



# Managing Uncertainty in Value-based SE



West Virginia University.

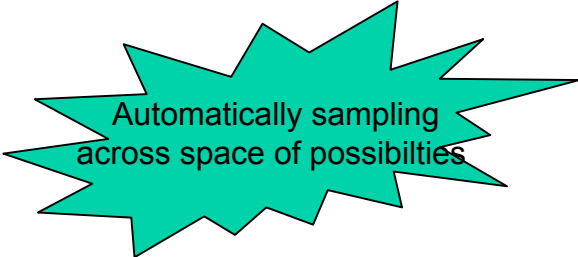
Tim Menzies (tim@menzies.us)  
Phillip Green II,  
Oussama Elwaras

10/27/08  
23rd International Forum on  
COCOMO and Systems/Software Cost Modeling



# This is two talks

- ➔ One on value-based SE
- ➔ Another on how and why we want to....



Automatically sampling  
across space of possibilities



Without  
calibration

- ➔ <http://unbox.org/wisp/tags/star>



# Problems, and Solutions?

- ➔ “I need data. I want I want I want . We keep saying this and we don’t get it. So what do we do?”
  - Stop calibrating our models
  - Automatically sample across space of possible calibrations
- ➔ “Need more trade studies”
  - Automatically sample across space of possibilities
  - Days to define goals, seconds to run the trade study
- ➔ “Death to point estimates”
  - Report results from an automatic sample across a space of possibilities.
- ➔ “Cost is not enough”
  - Search space of possibilities for methods to improve a value function
- ➔ “Need more models of different types”
  - Generate skeletons of expert intuitions
  - Sample across space of possibilities within the space of possibilities.

# PROMISE '09



**CALL FOR PAPERS**  
<http://promisedata.org/2009>

## PROMISE 2009

International Conference on  
Predictor Models in  
Software Engineering –  
Vancouver, Canada  
May 18-19, 2009

The PROMISE conference leverages the successful experience from the four previous workshops. The objective of the conference is to deliver to the software engineering community useful, usable, verifiable, and repeatable models applicable to better manage software processes and projects (<http://promisedata.org/2009>).

**Important Due Dates:**

- ✓ Abstracts: Jan 12, 2009
- ✓ Submissions: Jan 26, 2009
- ✓ Author's notification: Mar 2, 2009
- ✓ Camera ready papers: Mar 16, 2009

A CO-LOCATED EVENT WITH  
ICSE 2009

- ➔ [www.promisedata.org/2009](http://www.promisedata.org/2009)
- ➔ Reproducible SE results
- ➔ Papers:
  - and the data used to generate those papers
  - [www.promisedata.org/data](http://www.promisedata.org/data)
- ➔ Keynote speaker:
  - Barry Boehm, USC
- ➔ Motto:
  - Repeatable, refutable, improvable
  - Put up or shut up



**Do We Need to  
Calibrate Models?**



# Sources of estimation error

- ⇒ Estimate = projectDetails \* modelCalibration
  - Estimate error = projectError and calibrationError

- ⇒ We must have accurate modelCalibration when...

$$\text{Estimate} = \text{projectDetails} * \text{modelCalibration}$$

- ⇒ But we don't when...

$$\text{Estimate} = \text{projectDetails} * \text{modelCalibration}$$

# Calibration vs Project uncertainty: David vs Goliath?



project	feature	ranges		values	
		low	high	feature	setting
Flight:	rely	3	5	tool	2
	data	2	3	seed	3
	cplx	3	6		
	time	3	4		
	stor	3	4		
	pvol	2	4		
	acap	3	5		
	apex	2	5		
	pcap	3	5		
	plex	1	4		
	ltex	1	4		
	pmat	2	3		
	Ksloc	7	418		
Ground:	rely	1	4	tool	2
	data	2	3	seed	3
	cplx	1	4		
	time	3	4		
	stor	3	4		
	pvol	2	4		
	acap	3	5		
	apex	3	5		
	pcap	3	5		
	plex	1	4		
	ltex	1	4		
	pmat	2	3		
	Ksloc	11	392		

id	features	relative weight
1	Personnel/team capability	3.53
2	Product complexity	2.38
3	Time constraint	1.63
4	Required software reliability	1.54
5	Multi-site development	1.53
6	Doc. match to life cycle	1.52
7	Personnel continuity	1.51
8	Applications experience	1.51
9	Use of software tools	1.50
10	Platform volatility	1.49
11	Storage constraint	1.46
12	Process maturity	1.43
13	Language & tools experience	1.43
14	Required dev. schedule	1.43
15	Data base size	1.42
16	Platform experience	1.40
17	Arch. & risk resolution	1.39
18	Precedentedness	1.33
19	Developed for reuse	1.31
20	Team cohesion	1.29
21	Development mode	1.32
22	Development flexibility	1.26

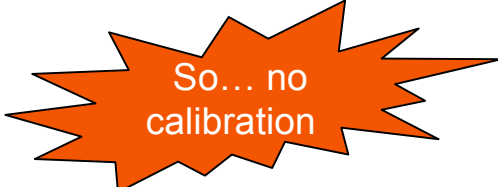
Figure 1: Relative effects on development effort. Data from a regression analysis of 161 projects [?].

$$\begin{aligned}
 11,022 = & \\
 & 3.53 * 2.38 * \\
 & 1.63 * 1.54 * \\
 & 1.53 * \\
 & 1.52 * 1.51 * \\
 & 1.51 * 1.5 * \\
 & 1.49 * 1.46 * \\
 & 1.43 * 1.43 * \\
 & 1.43 * 1.42 * \\
 & 1.4 * 1.39 * \\
 & 1.33 * 1.31 * \\
 & 1.29 * 1.32 * \\
 & 1.26.
 \end{aligned}$$



# An experiment

- ➔ Monte Carlo sampling over ...
  - ... the space of possible calibrations
  - ... the project options
- ➔ Apply AI search methods to select
  - Project options that most improve the estimate
  - But do not try to control the calibrations
- ➔ Q: Is controlling just project options enough to control estimates?
  - A: yes, if...



