78 = LXXVIII
4 = IV	30 = XXX	56 = LVI	79 = LXXIX
5 = V	31 = XXXI	57 = LVII	80 = LXXX
6 = VI	32 = XXXII	58 = LVIII	81 = LXXXI
7 = VII	33 = XXXIII	59 = LIX	82 = LXXXII
8 = VIII	34 = XXXIV	60 = LX	83 = LXXXIII
9 = IX	35 = XXXV	61 = LXI	84 = LXXXIV
10 = X	36 = XXXVI	62 = LXII	85 = LXXXV
11 = XI	37 = XXXVII	63 = LXIII	86 = LXXXVI
12 = XII	38 = XXXVIII	64 = LXIV	87 = LXXXVII
13 = XIII	39 = XXXIX	65 = LXV	88 = LXXXVIII
14 = XIV	40 = XL	66 =LXVI	89 = LXXXIX
15 = XV	41 = XLI	67 = LXVII	90 = XC
16 = XVI	42 = XLII	68 = LXVIII	91 = XCI
17 = XVII	43 = XLIII	69 = LXIX	92 = XCII
18 = XVIII	44 = XLIV	70 = LXX	93 = XCIII
19 = XIX	45 = XLV	71 = LXXI	94 = XCIV
20 = XX	46 = XLVI	72 = LXXII	95 = XCV
21 = XXI	47 = XLVII	73 = LXXIII	96 = XCVI
22 = XXII	48 = XLVIII	74 = LXXIV	97 = XCVII
23 = XXIII	49 = XLIX	75 = LXXV	98 = XCVIII
24 = XXIV	50 = L		99 = XCIX
25 = XXV	51 = LI		100 = C
26 = XXVI	52 = LII		

A Test Driven Development Problem

Matches Game

https://play.google.com/store/apps/details?id=net.con_data.android.nim



A Test Driven Development Problem

Matches Game

Following a TDD approach, create an AI player that plays the game as well as possible (always winning if a win is possible):

starting position 1
starting position 2
…

starting position n

At each iteration keep the code as simple as possible.
How many iterations do you need before the code works for all possible starting positions?

Testing - additional information

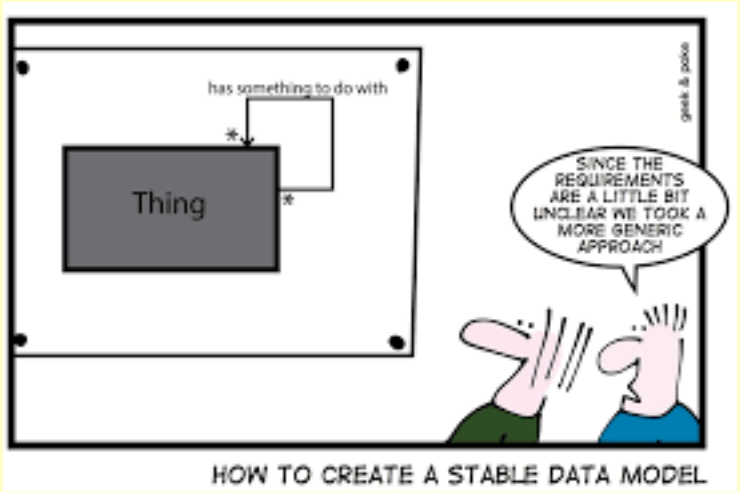
Test Types

1. Black Box
2. White box
3. Unit
4. Integration
5. System
6. Validation/ Acceptance
7. Usability
8. Security
9. Regression
10. Stress/Robustness/Fault Tolerance
11. Performance
12. Simulation and Random Testing
13. Complementary/Advanced Activities

Testing - additional information

Black Box

"I think you misunderstood me when I said I wanted our factory to go all green."

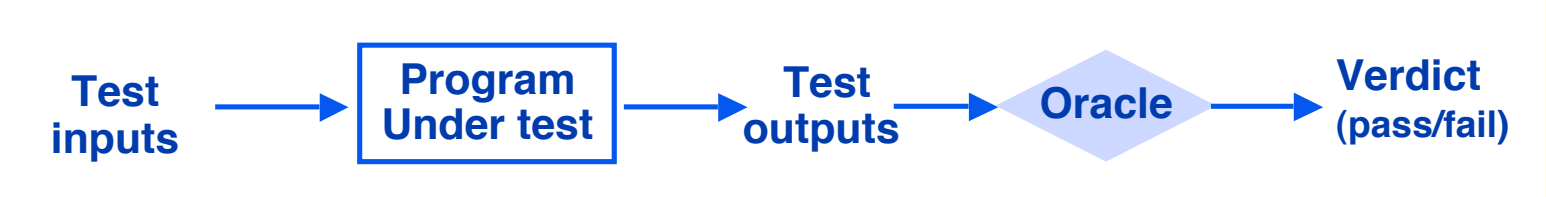


Requirements/Specification are critical to black box testing

Alternatively, the tests are your specifications … but how to validate the tests?

