4156 add Nitesh to calendar HW#1 & EC 9/15/15

recap: already covered Head First ch. 1-2, now on ch. 3

project planning

gather requirements by brainstorming, etc.

user stories on index cards
title
≤3 sentences description
time estimate in days

how long will it take to deliver?
add up all the time estimates
→ almost certainly too long!

one approach is for development team to pick set of stories for 1st release, e.g. 90 days

why is this a bad idea?

your team has effectively set these stories as higher priority than those left out
→ but customer needs to set priorities!

4156

9/15/15

help customer prioritize

shuffle deck of user story index cards & lay out on table

ask customer to select the stories for milestone 1.0 – 1st delivery

initially ignore time estimates so you can see what customer thinks is most important even if won't all fit

but then need to sanity-check for time
→ probably still too long so need to re-prioritize w/ customer

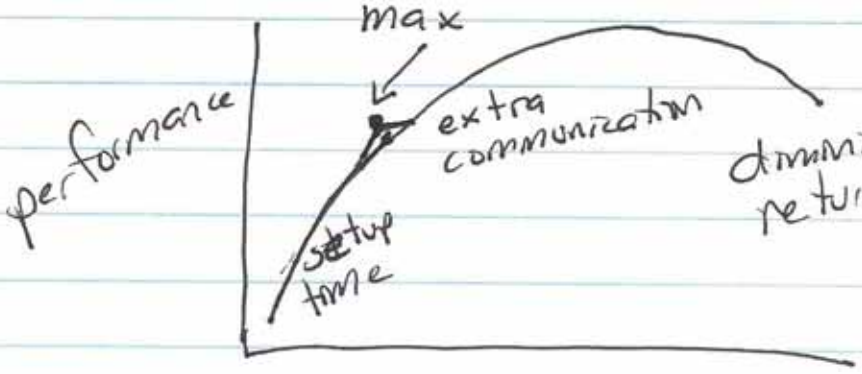
- cut functionality
- ship milestone build asap
- focus on baseline functionality
  smallest set of features to actually be useful

e.g. word processor
load, edit save are core

4156 9/15/15

what if cannot possibly fit baseline user stories into customer's deadline for milestone 1.0

- give up, walk away
- add more people



doubling people does NOT halve time

what is the desired maximum team size?

not a specific number, can sometimes improve productivity by dividing project into several subprojects = so each team is smaller

why 4 for 4156? smallest possible pair programming } for cross-pair integration }

4156

9/15/15

Within milestone 1.0, need to prioritize for iterations

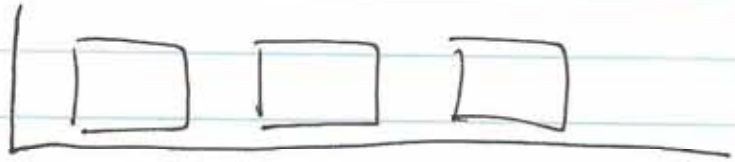
recall iteration no more than
20 work days
(30 calendar days)

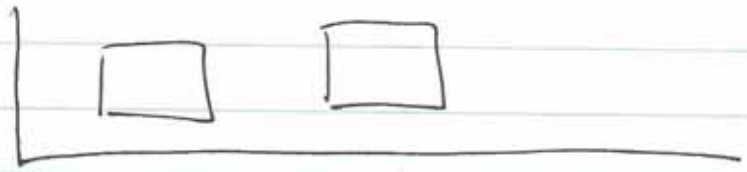
rank user stories according to importance (to customer)

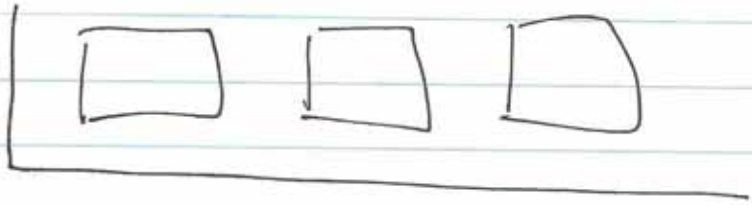
10 most
20
30
40
50 least ← still needs to be part of 1.0 can't leave out

(make all other user stories 60 for now)

ideas to group n.t. start ranking

1 

2 

3 

lay out stories into iterations

20 working days
x 4 developers
= 80??

4156 9/15/15

get feedback from customer at end of each iteration

continuous integration - software always builds so in principle could show customer something at any time

build = compile
package
test

(more on CI later in course)

principles: keep iterations short
keep iterations balanced

easier to deal w/ unexpected change as it arises

balance dealing w/ change, new features, debugging & testing,

so, we get 20 days worth of work time, per developer in each 30 calendar day month, RIGHT?

4156 9/15/15

WRONG!

"overhead" impacts how fast team really develops SW

developer estimates often do not account for overhead

velocity = % percent of time that is productive work

$$\frac{\text{days of work}}{\text{velocity}} = \text{days required to get work done}$$

for new projects start with 0.7 velocity

adjust at end of each iteration

$$\text{velocity} = \frac{\text{actual work}}{\text{scheduled work}}$$

$$= \frac{\text{total of estimates for user stories completed}}{\text{total for stories scheduled}}$$

1 calendar month = 20 working days = 14 days real work

415-6

9/15/15

need to consider velocity when planning w/ customer
— don't surprise later

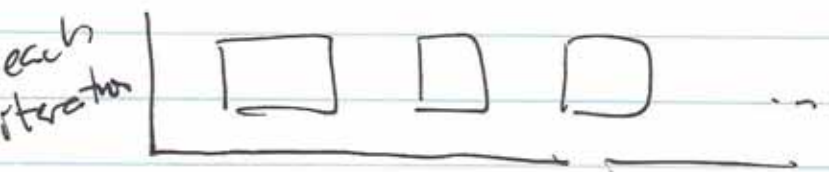
let's say 3 iterations for 4-developer team, so max 80 days per month
x 3 months until milestone 1.0
= 240 days ~~of real work~~

estimates were
Iteration 1 - 73 days
2 - 67
3 - 77
---
217 days

looks good until consider velocity

$\frac{217}{.7} = 310$!! Uh oh

team can really do only
4 x 20 x 0.7 = 56 days per month
= 168 days per quarter
NOT 240



max 56
not 80

4156

9/15/15

What if you've considered velocity all along but are still running behind schedule

tell customer
add an iteration
postpone overflow work

Where does 30% lost time go?

(ust work ~~time~~, not considering ~~personal~~ personal or sick days)

software installation, upgrades, patches ←

customers & other professional visitors

training staff etc.

and maybe not really 70%, that's just a velocity to start 1st iteration, later iterations measure previous velocity & adjust

velocity helps you promise & deliver rather than overpromise & fail

4156

9/15/15

"big board" development dashboard