So... no calibration

Estimate = **projectDetails** \* **modelCalibration**





**Why even try?**

**(Problems with  
Calibration)**

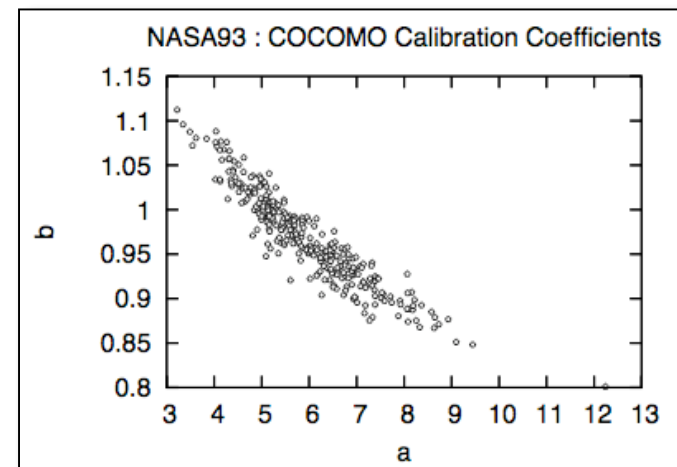
# Variance in COCOMO calibrations

⇒ Much larger than reported:

- For 93 NASA records from Hihn
- For 63 records from Boehm81

⇒ Makes a nonsense of reports of the form

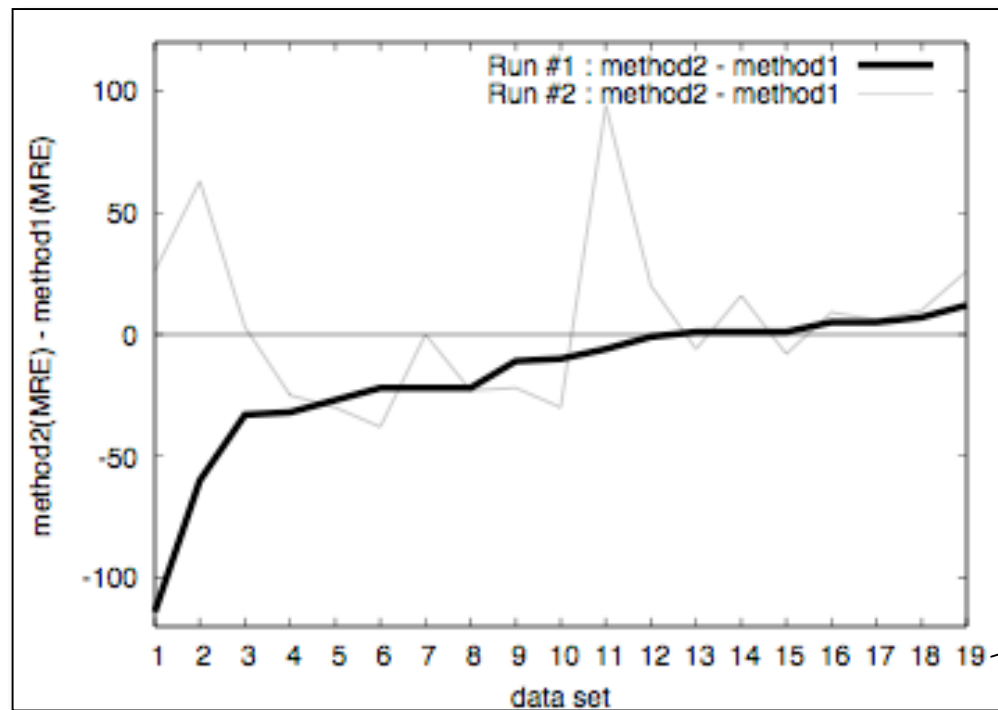
- “A = 2.95, B= 1.01”
- “Method A is better than method B for calibrating COCOMO”
- “There are best subsets of the COCOMO features.”
- “Hooray: I’ve improved MMRE / PRED(25) by 5%”<sup>å</sup>



1000 \* {remove any 10, run LC on rest}

# Variance problems

## Two runs of a 10-way cross-val

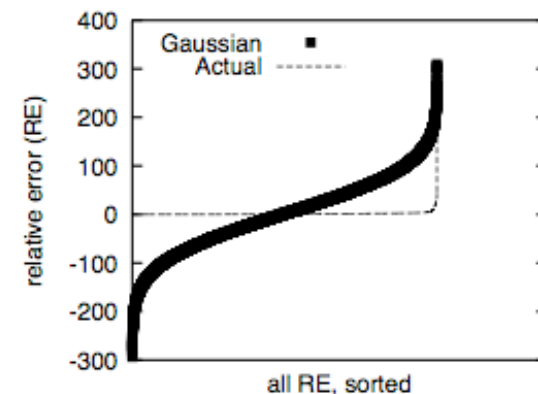
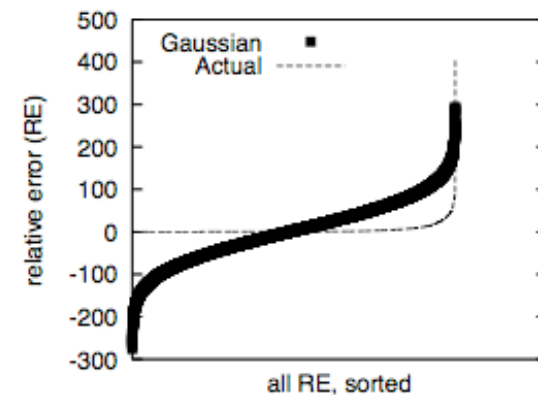