Testing - additional information

Black Box

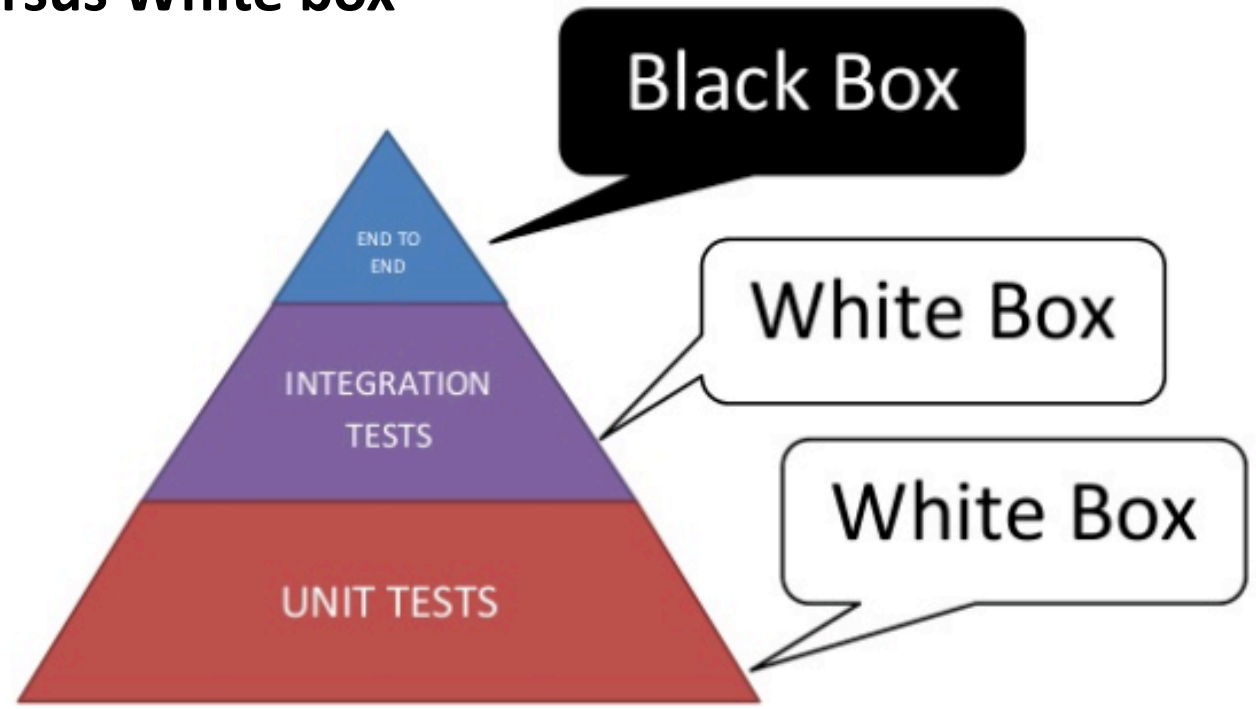


Oracle problem = how to decide about the correctness of the observed outputs?

Manual computation of expected results, executable specification, back-to- back testing of different versions, output plausibility checks, ...

Testing - additional information

Black versus White box

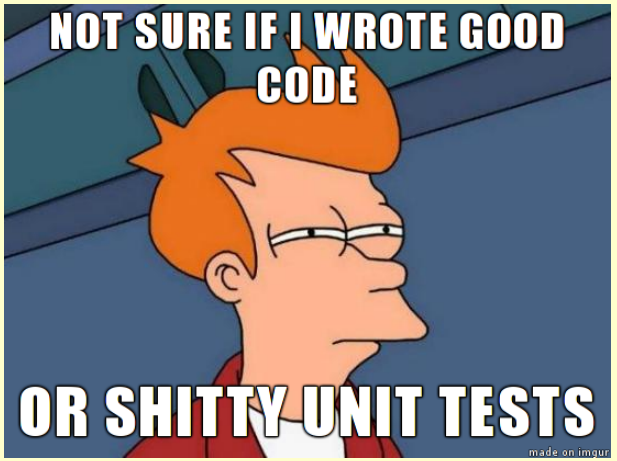


Traditional Pyramid View

Question: can/should we do black box testing in all phases?

