

4156

11/12/15

more black box testing

emphasis on black box

- test w/o considering how it works
- just enter inputs, receive outputs & check results
- aka behavioral testing

test case consists of inputs
AND expected outputs

↑
need a "test oracle"

Some way of checking that results are correct

normally the tester has some kind of requirements to check, and derives the test oracle from those req.

but sometimes there is no req. &/or no test oracle, so the tester performs "exploratory testing" to figure out what the program is supposed to do

4156

11/12/15

exploratory treat the software itself
as the spec -

methodically explore feature
by feature

use the features found to drive
further testing

test-to-pass - only assure that
SW minimally works
(e.g. for a demo)

test-to-fail - only after test-to-pass
now you want to find bugs
probe for weaknesses

- try to force error messages
or lack of error messages

how to select (or define)
test cases?

- equivalence partitions

I talked about this Tuesday, &
it sounded like I was adding
tons of testing - but really reducing
compared to - exhaustive testing

4156

11/12/15

systematically
try to find minimal effective
set of tests

the test cases in the same equivalence
class should reveal the same bug
- group similar inputs,
similar outputs,
similar operation

example: windows filenames

cannot contain certain
special characters

\ / : *
? " < > |

1 - 255 characters

so - valid chars, invalid chars
valid length, invalid length
↑ too short
and too long

be careful that selecting only one
(plus boundaries) of an equiv. class
does not reduce fault finding ability

4156

11/12/15

boundary conditions -
 Sub boundaries
 nulls
 bad data

test both
 sides of
 boundary
 at boundary

there is lot of data in most programs
 words in word processor
 numbers in spreadsheet
 # of shots left in space game
 pictures in photo SW
 backup files stored offline
 data sent over network

powers of 2 sub boundaries

let's say SW accepts 1-1000

bit	0 1	
nibble	0-15	14, 15, 16
byte	0-255	254, 255, 256
word	0-65,535	

particularly relevant if program
 does any sort of compression
 for storage or to
 send over network

11/12/15

ASCII

bms up
chapter on
screen for
table p.77

null	0	B	66
space	32	Y	89
/	47	Z	90
0	48	[91
1	49	'	96
2	50	a	97
9	57	b	98
:	58	y	121
@	64	z	122
A	65	£	123

so testing for alphabet characters
means A-Z a-z
@ [' £

ASCII - each char is 8 bits
Unicode - each char is 16 bits

software with default values
for when user enters nothing
(or deletes defaults if
they are displayed)

4156

11/12/15

or user could "fill up" a text entry field -

shoe test - put shoe on
a ^{auto-}repeating keyboard
+ go to lunch

blank, empty, null, zero, none, default
should get own partitions

also invalid, wrong, incorrect,
+ garbage data

if user enters garbage data +
something bad happens, the
user will blame the SW not
themselves - the SW has a bug

state testing - what are the
states in your program,
map them out
- might be different pages
in web app

what are the transitions, how
to get from one state to another

page 7

4156

11/12/15

let's try to break something