When  $< 0$ ,  
method2 better than method 1

Data sets sorted by run1 results

# Evaluation issues

- ➔ If you do multiple experiments with
  - S subsets
  - L learners
  - P pre-processes
  - Repeated N times
- ➔ Then somewhere in  $N*S*L*P$ 
  - Occasional massive outliers
  - Highly non-Gaussian
- ➔ Except in the COCOMO community
  - “mean” is deprecated
  - Not “1” but “first”
  - Ranked statistics, not ordinal statistics
    - Mann-Whitney, Wilcoxon
    - E.g. see Kitchenham TSE'07 review of studies
- ➔ Strongly recommend  $AR = \text{predicted-actual}$



# Cost driver instability (what can we throw away without hurting estimation accuracy)

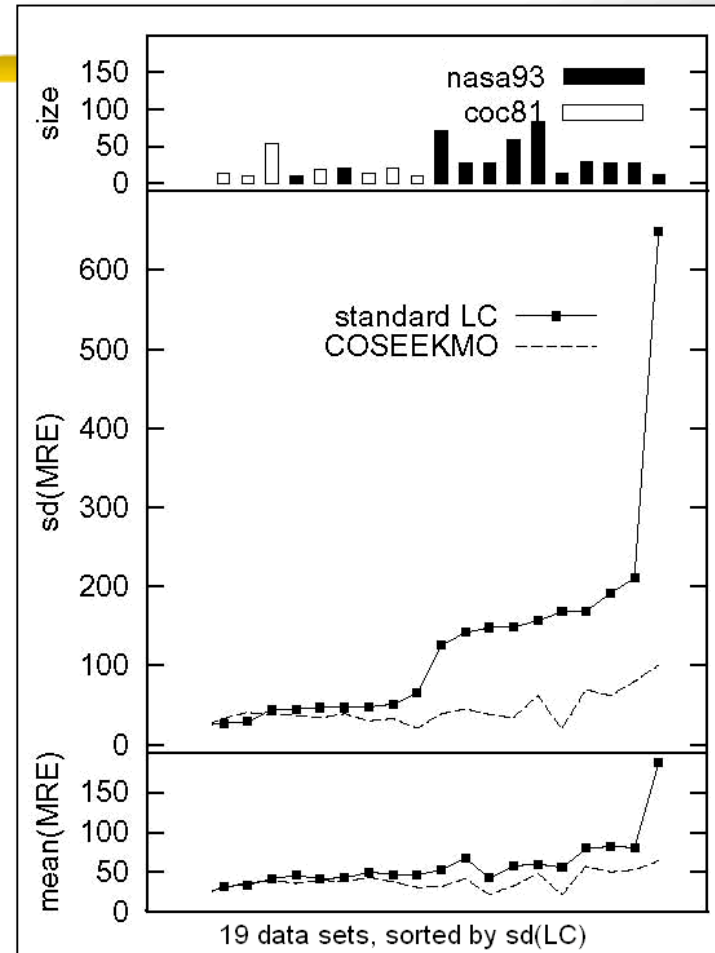
Data Subset	COCOMO 81 Cost Drivers															Number of Significant
	acap	time	cplx	aexp	virt	data	tum	rely	stor	lexp	pcap	modp	vexp	sced	tool	Cost Drivers
coc81_all	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	15
coc81_mode_embedded	○	●	○	○	●	○	○	○	○	●	▨	●	●	●	●	14
coc81_mode_organic	●	●	○	●	●	●	●	▨	○	▨	●	●	●	●	●	13
nasa93_all	●	●	▨	●	●	●	●	●	●	▨	▨	▨	▨	▨	▨	8
nasa93_mode_embedded	○	●	●	▨	●	●	●	●	●	○	○	▨	▨	▨	●	11
nasa93_mode_semidetached	●	▨	▨	●	▨	▨	▨	▨	▨	▨	▨	▨	○	▨	▨	3
nasa93_fg_ground	●	▨	○	●	▨	▨	▨	▨	▨	●	○	▨	▨	▨	▨	5
nasa93_category_missionplanning	○	●	●	▨	▨	●	●	●	▨	▨	●	○	▨	○	▨	9
nasa93_category_avionicsmonitoring	●	▨	▨	○	▨	▨	▨	▨	▨	▨	▨	●	○	○	○	6
nasa93_year_1975	●	●	●	●	●	●	▨	●	●	○	○	▨	▨	▨	▨	10
nasa93_year_1980	●	●	●	○	●	●	●	●	●	▨	▨	▨	▨	●	○	11
nasa93_center2	●	●	●	●	●	○	●	○	●	●	●	●	●	▨	●	14
nasa93_center5	▨	●	●	○	●	●	○	●	●	○	▨	▨	▨	▨	▨	9
nasa93_project_gro	○	○	●	○	●	▨	●	○	○	●	○	●	●	▨	○	13
nasa93_project_sts	▨	●	●	▨	●	●	●	●	●	▨	▨	▨	▨	▨	▨	7
<b>Usually Significant</b>	<b>5</b>	<b>1</b>	<b>3</b>	<b>5</b>	<b>0</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>4</b>	<b>1</b>	<b>2</b>	<b>2</b>	<b>3</b>	
<b>Always Significant</b>	<b>8</b>	<b>11</b>	<b>9</b>	<b>7</b>	<b>11</b>	<b>9</b>	<b>9</b>	<b>8</b>	<b>8</b>	<b>5</b>	<b>4</b>	<b>6</b>	<b>5</b>	<b>5</b>	<b>4</b>	
<b>Total Number of Significant Occurrences</b>	<b>13</b>	<b>12</b>	<b>12</b>	<b>12</b>	<b>11</b>	<b>11</b>	<b>11</b>	<b>11</b>	<b>11</b>	<b>8</b>	<b>8</b>	<b>7</b>	<b>7</b>	<b>7</b>	<b>7</b>	

Legend:

