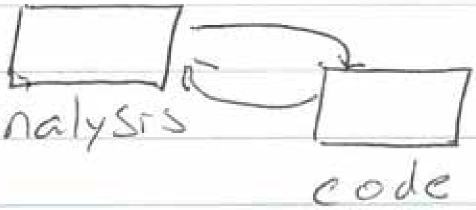
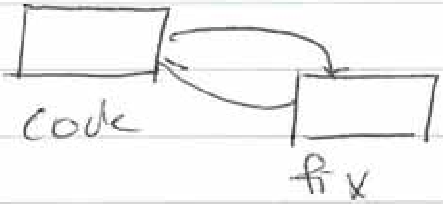
Tuesday 10/7/14 4156

design asgn - risk list start review on Thu?

software process

every SW process includes at least 2steps





a simple 2-step (iterating) process may be sufficient for personal use where the developer = user (& only 1 developer) but not practical for most systems - more steps needed for delivery to custumer (even internal customer)

→ waterfall model, presented by Win Royce - described current practice, not invented (see paper show figures)

presented as if always forward progress but in reality backlinks - and not just to prior step (with more manageable change process)

fundamental problem is testing comes too late -

Thursday 10/7/14 d d d + hcd 1425 4156

when late testing finds insurmountable problems, may need to return to origin - 100% cost & schedule overruns

various processes introduced since then to combat this problem (TDD is only one ~~?~~ or part of one)

solution in paper is to design for "risky" steps early & then treat as constraint (part of requirements)

and also produce huge amount of documentation at every step

also "do it twice" - not just a prototype but a small scale model of system

special emphasis on testing - separate specialist testers independent from developers, visual code inspection, coverage of every logic path

involve customer throughout, commits at various earlier points prior to delivery

result is complicated version of waterfall

Tuesday 10/7/14 4156

in 1970 had gist of many modern ideas on software engineering
- except documentation parts
customer usually wants working software, not docs (other than military)

unfortunately, many readers took away only the strawman version of waterfall - as if that's what they should do, r.t. what they should *not* do

modern processes are mostly variants of
- incremental delivery } possibly
- iterative planning } combined
w or w/o TDD

pure incremental delivery
- do entire requirements + design
- then code, test, delivery increment

more realistic
- partial requirements & design
- corresponding partial code, test, deliver increment

emphasis on *delivering* increments

Tuesday 10/7/14 4156

pure iterative planning -
entire lifecycle on each iteration,
w/ new version of sw
minimal code ~~[illegible]~~ reuse
maximal experience reuse

"extreme programming" is an extreme
combination of incremental
& iterative
- agile plus dogma
- TDD arose from XP

show XP rules from courseworks

goal of XP, & all sw processes,
is to reduce risk

development team
someone may leave
or fall behind
customer
changes mind
goes out of business
need to entice new customer

Tuesday 10/7/14 4156

technological

is it even possible to
implement this
w/in time and budget?

class exercise

get together w/your team members

what are biggest risks to
your own projects?

volunteers present