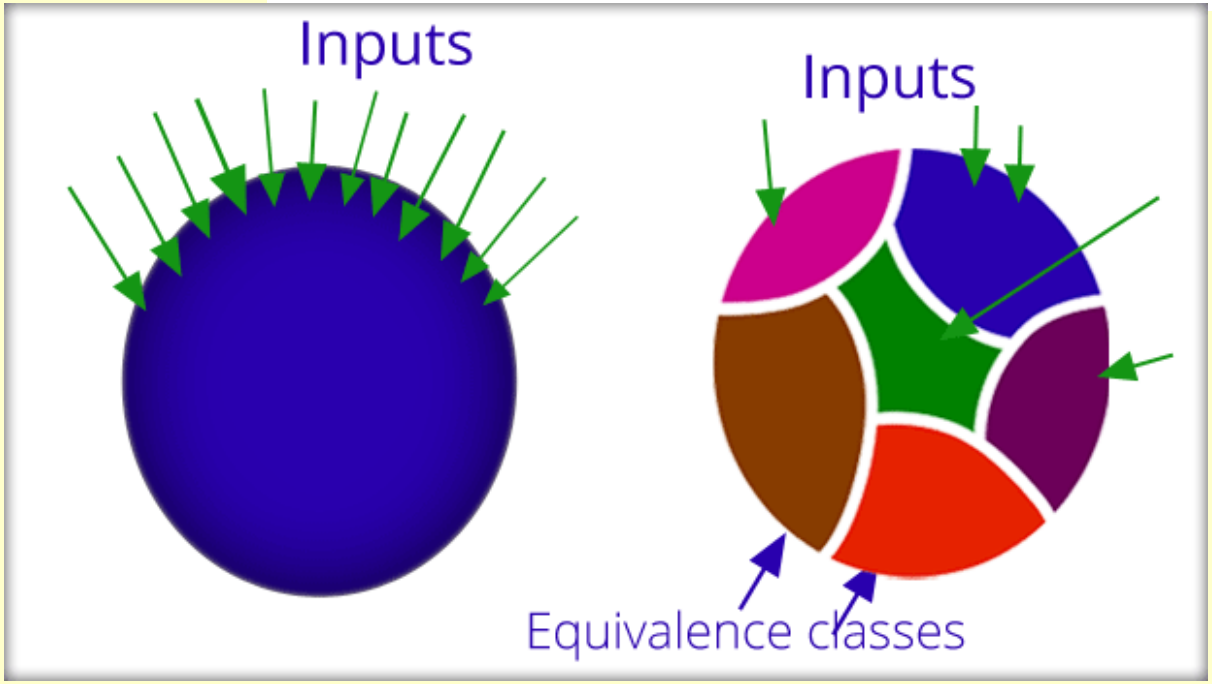
Testing - additional information

Unit



Testing - additional information

Unit **Equivalence classes + boundary values**



http://www.testnbug.com/2015/01/equivalence-class-partitioning-and-boundary-value-analysis-black-box-testing-techniques/

Testing - additional information

Unit

Pair-wise / All pairs testing

Example

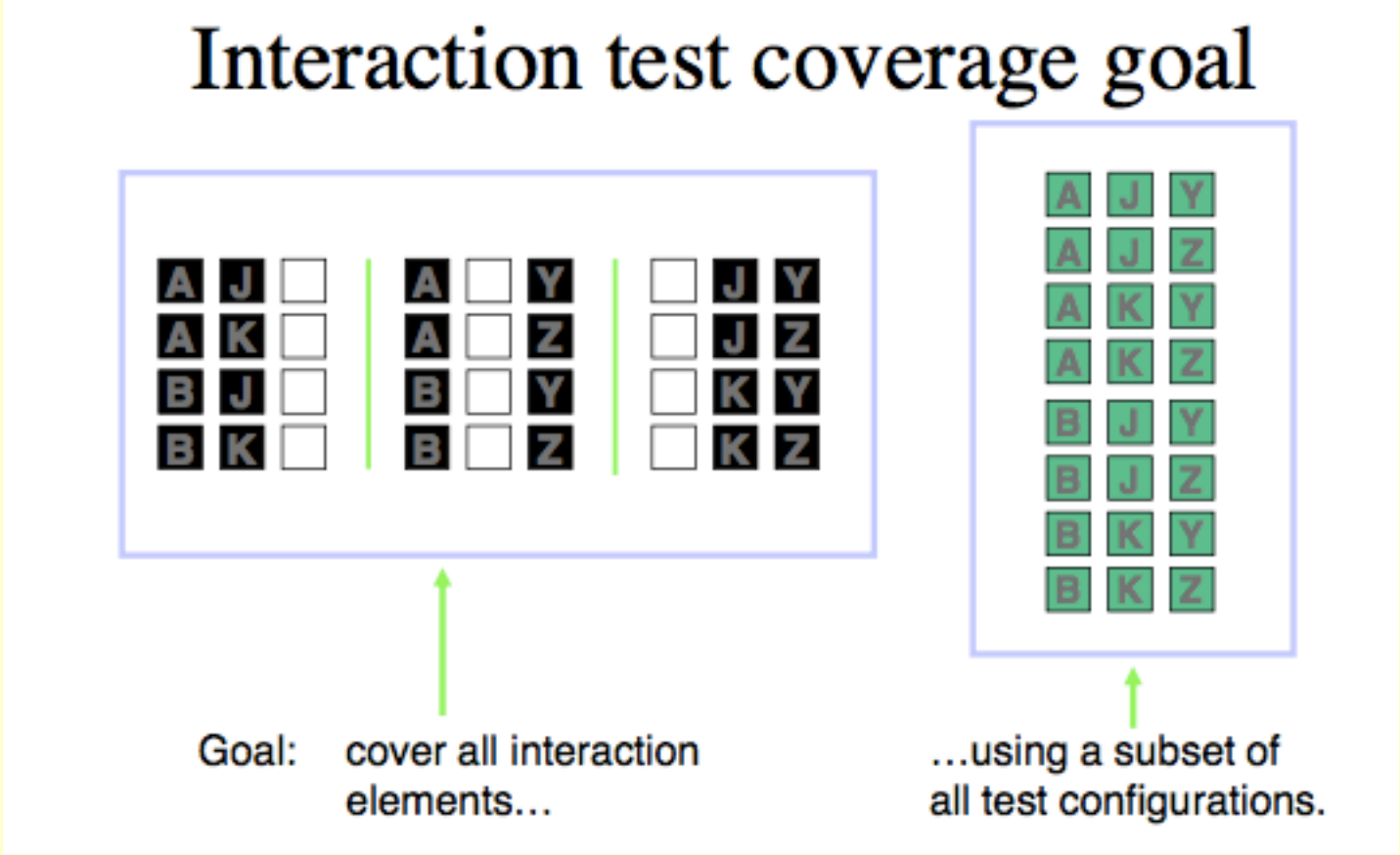
- Suppose that we have three parameters.
- For each parameter, there are two possible values.
  - Values are :
    - A, B for parameter 1.
    - J, K for parameter 2.
    - Y, Z for parameter 3.
- Degree of interaction coverage is 2.
  - We want to cover all potential 2-way interactions among parameter values.