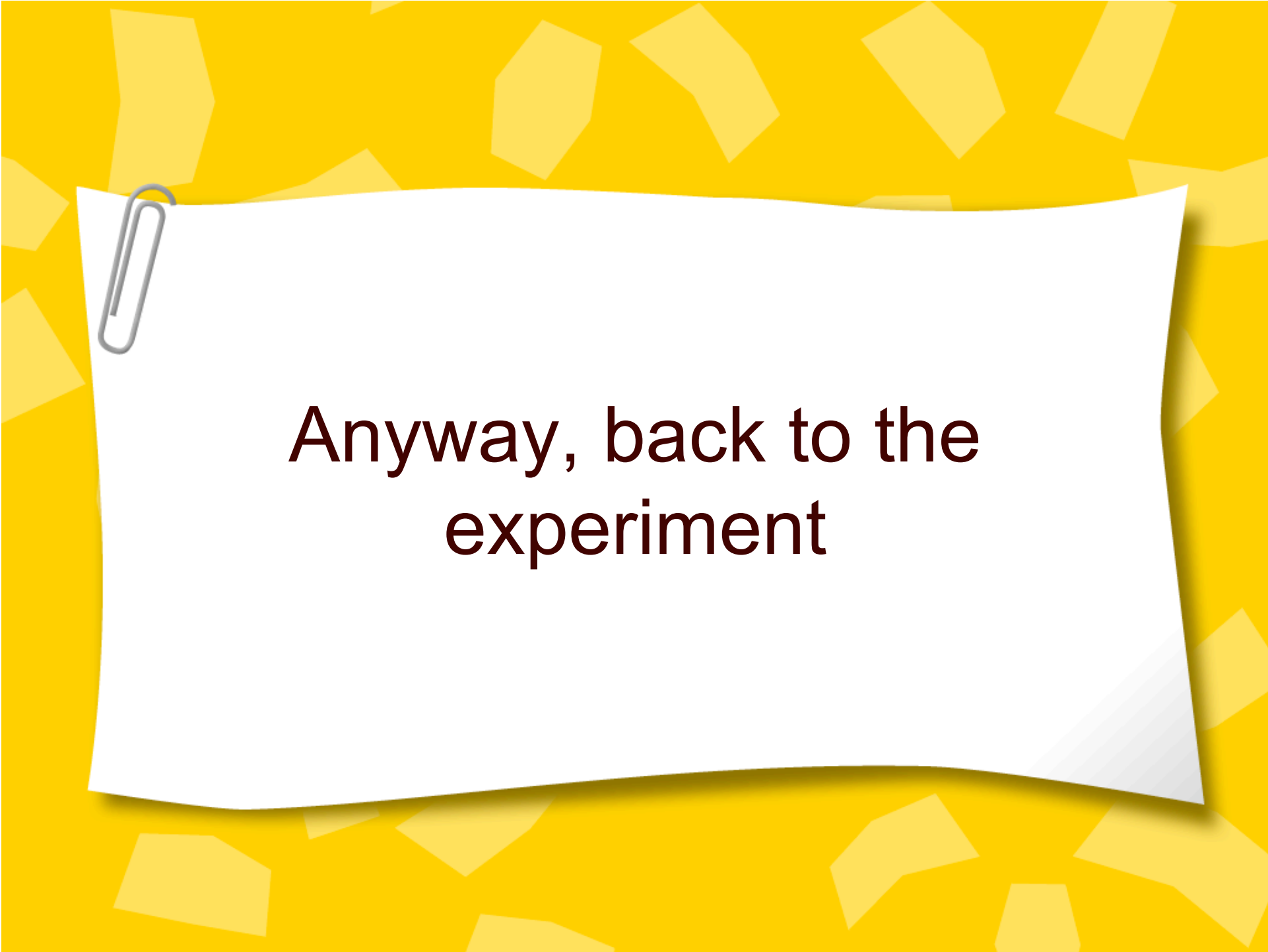
- = Not significantly different than 10 at a 95% Confidence Interval
- = Not significantly different than 9 or greater at a 95% Confidence Interval

# Solving the variance problem?

- ➔ More data?
  - Yeah, that's easy to do
  - And it may not help
- ➔ Feature subset selection
  - Chen'05 (USC)
  - Lum, Hihn '06 (JPL): see last slide
- ➔ Constrain the learning
  - “A Constrained Regression Technique for COCOMO Calibration”
  - Nguyen & Steece & Boehm
  - Cocomo Forum '08



30 \* {test = any 10, train = all - test}



**Anyway, back to the  
experiment**

# What is the space of project options?

project	ranges			values	
	feature	low	high	feature	setting
Flight:	rely	3	5	tool	2
	data	2	3	sced	3
	cplx	3	6		
	time	3	4		
	stor	3	4		
	pvol	2	4		
	acap	3	5		
	apex	2	5		
	pcap	3	5		
	plex	1	4		
	ltex	1	4		
	pmat	2	3		
	Ksloc	7	418		
	Ground:	rely	1	4	tool
data		2	3	sced	3
cplx		1	4		
time		3	4		
stor		3	4		
pvol		2	4		
acap		3	5		
apex		3	5		
pcap		3	5		
plex		1	4		
ltex		1	4		
pmat		2	3		
Ksloc		11	392		

“Values” = fixed

“Ranges” = Loose (select within these ranges)

project	ranges			values	
	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
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



# What is the space of possible calibrations?

## → COCOMO effort estimation

- Effort multipliers are straight (ish) lines
- when  $EM = 3 = \text{nominal}$ ...
  - multiple effort by one (I.e. nothing)
- i.e. they pass through the point  $\{3,1\}$ ;

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

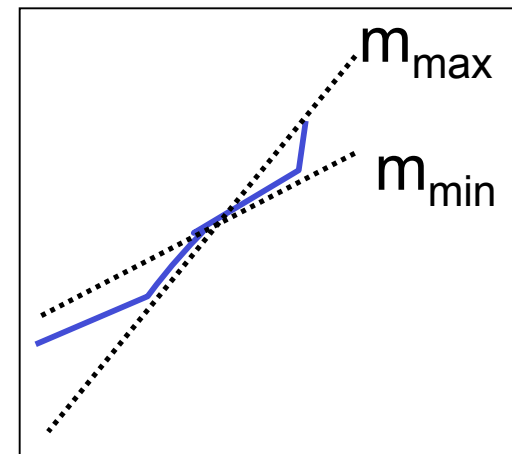
$$(0.073 \leq m_a^+ \leq 0.21) \wedge (-0.178 \leq m_a^- \leq -0.078)$$

