csc444h:
software engineering I

matt medland
matt@cs.utoronto.ca
http://www.cs.utoronto.ca/~matt/csc444

announcements

- one a2 left to be handed back
- lab tomorrow for demo/questions
- presentations next week, scrum standup meetings in lab next week

APPLIED RESEARCH
IN ACTION 2015

TUESDAY, DECEMBER 1
4:00 - 7:00 PM

Join us for a showcase of R&D internship projects by the 2014 cohort of the Master of Science in Applied Computing (MScAC) program.

DEPARTMENT OF COMPUTER SCIENCE
INNOVATION LAB (DCSIL)
GERSTEIN SCIENCE INFORMATION CENTRE
9 KING'S COLLEGE CIRCLE, 2ND FLOOR
TORONTO, ON

RSVP uoft.me/aria2015

Computer Science
UNIVERSITY OF TORONTO

effort estimation

estimates

- estimates are never 100% certain
- ex. we may estimate a feature to be 20 ECDs (ideal 8-hour developer days)
  - are we saying it will be done in 20 ECDs? no.
  - so, then what exactly are we saying?
    - is it optimistic?
    - pessimistic?
    - how confident are we in it?
- a quantity whose value depends upon unknowns (or randomness) is called a *stochastic variable* - our plan is full of these!

estimation techniques

Source: Adapted from van Vliet, 1999, section 7.3.5

- function points

  $FP = a_1I + a_2O + a_3E + a_4L + a_5F$

  the a_is are "weighting factors"

I	=	number of user inputs (data entry)
O	=	number of user outputs (reports, screens, error msgs)
E	=	number of user queries
L	=	number of files
F	=	number of external interface (other devices, systems)

- an example might be:

  $FP = 4I + 5O + 4E + 10L + 7F$

estimation techniques (2)

- three-point estimating
  - tends to provide better estimate than asking for a range

w	=	worst-case estimate
m	=	most likely estimate
b	=	best-case estimate

$$E = \sum_i \frac{w_i + 4m_i + b_i}{6}$$

confidence intervals

- toss a coin 5000 times
  - expect heads about half the time (2500)
  - exactly 2500? only about 1.1%
  - ≤ 2500? chance is 50%, on repeated experiments, half will be ≤ 2500, half will be > 2500
  - ≤ 2530? chance is now about 80%
  - ≤ 2550? chance is now about 92%
- these (50%, 80%, 92%) are called confidence intervals
  - with 80% confidence we can say that the number of heads will be less than 2530

confidence intervals

- toss a coin 5000 times
  - – expect heads about half the time (2500)
  - – exactly 2500? only about 1.1%
  - – ≤ 2500? eriments, half will
  - – ≤ 2530?
  - – ≤ 2550?

how do you estimate when a feature will be done with 80% confidence?

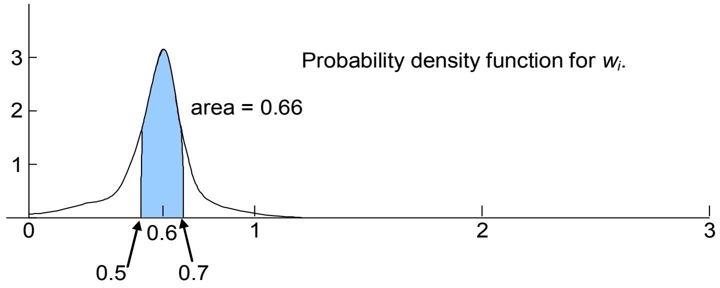
- these (50%, 80%, 92%) are called confidence intervals
  - – with 80% confidence we can say that the number of heads will be less than 2530

stochastic variables

- consider a developer with a work factor, *w*
  - – *w* (even measured) is a stochastic variable
  - – stochastic variables are described by statistical distributions
  - – a statistical distribution will tell you:
    - for any range of *w,* the probability of *w* being within that range
  - – can be described completely with a probability density function (PDF)
    - x-axis: possible range of the variable
    - y-axis: numbers (density) ≥ 0
    - probability the value is between two values, a and b, is the area under the PDF between a and b

PDF for work factor

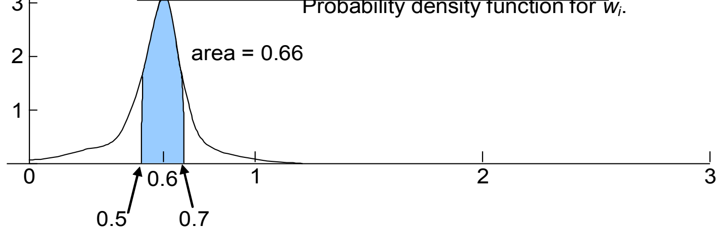
- probability that 0.5 < *w* < 0.7 = 66%
- looks to be fairly accurate in practice
  - – finite probability of being 0
  - – not much chance of being bigger than 1.2 or so



PDF for work factor

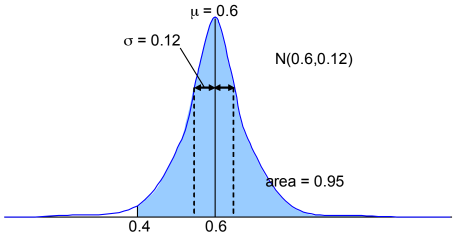
- probability that 0.5 < *w* < 0.7 = 66%
- looks t
  - – finite
  - – not r or so

