University of Southern California
Center for Software Engineering

West Virginian University
Modelling Intelligence Lab
http://unbox.org/wisp/tags/STAR

Accurate Estimates without Calibration?

Tim Menzies[1] Oussama Elrawas[1] Barry Boehm[2]
Raymond Madachy[2] Jairus Hihn[3] Daniel Baker[1] Karen Lum[3]

[1]WVU [2]USC [3]JPL

May 10, 2008
(for more info: tim@menzies.us)

ICSP2008
International Conference on Software Process

Process models: *ganz falsch*?

- Wolfgang Pauli: scathing critic of poor theories
  - Labeling then *ganz falsch*, utterly false.
- And *"ganz falsch"* was not as bad as it gets:
  - He hated unclear theories, poorly presented, untestable, unassessable.
  - Famously, he wrote:
    "That's not right. It's not even wrong."
- Two questions for process models:
  1. Are our estimates "correct"?
  2. What are those estimates?
- Our models have variance: $\alpha \leq f(x) \leq \beta$
- If $(\alpha - \beta)$ is large
  1. Can't tell if they are "correct" since …
  2. … we don't know our estimates

Variability inside COCOMO models

$Em_i = m_i x_i + b_i$

$Em_i = 1$ when x_i when $=3$

$$\boxed{\forall x \in \{1..6\}\ EM_i = m_a(x-3)+1}$$

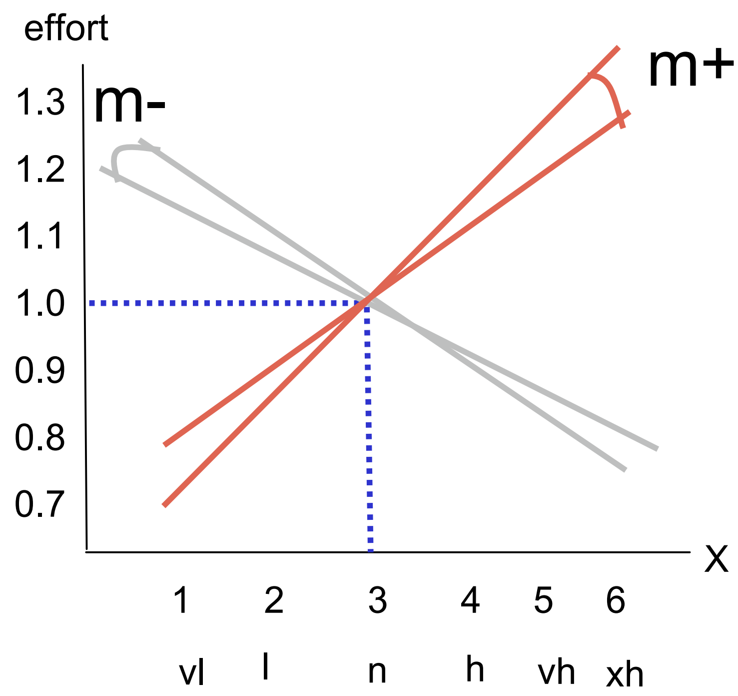
$$(0.073 \le m_a^+ \le 0.21) \wedge (-0.178 \le m_a^- \le -0.078)$$