Increase effort

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

decrease effort

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

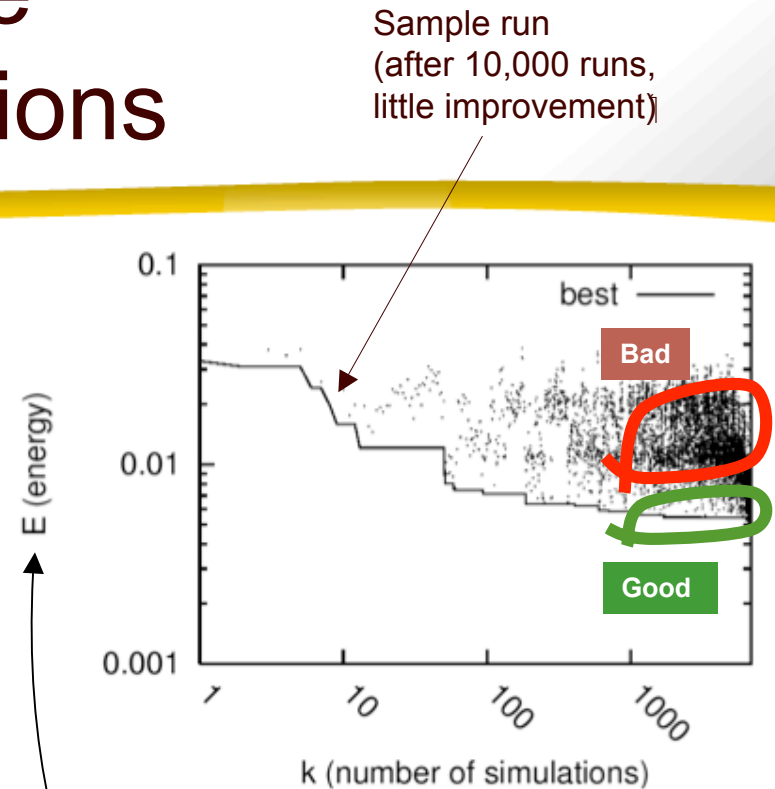


Repeat for  
Scale factors

Repeat for  
COQUALMO

# Searching the space of options + calibrations

- Using simulated annealing, Monte Carlo simulated annealing across intersection of
  - A particular project type
  - Space of possible tunings
- Rank options by frequency in **good**, not **bad**
- For  $r$  options
  - Try setting the  $1 \leq x \leq R$  top ranked options
  - Simulate (100 times) to check the effect of options 1 ..  $x$
- Smile if
  - Reduced median and variance in defects/ efforts/ time/ threats



But what is the  
Performance score?

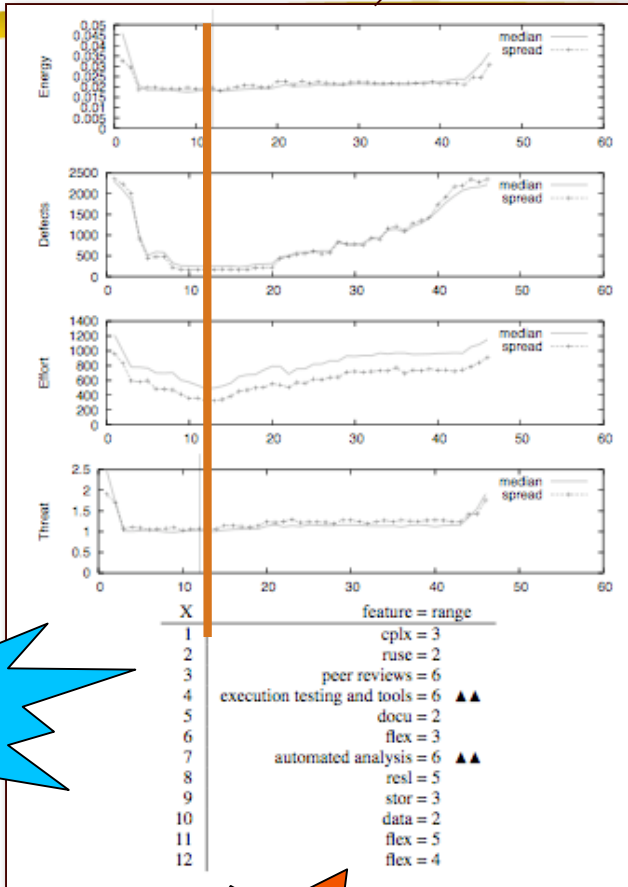
Automatically sampling  
across space of possibilities

Note: no  
calibration

# Results: JPL flight systems (GNC) (controlling just “tactical” features)

flex resl stor  
data ruse docu  
tool sced cplx  
aa ebt pr


project	feature	ranges		values	
		low	high	feature	setting
Flight:	rely	3	5	tool	2
	data	2	3	seed	3
	cplx	3	6		
	time	3	4		
	stor	3	4		
	pvol	2	4		
	acap	3	5		
	apex	2	5		
	pcap	3	5		
	plex	1	4		
	ltex	1	4		
	pmat	2	3		
	Ksloc	7	418		
Ground:	rely	1	4	tool	2
	data	2	3	seed	3
	cplx	1	4		
	time	3	4		
	stor	3	4		
	pvol	2	4		
	acap	3	5		
	apex	3	5		
	pcap	3	5		
	plex	1	4		
	ltex	1	4		
pmat	2	3			
Ksloc	11	392			