- use w = 0.6 if you don't have any previous data
- why is there a chance of w = 0?
- why does the x-axis end at 3?



normal for work factor

- assume work factor is described by a normal distribution
- 2-points needed to fit a normal, average case, and some reasonable “worst case”
  - avg. case, half the time less, half more = 0.6
  - “worst” case: 95% of the time *w* won’t be that bad = 0.4
    - normal that fits is N(0.6, 0.12)



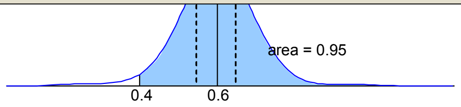
normal for work factor

- assume work factor is described by a normal distribution
-

normal distribution is easy to work with because it’s symmetric about the mean

using normal is ok because (human) estimation errors are likely to dominate over the choice of PDF

normal extends to ∞ in both directions, so we are working with a cutoff version…but not cutting much



back to estimation

- ex. for a feature estimate of 1 week
  - post-facto
    - what are the units?
    - 40 hrs? longer? shorter? dedicated? disrupted (calendar)? one developer? two?
  - stochastic
    - 1 week best case?
    - 1 week worst case?
    - 1 week average case?
    - need a PDF
- depending on these concerns, my “1 week” may be someone else’s 4 weeks!

stochastic capacity constraint

- ***T*** is fixed
- ***F*** and ***N*** are both stochastic variables
- can only speak about the chance of all the features fitting in the release or sprint
- say ***F*** = 400, ***N*** = 10, and ***T*** = 40, are we good to go?
  - can’t say for sure
  - need precise distributions for ***F*** and ***N*** to answer, and then, only with some confidence interval

summing distributions

- ***F*** and ***N*** are sums over many contributing stochastic variables.
  - ex. $\boldsymbol{F = f_1 + f_2}$
  - if $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$ have associated statistical distributions, what is the distribution of ***F***?
    - in general case, no answer
  - however, if $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$ are both normal, then
    - ***F*** is also normal
    - mean of ***F*** is sum of means of $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$
    - standard deviation of ***F*** is the square root of the sum of squares of the standard deviations of $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$

law of large numbers

- if we sum lots and lots of stochastic variables, the sum will approach a normal distribution
- therefore, something like ***F*** is going to be pretty close to normal (for large releases, or longer horizons)
  - ex. dozens of feature estimates summed up
- ***N*** will also be close to normal, but probably less so
  - ex. 5 developer's work factors summed up

delta statistic

- ***D(T) = N × T – F*** (delta)
- we have normal approximations for ***N*** and ***F*** and can compute the normal curve for ***D*** as a function of various values for ***T***
- we are interested in ***P(D(T) ≥ 0)***
  - the probability all features will be finished on time
  - negative delta means it's late!
- in choosing ***T*** (assuming we can) we want a confidence interval the company can live with
- ex. if the company can live with an 80% confidence interval, choose ***T*** such that ***D(T) ≥ 0*** 80% of the time

example: picking T

		confidence level						
		25%	40%	50%	60%	80%	90%	95%
	30	**-39**	**-77**	**-100**	**-123**	**-177**	**-217**	**-250**
	35	**14**	**-26**	**-50**	**-74**	**-130**	**-172**	**-207**
	40	**67**	**25**	**0**	**-25**	**-84**	**-128**	**-164**
T	45	**121**	**77**	**50**	**23**	**-38**	**-85**	**-123**
	50	**174**	**128**	**100**	**72**	**7**	**-41**	**-82**
	55	**228**	**179**	**150**	**121**	**52**	**1**	**-41**
	60	**282**	**231**	**200**	**169**	**97**	**44**	**0**

- ***F*** is normal with mean 400 and 90% worst case 500
- ***N*** is normal with mean 10 and 90% worst case 8
- cells are ***D(T) = N × T - F*** at the indicated confidence level
- important is transition through 0

example: picking T (2)

		confidence level						
		25%	40%	50%	60%	80%	90%	95%
	30	-39	-77	-100	-123	-177	-217	-250
	35	14	-26	-50	-74	-130	-172	-207
	40	67	25	0	-25	-84	-128	-164
T	45	121	77	50	23	-38	-85	-123
	50	174	128	100	72	7	-41	-82
	55	228	179	150	121	52	1	-41
	60	282	231	200	169	97	44	0

- 95% chance of hitting dates, choose ***T*** = 60, or…
- ***T*** = 40 ⇒ only a 5% chance of being > 20 days late
- to be 80% sure, select ***T*** = 49
- gamble with only a 25% chance, pick ***T*** = 33

shortcut

- ask for 80% worst case estimates for features
- if ***F = N × T*** using the 80% worst case values, then there is an 80% chance of finishing on time
- deterministic release plan can be based on this approach

- note: if you also ask for average cases you can fit a normal curve for ***D(T)*** and predict ***P(D(T)) < 0*** (i.e. missing the date)

the end