There are 8 possible test configurations

Testing - additional information

Unit

Pair-wise / All pairs testing

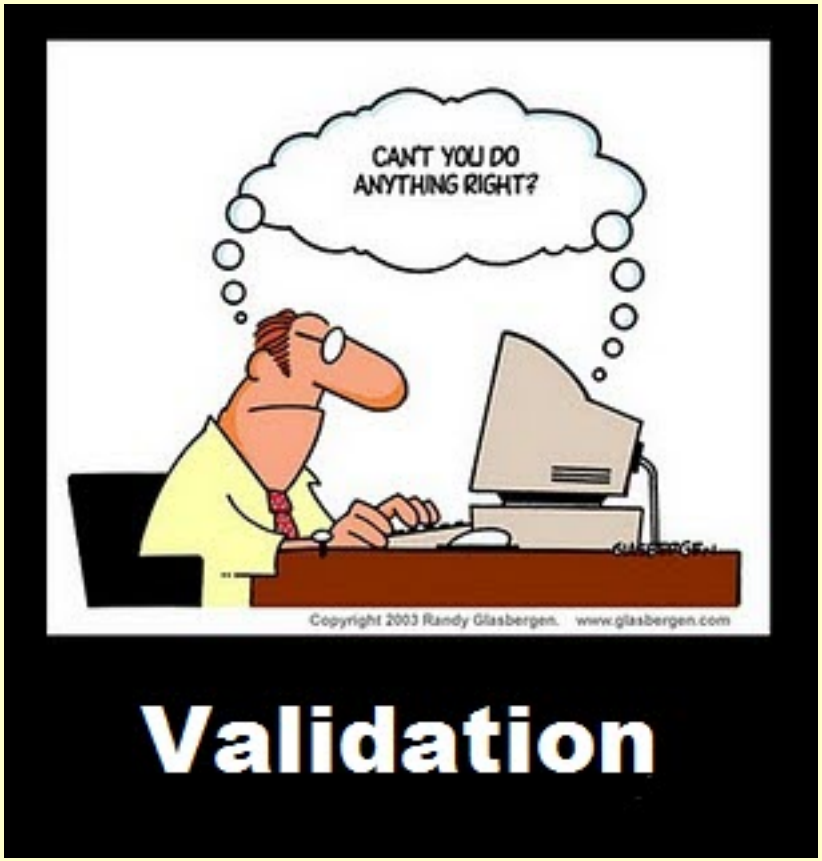


Testing - additional information

Integration

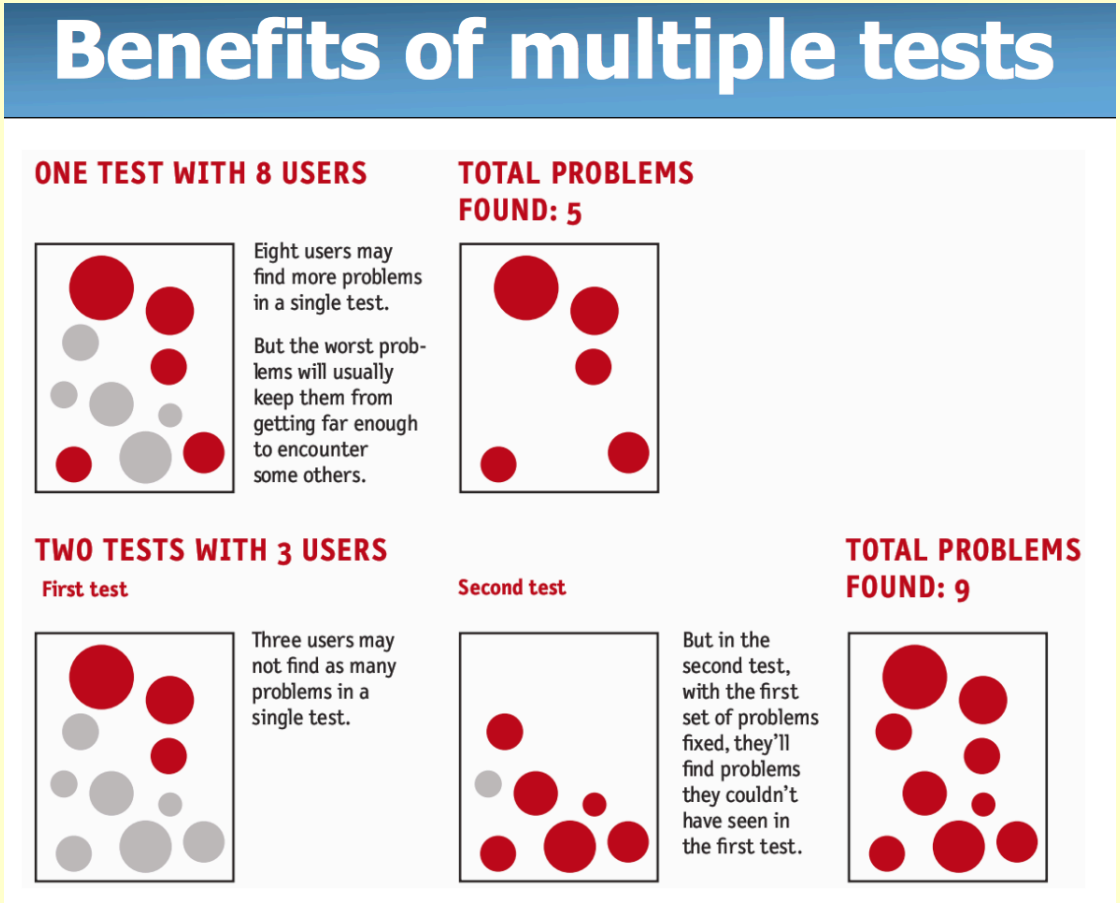
Testing - additional information

