Thursday 10/30/14 4156

return to testing black box
grey box
white box / glass box

black box focuses on input & output
functionality from user PoV.

user can be end-user - acceptance tests
or other code - unit tests

set of test cases for each user story
or use case

set of test cases for each public method

input validation - define what is
"good" input & reject everything else
in a way that user can understand
- whether human or code user

why not define "bad" input &
only reject that?

are outputs correct for each test input?
how do you know? "test oracle"
what if you don't know?
- addressed next Thursday

Thursday 10/30/14 4156

state transitions from external viewpoint
map states of full system not
individual objects

boundary cases & off by one errors -
programmers typically make small mistakes

grey box testing peeks beneath covers
independent testers PoV

verify any auditing & logging

check data sent to other systems
format & contents

check system added information
checksums, hashes, timestamps

scraps
resource leaks
make sure data really allocated
& deleted (cleared)
make sure uninstalls cleanly
files, registry entries

Thursday 10/30/14 4156

white box testing

usually done by developes because relies on access to source code

goal is to try to break-force errors

coverage / mutation ??

test all ~~[illegible]~~ functions

test all statements (aka line coverage)

test all branches - have all decisions been covered, need to force each path

test all conditions - has every boolean subexpression evaluated to both true & false

if code corresponds to state machine, have all states been exercised

parameter value coverage - have equivalence partitions & bondaries been exercised for every parameter

Thursday 10/30/14 4/56

more white box testing

check proper error handling
invalid data, cleanup up

check for thread-safe, null
arguments & return values,
security roles

check any resource constraints
memory, disk, network &
database connections

need to fail gracefully

test coverage tools tell which have/have not
been exercised by set of tests
functions
branches
statements (or lines)

hard to reach 100%, often 85-90%

model checking - return every possible
error code from APIs, libraries,
system calls

Thursday 10/30/14 4156

stress testing

"quick test"
- put cursor on text input field
- put shoe (or other heavy object on keyboard)
- go to lunch
- relies on keyboard auto-repeat to overflow input buffer

interference tests

- force screen to refresh
- change video resolution
- toggle "accessibility" options
- change system date / time
  - to far past, far future
  - to near past, near future
- change localization settings
- click mouse randomly all over screen
- set this program or another program's timer to go off
- change focus to another application, do something, return
- load enough other applications to force out of memory to disk

Thursday 10/30/14 4/56

more interference tests

- lock necessary database records to another program
- cancel processes
- pause/kill client or server (other one)
- pause program for long period then resume
- leave running a long time

file system interference tests

- remove CD-ROM, flash drive or other media while in use
- fill file system to capacity
- assign invalid file name
- vary file names & access permissions
- change or corrupt contents of files while being read/written

scalability tests

- connect large number of clients
- connect, disconnect, reconnect repeatedly
- bombard with requests
- benchmarks & tools available