**Automated Trade studies**

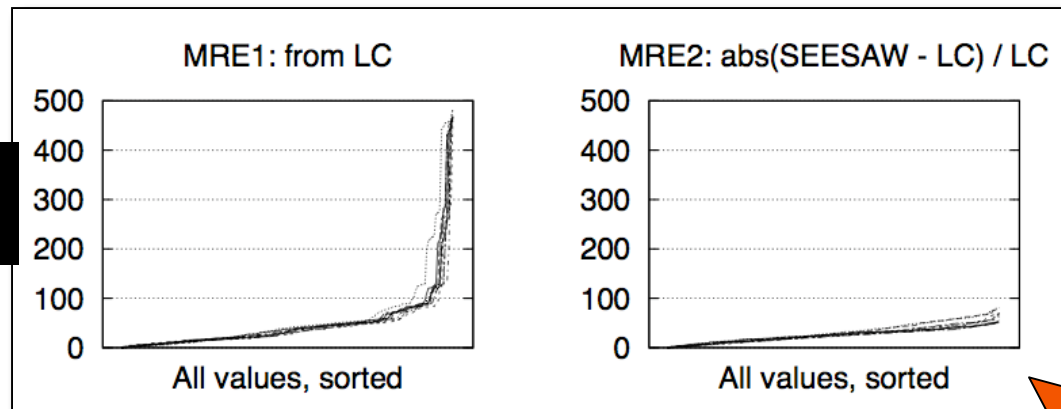
Automatically sampling across space of possibilities

Note: no calibration



# AI search's effort estimates are (almost) the same as LC

Ten case studies



Note: no calibration

→ So...

Estimate = **projectDetails** \* modelCalibration

→ What can we use this for?



**Managing Uncertainty  
in Value-based SE**



# Two Goal Functions

## → “ENERGY”

- a domain general “value” proposition
- Menzies, Boehm, Madachy, Hihn, et al, [ASE 2007]
- Reduce effort, defects, schedule

## → “Huang06” :

- minimize a local value proposition
- A variant of USC Ph.D. thesis
  - [Huang 2006]: Software Quality Analysis: a Value-Based Approach
- Balances beating everyone to market against more/worse bugs
  - and being last to market with few/minor bugs

```
(defun energy ()  
  "Calculates energy based on cocomo pm, tdev, coqualmo defects,  
  Madachy's risk."  
  (let* ((npm (calc-normalized-pm))  
         (ntdev (calc-normalized-tdev))  
         (ndefects (calc-normalized-defects))  
         (nrisk (calc-normalized-risk))  
         (pm-weight 1)  
         (tdev-weight 1)  
         (defects-weight (+ 1 (expt 1.8 (- (xomo-rating? 'rely) 3))))  
         (risk-weight 1))  
    (/ (sqrt (+ (expt (* npm pm-weight) 2)  
                (expt (* ntdev tdev-weight) 2)  
                (expt (* ndefects defects-weight) 2)  
                (expt (* nrisk risk-weight) 2))))  
       (sqrt (+ pm-weight tdev-weight  
                defects-weight risk-weight))))))
```

```
(defun risk-exposure ()  
  "Calculates risk exposure based on rely"  
  (let* ((pm (calc-pm))  
         (size-coefficient (calc-size-coefficient 'rely)))  
    (defects (calc-defects))  
    (defects_vl (calc-defects-with-vl-rely))  
    (loss-probability (/ defects defects_vl))  
    (loss-size (* (expt 3 (/ (- (xomo-rating? 'cplx) 3) 2) )  
                 size-coefficient  
                 pm))  
    (software-quality-re (* loss-probability loss-size))  
    (market-coefficient (calc-market-coefficient 'rely))  
    (market-erosion-re (* market-coefficient pm))  
    (+ software-quality-investment-re  
       market-erosion-re)))
```