Validation/Acceptance



Testing - additional information

Usability



Testing - additional information

Security http://heartbleed.com

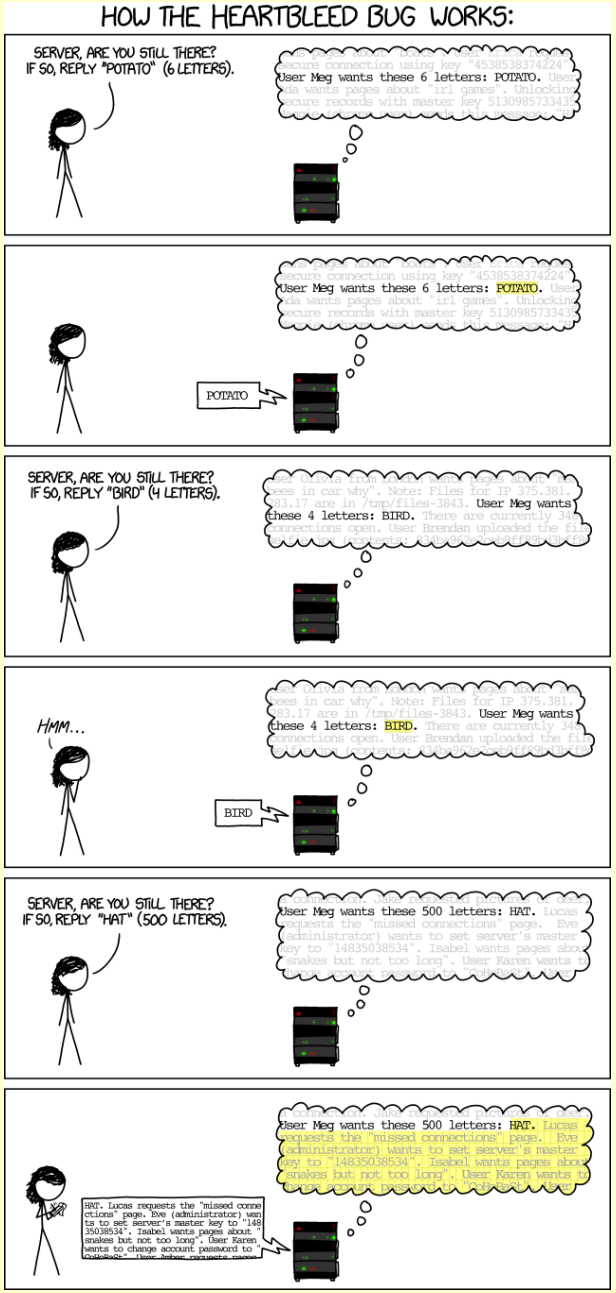
The Heartbleed Bug is/was a serious vulnerability in the popular OpenSSL cryptograph
software library