increase effort

decrease effort

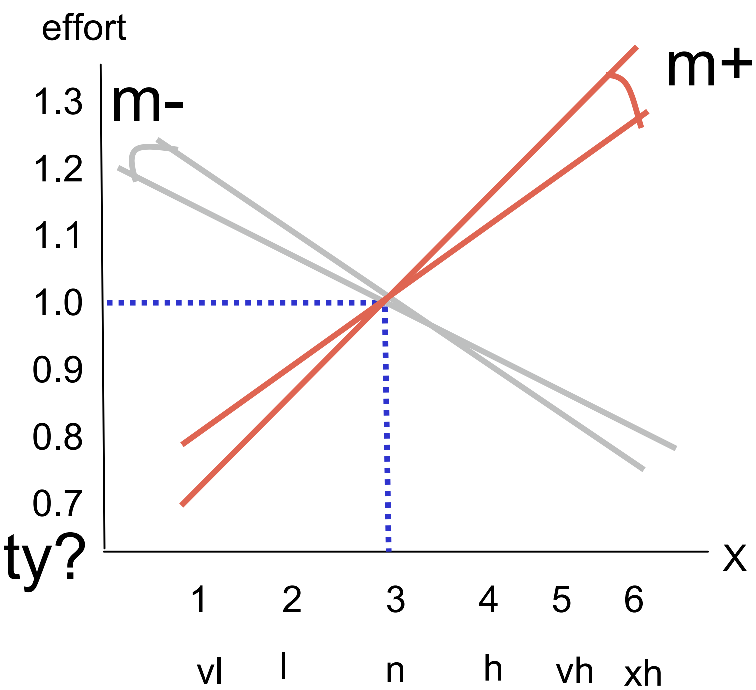
cplx, data, docu
pvol, rely, ruse,
stor, time

acap, apex, ltex, pcap,
pcon, plex,sced,
site,tool



This talk

- Standard approach
  - Use local data to reduce the uncertainty in these slopes
- Our approach
  - Let the internal model values wander
  - Use AI to find constraints in model input
- Q: is constraining inputs enough to control internal model variability?
  - A: yes, see below



Taming variance #1: Use less model

- E.g. local calibration: Boehm '81
  - Only tune 2 vars for linear, exponential effects

- E.g. feature selection:
  - Few variables, less variance (Miller'02)
    - $Y = f(x) = f0 + \sum f_i(x_i) + \sum\sum f_{ij}(x_i,x_j) + \sum\sum\sum f_{ijk}(x_i,x_j x_k) + \ldots$
    - $Var(Y) = V = \sum V_i + \sum\sum V_i V_j + \sum\sum\sum V_i V_j V_k + \ldots.$
  - Menzies et al. Ase'05, TSE'06; Chen et al. IEEE Software '05

- But :
  - The reduced models still exhibit alarming large variances
  - Feature selection still needs data to inform the selection
  - Also it seems wrong-headed to limit modeling
  - Surely the goal should be to extend, not restrict, what we can say?

