"static" testing

black box- review requirements
                          & design
              → documents

white box - review code

why?   find problems earlier
            find problems that
                dynamic testing can't
          (e.g., unreadable code)

four essential elements

   - identify problems in document,
      not author

          find missing items
          as well as "wrong" items

   - follow rules

          define roles   external
              moderation - QA?
              recorder

define length of time or
amount of document
(pages, loc)

- prepare

  do mini-review in advance
  alone or in pair

- write report

  Summarize results
    how many problems found
    where found
    what kind

  in addition to bug reports
  or change requests

  why? what can report be used for?

walkthrough - read through informal
           line by line   review

explain what code does & why

4156

inspection - reader is not   formal
author   review

more roles besides moderator,
recorder

real or
role-
played

represent end-user,
tester,
technical support

each problem classified   find problems, ⭐
problems
don't
fix
them

mmor - author fixes on own
moderate - moderator reviews
severe - whole team meets again

two hours standard length

pair programming sometimes
replaces code review

coding standards are more
common now so less need
to check for styles
- state tools can do

QAPlug
↑

checkstyle
findbugs

"dynamic" testing

    black box - equivalence partitions
                & boundaries

    white box - look at code
        thoroughness of testing

directly test low-level functions
    classes & methods
    APIs / libraries

also test at top-level, but considering
    how sw operates

access state information to
    determine whether tests are
    doing what you think they are

force software to do things that
    would be hard to force from top level

measure how much code is hit,
    what specifically was missed
        remove redundancy ← don't lose
           add missing   fault
                          finding
                          ability!!

dynamic white-box testing is not
   debugging = fix bugs

   altho like all testing goal
   is ultimately to find bugs

   both debug + dynamic
     white box "isolate" bugs
    - where & why bug occurs

good sw tester should help w/
  debugging by narrowing down
  problem to simplest test case
  that demonstrates the bug

   - if found during white box
   testing, we know suspicious
   lines of code (the lines
    exercised by that test
    & not exercised by other
    test that passed)

~~dependencies~~ dependencies among tests
   - tests can cause
     false positives (fails) &
      false negatives (passes)

bugs hide each other

  when run in "wrong" order

integration testing

test units 1st

$\boxed{\text{incremental}}$

put together full system adding
one unit (class) at a time

bottom up — need drivers
top down — need stubs

both need "mocks" for external
systems, databases, etc.

coverage — dataflow +
control flow

included in
Thursday
lecture