4156

11/10/15

new assignment posted

combine upcoming lectures

black box testing

purpose of testing (black box or otherwise) is to find bugs

what is a bug?

1. sw doesn't do something that requirements says it should do

2. sw does something that req. says it should <u>not</u> do

3. sw does something that req doesn't mention

4. sw doesn't do something that req doesn't mention but should

5. sw is difficult to understand, hard to use, slow, etc.

consider simple calculator

#1 press + key, nothing happens
or get wrong answer

#2 after some period of time
or number of uses, calculator
stops responding

#3 calculator displays all 0's
when battery weak

#4 does add, subtract, multiply,
divide & undocumented
square root

#5 buttons too small
= key in odd place
display cannot be read
in bright (or dim) lights

which are bugs in calculator
implementation vs. bugs in
requirements?

4156

11/10/15

black box writes tests from requirements, not from the code

- if we don't know what to test or how to test it, that may be a bug in the requirements

good code is testable

how do we know when we have a "good" set of tests (from black box perspective, consider grey/white separately)

- find all the "inputs" to the system, as implied by use cases or user stories

  not all input comes from human user

  can come from environment or external systems

  is there at least one test case for each different input point? (not value)

  opportunity to provide a value

- same for all "outputs"

4156

11/10/15

for each input point, there might be "infinitely" many possible values

- cannot try all of them (exhaustive testing)

- need to try more than one

- how to decide which to try?

"equivalence partitions" (or equivalence classes)

- set of values where we expect SW to behave the same (equivalent but not necessarily equal results)

with a calculator, might expect all positive integers (up to some maximum) to behave same

but zero or negative might behave differently positive, zero, & negative integers are valid for most calculators

so are floating point - to some degree of precision

but alphabetic characters are not valid inputs

set of equivalence partitions for arbitrary sw might include multiple valid partitions & multiple invalid partitions

not all invalid inputs are same!!

consider dates, with month represented as integer 1 to 12

0 invalid 13 invalid
-1 invalid string invalid

but how sw reacts to the invalid input might be different

- different error messages or exceptions
- crashes instead of error message

4156 11/10/15

- hangs or loops forever instead of error message

- "easter egg"
  some sw builds in special responses for certain input, usually invalid inputs as a joke

Consider a calendar program, which does something different based on season

it assumes

12, 1, 2	winter
3, 4, 5	spring
6, 7, 8	summer
9, 10, 11	fall

4 different valid equivalence classes

need to test month on one value from each

what if end user thinks

1,2,3 winter 4,5,6 spring
7,8,9 summer 10,11,12 ~~oo~~ fall

11/10/15

categories of possible equivalence
classes to consider

for ordered values there may
be range issues — below
within
above

for inputs w/ size or length,
then more range issues
too big, toosmall, within limits

sets have members & non-members

container data structures have
set of contents and
max/min number of contents

in or not in container
container empty or full
is same contents allowed more
than once?
must contents be in some order?

→ various "data structures"
issues

4156

11/10/15

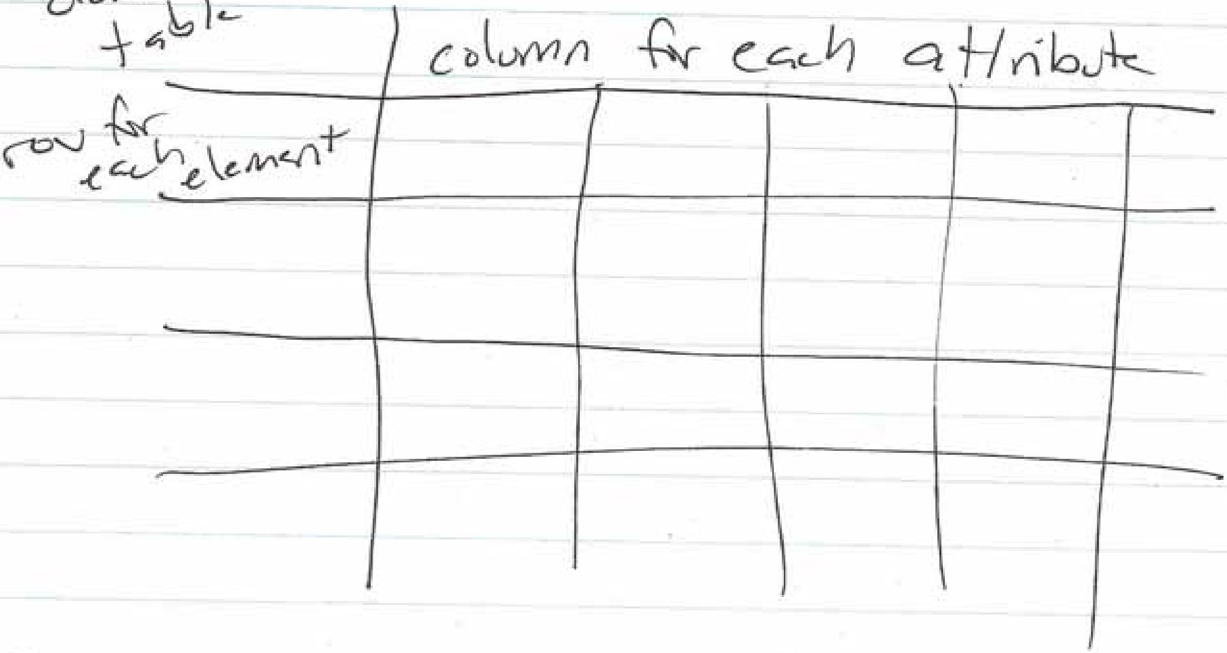
number - negative, zero, positive
number of decimal points
or range of exponent

max
min
integer
overflow

String - printing vs. non-printing
Special character sets
length (buffer overflow)

table - with & without "missing" or
"repeated" values
sorted or not sorted by "key"

database table



file - exists or not
readable, writable, neither (permissions)
correct format or not

4156

11/10/15

equivalence partitions need to be tested for EVERY input point

also need to consider any relationships among inputs

combination of values
consistent or not
- date of birth & age

• in addition of at least one value from every partition, need to ~~con~~ consider BOUNDARY cases

- aka corner, edge cases
off by one, fenceposts

what constitutes a boundary & where to look for them

min min-1 min+1
max max+1 max-1

consider how data represented in computer
$2^n - 1$, 2^n, $2^n + 1$, 0
default, null, empty, blank, none, etc.

4156

11/10/15

Wow, that's A LOT of test cases

yes it is

and we haven't even gotten to grey / white yet!

good code often has 3 x as many LOC test cases as application code — really!!

good code also has independent test cases

can run in any order w/ same result

if multiple test cases need the same state setup, put in same test CLASS with common setup & teardown

blackbox can be either unit or system level

- or integration, 2 or more units but not full system
- when not full system need driver w/ setup / teardown
- ideally also automated for system