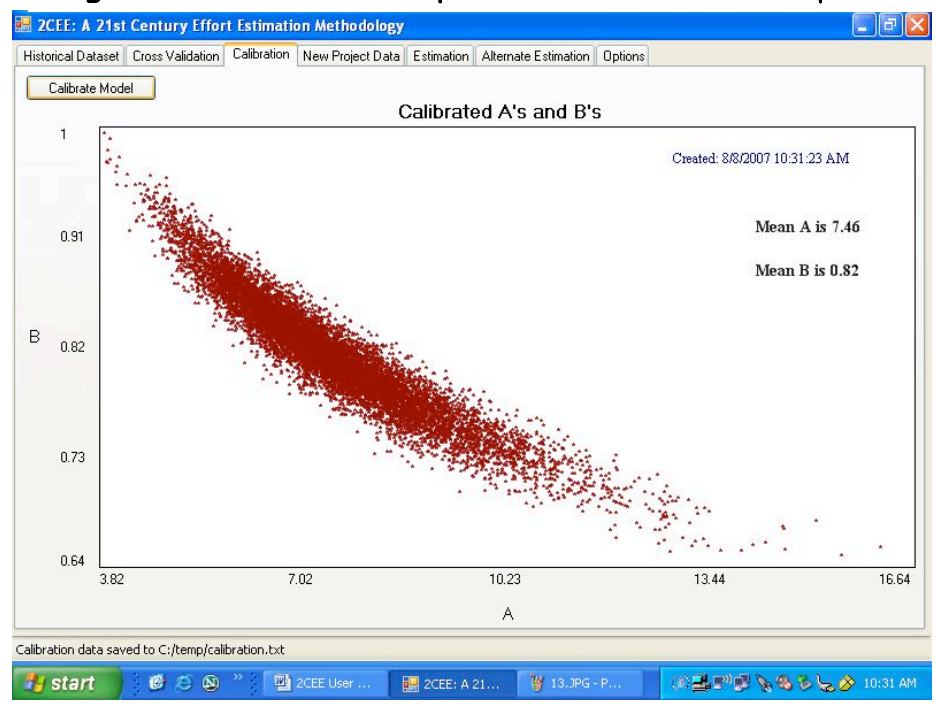
Taming variance #2: Use more data

- May take a while

The data drought

- After 26 years of trying,
  - only < 200 sample projects for COCOMO's database
- Do we need so many?
  - Menzies et al. ICSE'04
  - COCOMO prediction
    - PRED(30)> 70% after 20 records
- But….
  1. COCOMO is a small model and larger models need more data
  2. Finding even 20 records is hard
  3. Subsequent COCOMO simulations showed worrying variance in the conclusions

92* 20*90% samples, local calibration regression to learn slope and"a" and intercept "b"



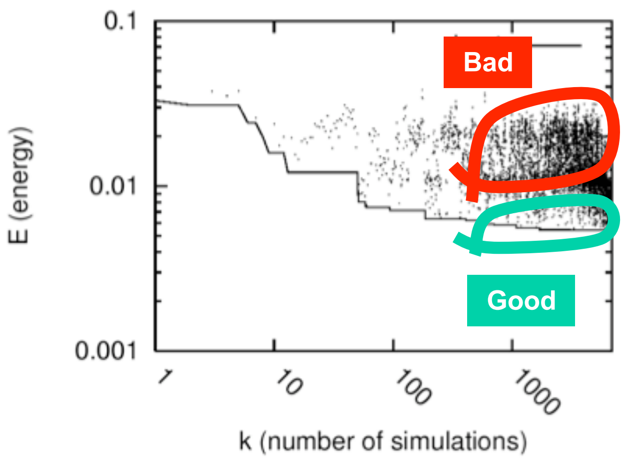
Q: why not reported previously?
A: prior reports discuss mean/median behavior, but not variance.

Taming variance #3: STAR
(1) sample (2) rank (3) try

1) <u>SAMPLE</u> with simulated annealing

Vary the controllables,

Seek lower energies

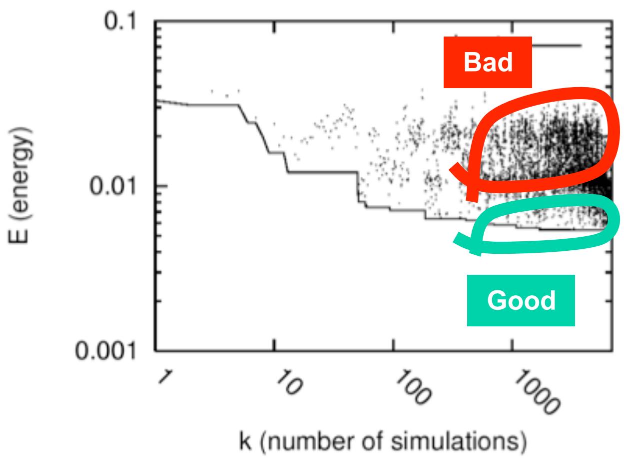


Taming variance #3: STAR
(1) sample (2) rank (3) try

1) SAMPLE with simulated annealing

Vary the controllables,

Seek lower energies



2) RANK (e.g. acap=2)