The buggy code

```
/* Read type and payload length first */
hbtype = *p++;
n2s(p, payload);
pl = p;
```

Testing - additional information

Security



Testing - additional information

Security "goto fail" – Apple's SSL bug

```
. . .
hashOut.data = hashes + SSL_MD5_DIGEST_LEN;
hashOut.length = SSL_SHA1_DIGEST_LEN;
if ((err = SSLFreeBuffer(&hashCtx)) != 0)
    goto fail;
if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;  /* MISTAKE! THIS LINE SHOULD NOT BE HERE */
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
    goto fail;

err = sslRawVerify(...);
. . .
```

if `err` is zero and there is actually no error to report.
The result is that the code leaps over the vital call to `sslRawVerify()`, and exits the function.
This causes an immediate "exit and report success", and the TLS connection succeeds, even though the verification process hasn't actually taken place.
A skilled attacker can easily exploit this

Testing - additional information

Security Intel's AMT Vulnerability

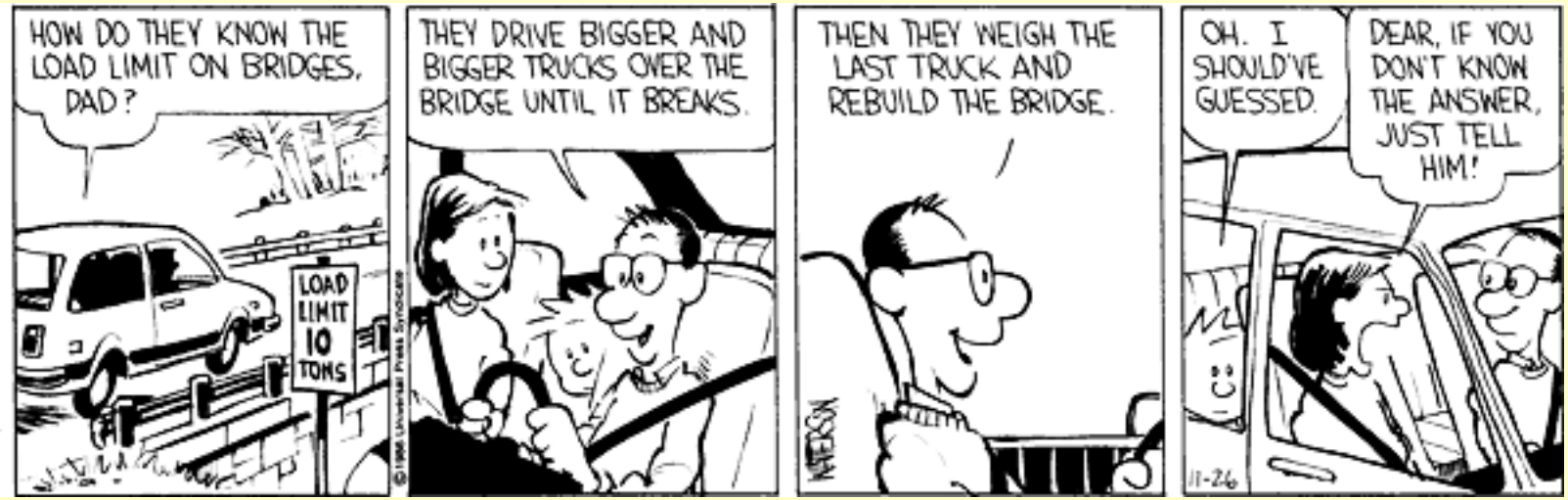
The buggy code

```
int main () {
  string realpass = "secret";
  string userpass = "user-secret";
  int equal = strncmp(realpass.c_str(), userpass.c_str(), userpass.size());
  if (equal == 0) {
     printf ("'%s' equals to '%s'", realpass.c_str(), userpass.c_str());
  }
  return equal * equal; // make sure it's positive
}
```

Question: can you see the problem?