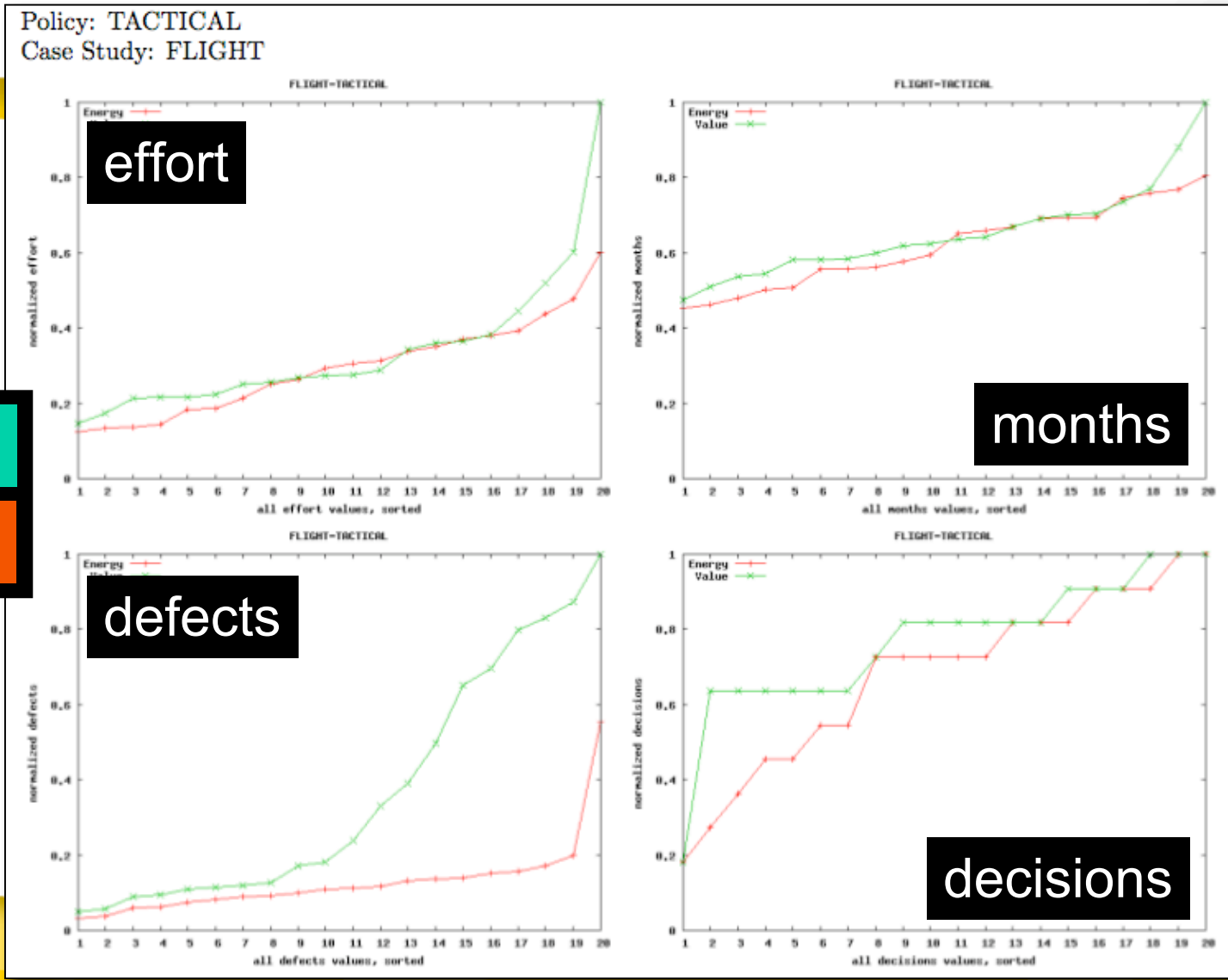
Note: no calibration

# JPL Flight systems: Tactical

20 times, find the fewest decision that lead to min {effort, months, defects}



value  
energy



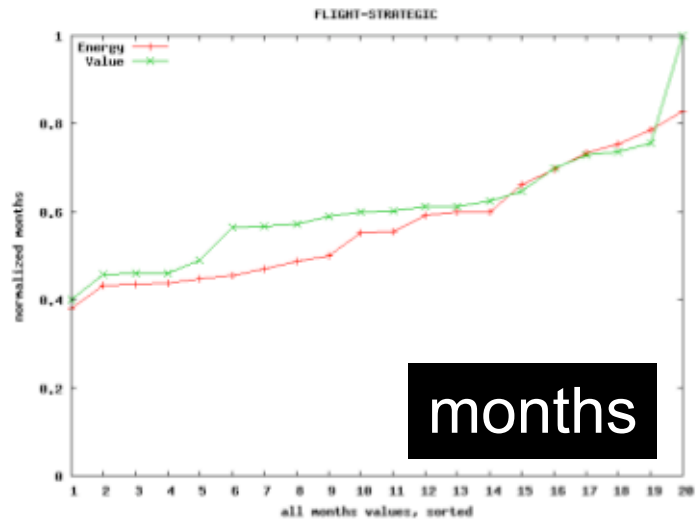
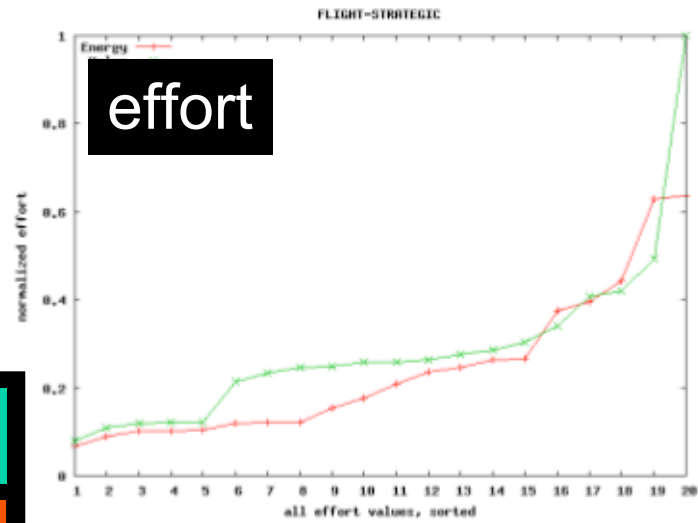
Note: no calibration

# JPL Flight systems: Strategic

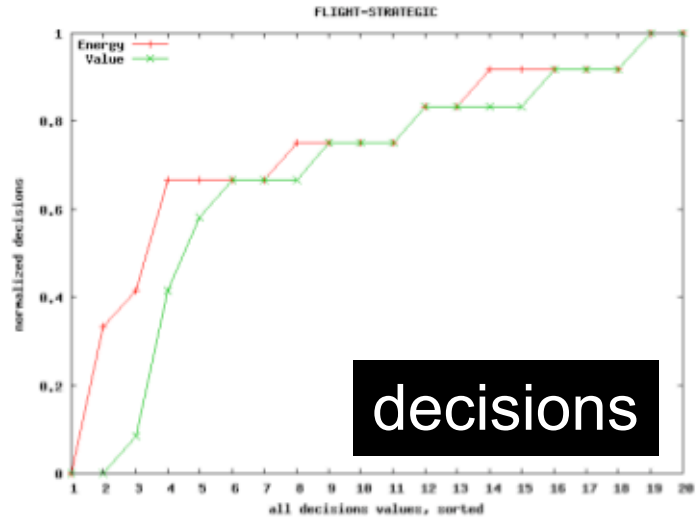
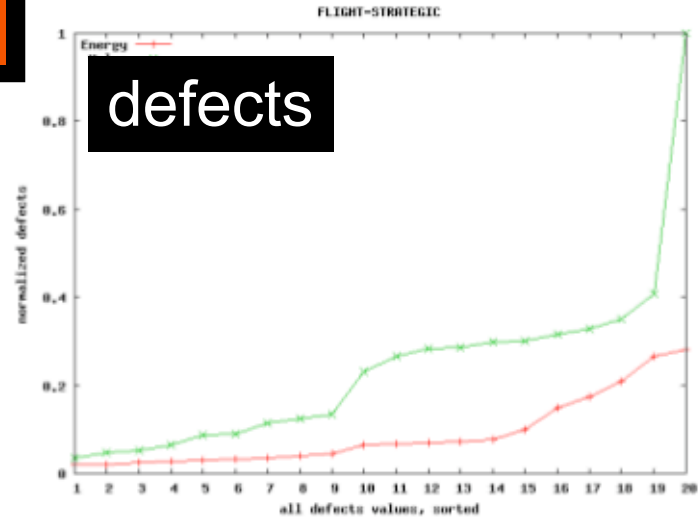
20 times, find the fewest decision that lead to min {effort, months, defects}