A = frequency in 10% good

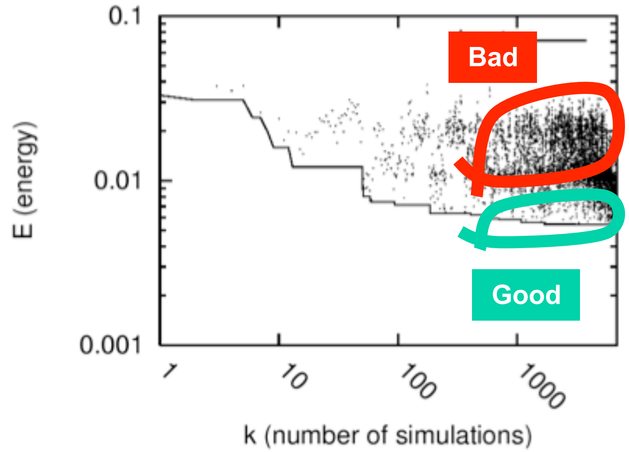
B = frequency in 90% bad

Rank = $a^2 / (a+b)$

Bayesian elite sampling

Taming variance #3: STAR (1) sample (2) rank (3) try

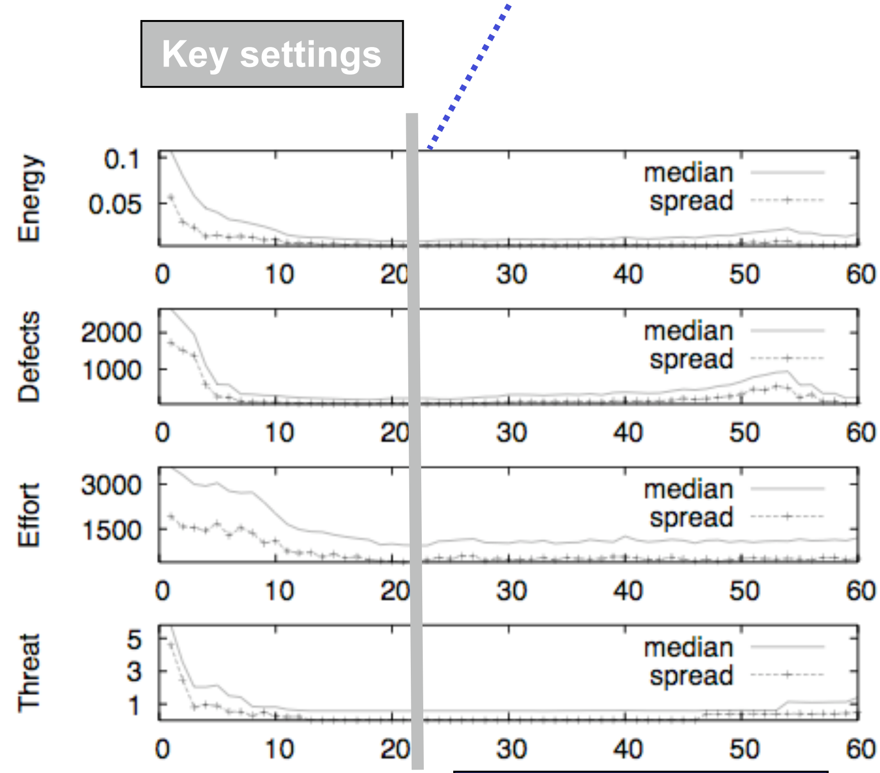
1) SAMPLE with simulated annealing
   - Vary the controllables,
   - Seek lower energies



2) RANK (e.g. acap=2)
   - A = frequency in 20% good
   - B = frequency in 80% bad
   - Rank = $a^2 / (a+b)$

bayesian elite sampling

3) TRY: run*1000 top X ranked items until "min"



Median= 50 percentile
Spread = (75 - 50) percentile

Four COCOMO-family models

- predictions = model(project Options)
  - d = defects = coqualmo (projectOptions) ; ***Chulani '99***
  - f = effort = cocomo(projectOptions) ; ***Boehm et al '81 & '00***
  - m = months = cocomo(projectOptions) ; ***ditto***
  - t = threats = madachyRiskModel(projectOptions) ; ***Madachy '97***

- plan = *least* change to options that *most* improve predictions
  - e = energy= $(\alpha d^2 + \beta f^2 + \chi m^2 + \delta t^2)^{0.5} / (\alpha + \beta + \chi + \delta)^{0.5}$

utilities: $\alpha, \beta, \chi, \delta$

COCOMO-family variables

		strategic?	tactical?
scale factors (exponentially decrease effort)	prec: have we done this before?	✓	
	flex: development flexibility		✓
	resl: any risk resolution activities?		✓
	team: team cohesion		✓
	pmat: process maturity	✓	
upper (linearly decrease effort)	acap: analyst capability	✓	
	pcap: programmer capability	✓	
	pcon: programmer continuity	✓	
	aexp: analyst experience	✓	
	pexp: programmer experience	✓	
	ltex: language and tool experience	✓	
	tool: tool use		✓
	site: multiple site development	✓	
	sced: length of schedule		✓
lower (linearly increase effort)	rely: required reliability		
	data: secondary memory storage requirements		✓
	cplx: program complexity		✓
	ruse: software reuse		✓
	docu: documentation requirements		✓
	time: runtime pressure		
	stor: main memory requirements		✓
	pvol: platform volatility		
COQUALMO defect removal methods	auto: automated analysis	✓	✓
	execTest: execution-based testing tools	✓	✓
	peer: peer reviews	✓	✓