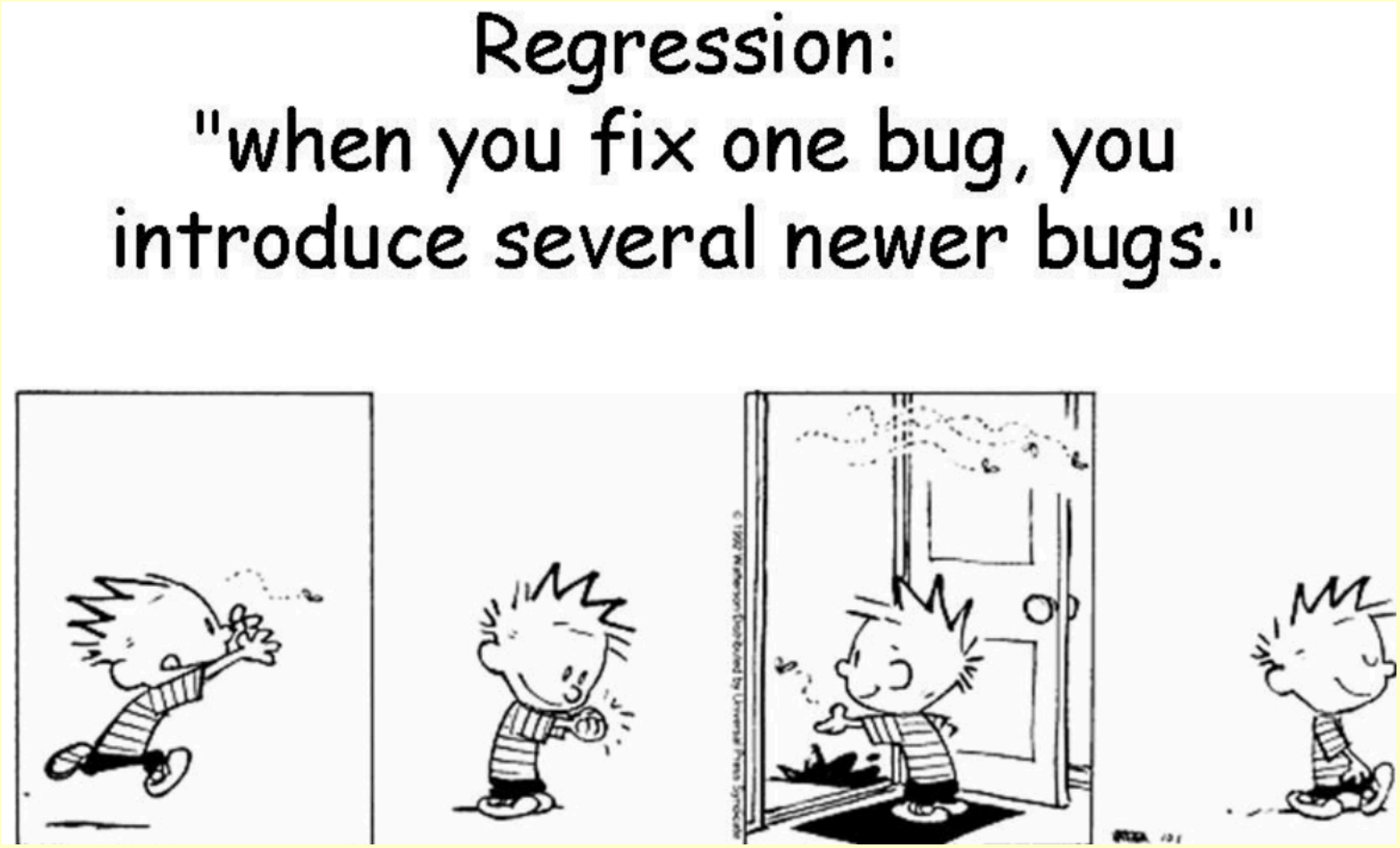
Testing - additional information

Stress Testing



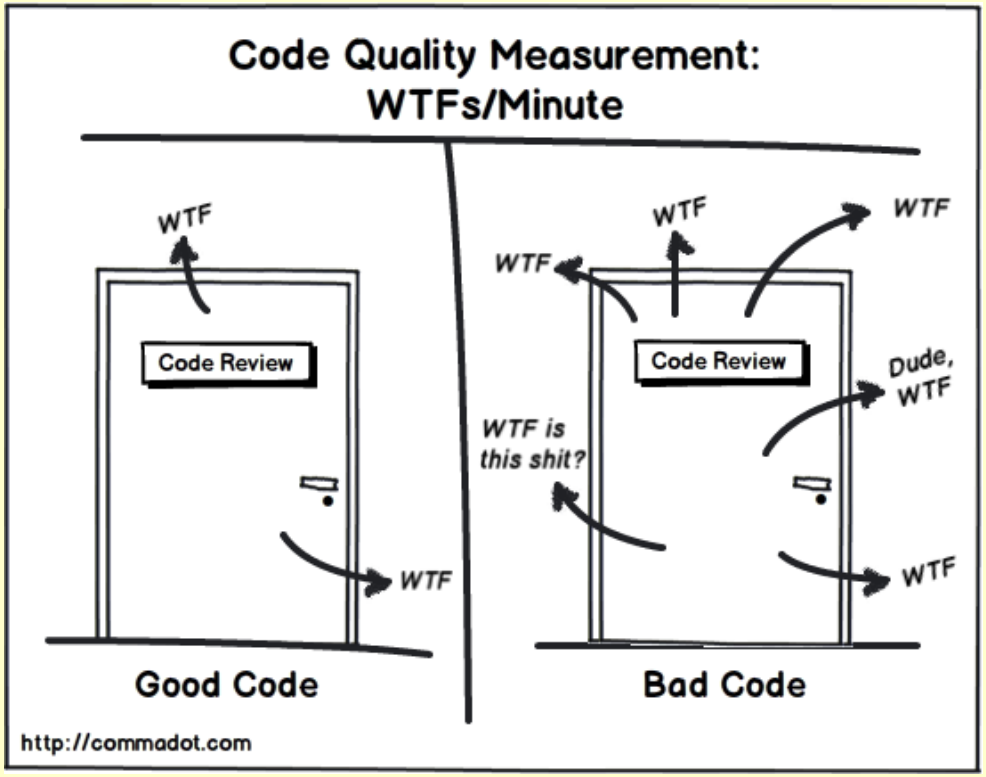
Testing - additional information

Regression Testing

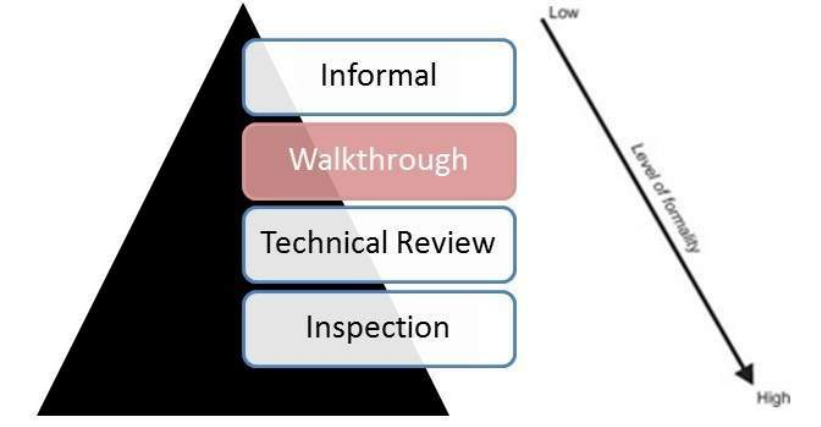


Testing - additional information

Complementary Activities



Code Walkthroughs

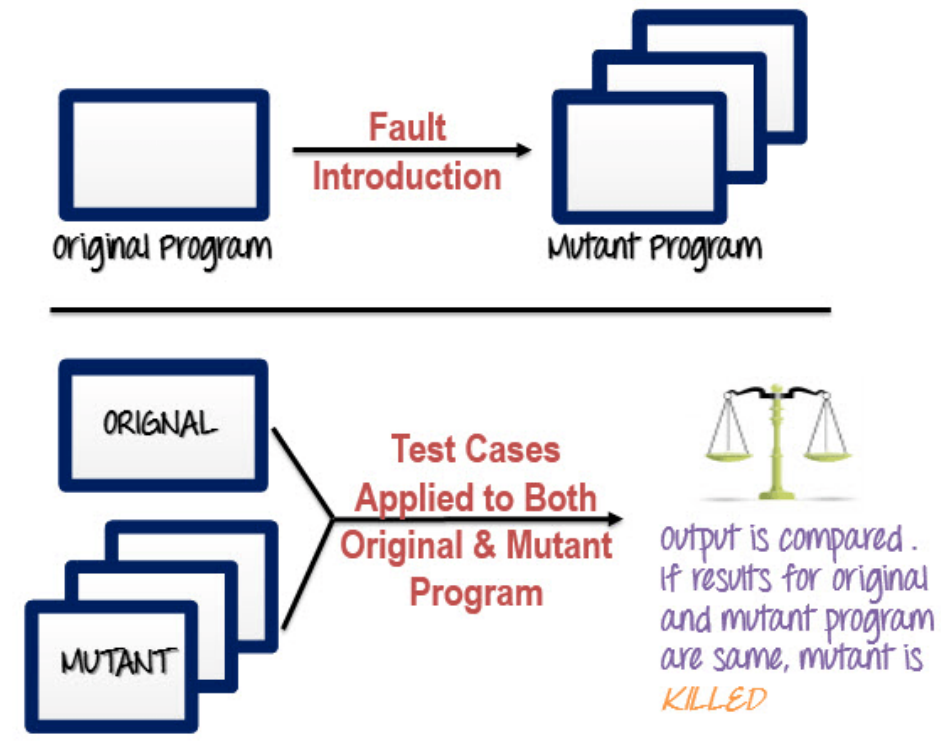


Testing - additional information

Complementary Activities

Fault Injection: software mutation testing

http://www.guru99.com/mutation-testing.html



Testing - additional information

Complementary Activities

Maintenance Issues

Code refactoring and impact on testing?

Testing - additional information

Complementary Activities

Code refactoring and impact on testing?

Most unit tests won't be affected by a single refactoring.

If the refactoring touches multiple classes then we can fall back to the acceptance tests. Now, the acceptance tests guarantee us that the functionality remains the same.

But, if we change the interface of a class, a lot of its unit tests and some acceptance tests will not run anymore.

Problem 2 TDD

Queues from stacks

Specify Design and Implement Tests (of all types) for the following requirement:

You are to implement a FIFO queue of integers with a maximum size MAX, using only 2 LIFO stacks of integers (also with a maximum size MAX)