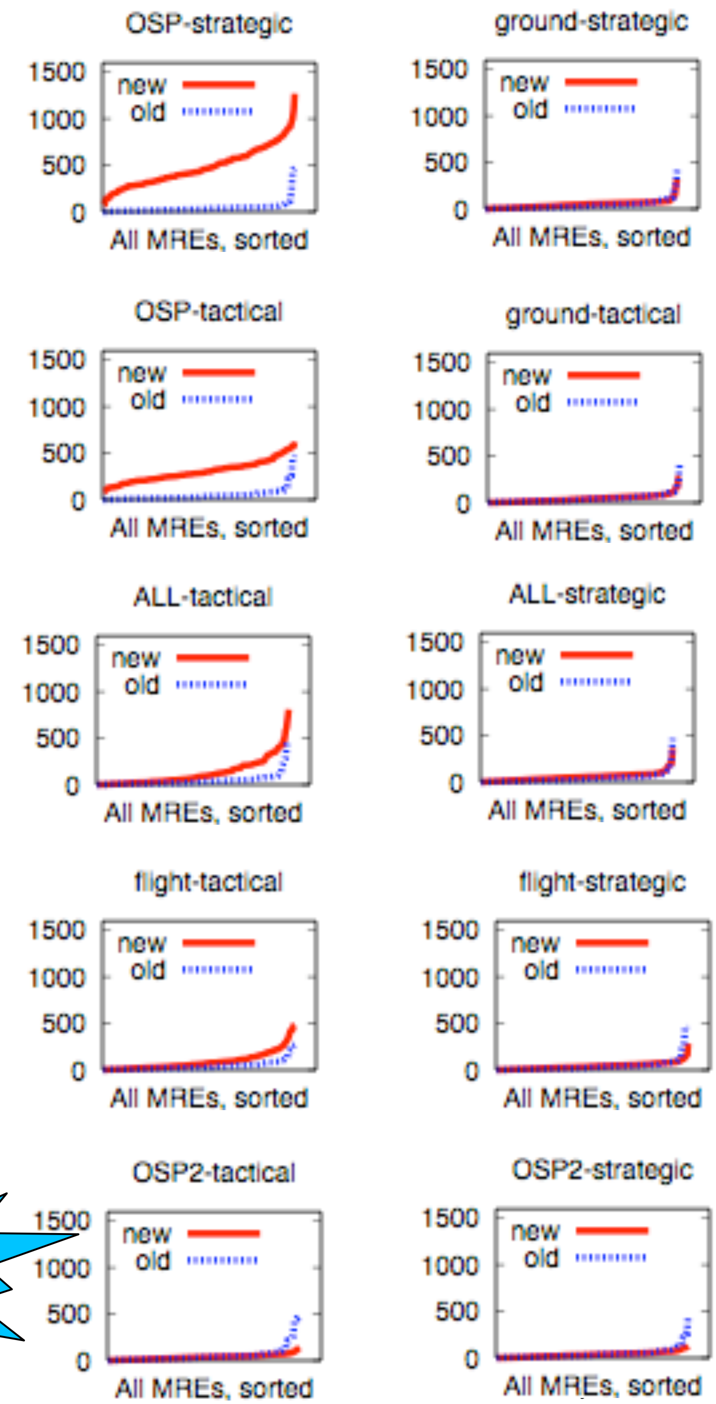
Can be changed intra-project

Cannot

Results

- P = Generate projects from the minimum energy point, estimate each with STAR
- Using Boehm's LC procedure
  - Train on historical NASA projects,
  - "new": Test on P,
    - Generate deltas comparing LC estimates to STAR's
  - "old": Test on NASA historical data,
    - Generate deltas comparing LC estimates to actuals in NASA data
- Sometimes, old and new deltas are different
  - Lesson1: stochastics introduce unknown factors (so use local data, if possible)
- Usually, old and new deltas very close
  - Even though STAR and LC have different goals
  - Lesson2: if you can't get old data, it is still possible to make process predictions and decisions.

Median delta of area under curve = 40%



How “big” is a 40% delta?

- a = try controlling anything
- s = try control the inter-project strategic factors

  {prec pmat acap pcap pcon aexp pexp ltex site auto execTest peerReview}
- t = try control the intra-project tactical factors

  {flex resl team tool sced data cplx ruse docu stor auto execTest peerReview}

project	ALL			OSP			OSP2			flight			ground		
policies	a	s	t	a	s	t	a	s	t	a	s	t	a	s	t

JPL flight and ground systems

standard COCOMO, no restrictions

Two generations of a NASA GNC system

STAR, outputs
reduction% = final / initial

- a = try controlling anything
- s = try control the inter-project strategic factors

  {prec pmat acap pcap pcon aexp pexp ltex site auto execTest peerReview}
- t = try control the intra-project tactical factors

  {flex resl team tool sced data cplx ruse docu stor auto execTest peerReview}

project	ALL			OSP			OSP2			flight			ground		
policies	a	s	t	a	s	t	a	s	t	a	s	t	a	s	t
effort	6	14	55	44	73	67	89	74	112	15	24	64	19	24	67
defects	1	14	10	15	21	13	12	12	17	2	24	22	14	7	12
threat	0	0	*106*	93	*111*	68	0	0	0	0	0	0	0	0	0
months	37	50	59	69	90	81	86	91	95	50	61	81	55	62	82

- Mostly: very large defect reductions
- Often: large effort reductions
- Least reductions in OSP2. Why?

Least reduction is OSP2. Why?

- "OSP2" :
  - the most restricted problem processed to date.
  - Achieved the least reductions
  - If you fix everything,
    - There's nothing left to fix

	ranges			values	
project	feature	low	high	feature	setting
OSP: Orbital space plane	prec	1	2	data	3
	flex	2	5	pvol	2
	resl	1	3	rely	5
	team	2	3	pcap	3
	pmat	1	4	plex	3
	stor	3	5	site	3
	ruse	2	4		
	docu	2	4		
	acap	2	3		
	pcon	2	3		
	apex	2	3		
	ltex	2	4		
	tool	2	3		
	sced	1	3		
	cplx	5	6		
	KSLOC	75	125		
OSP2	prec	3	5	flex	3
	pmat	4	5	resl	4
	docu	3	4	team	3
	ltex	2	5	time	3
	sced	2	4	stor	3
	KSLOC	75	125	data	4
				pvol	3
				ruse	4
				rely	5
				acap	4
				pcap	3
				pcon	3
				apex	4
				plex	4
				tool	5
				cplx	4

Conclusion

- A little AI goes a long way
  - Simulated annealing
    + elite bayesian sample
  - Simple to code
- The right project decisions can tame variance
  - Models contain “key constraints”
  - Set the keys via project decisions
  - Shown here: setting the keys
    - Reduces variance
    - While improving targets
      - Effort (cost), month (schedule), defects, threats

- Don’t need to know everything before you plan
  - Tuning process models to local data is the preferred options.
  - But unturned models can be surprisingly effective
- Uncertainty is an ally
  - Don’t delay in seeking stable conclusions within a space of partially defined options
  - If you fix everything, there’s nothing left to fix.

Process models: *ganz falsch*?

- What is the effect on model output from internal model uncertainty?
  - Can that variance be tamed:
    - Without additional data?
    - Without discarding parts of the model?
  - If not, will Dr.Pauli revoke our license to model?
    - “Not even false”

- At least for COCOMO-family models,
  - We can find definite conclusions from process models, despite the data drought
  - Method
    - Find the key constraints
    - Constrain the keys
    - Tame uncertainty
  - More process planning, earlier, with less data

Questions?
Comments?