Policy: STRATEGIC  
Case Study: FLIGHT



value  
energy

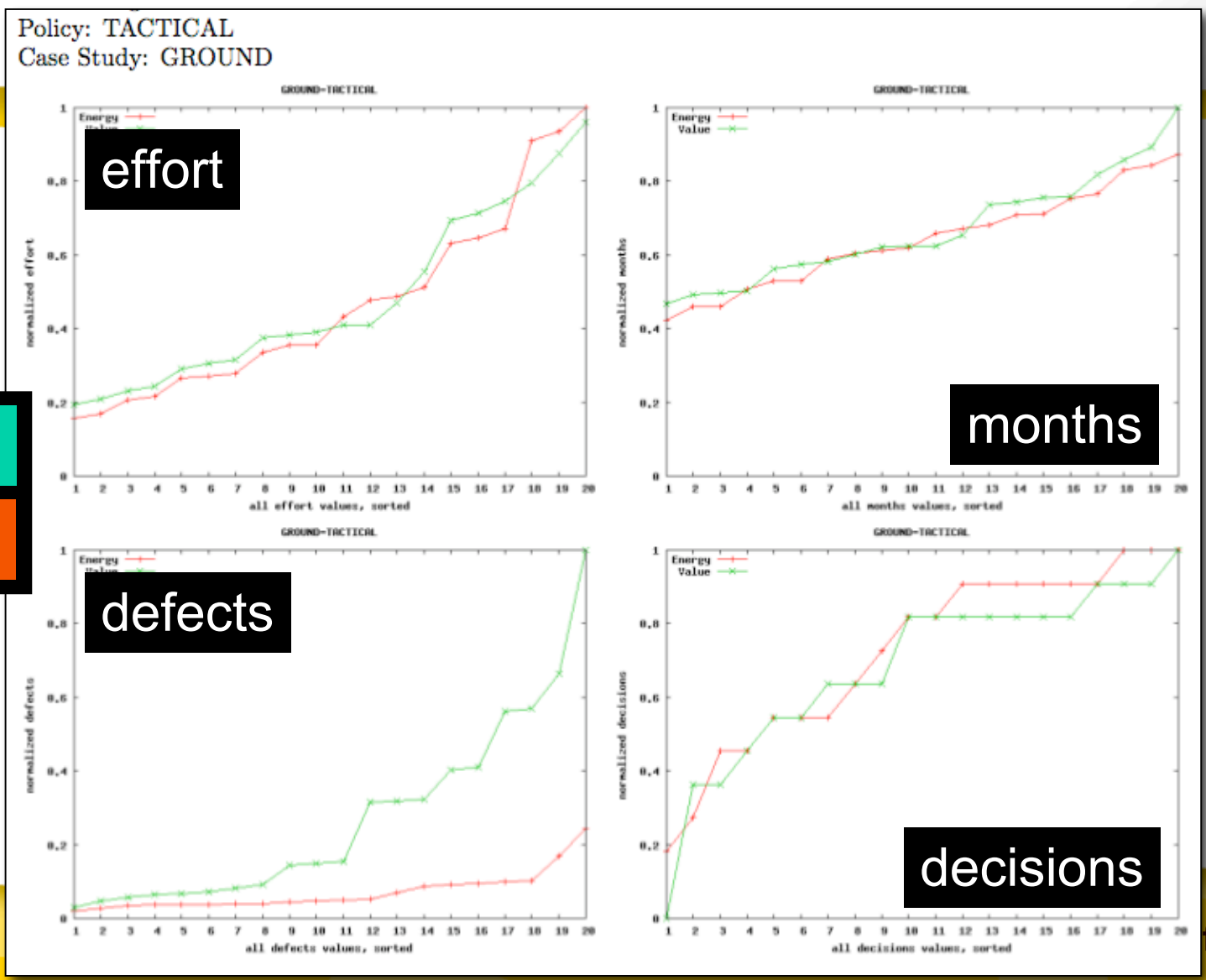




Note: no calibration

# JPL Ground systems: Tactical

20 times, find the fewest decision that lead to min {effort, months, defects}



effort

months

defects

decisions

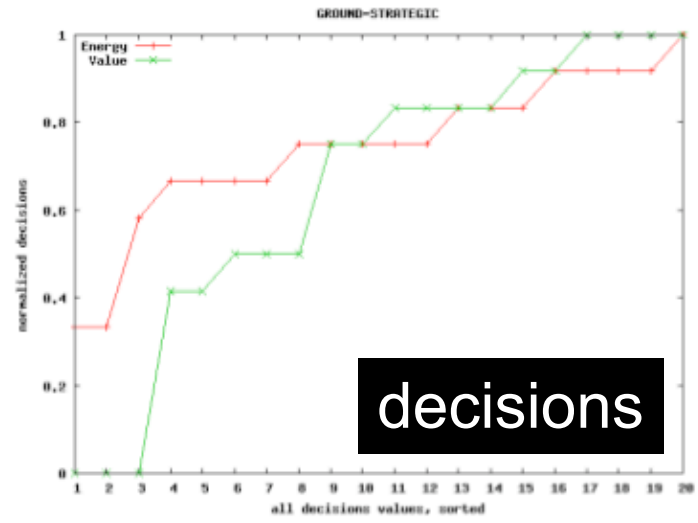
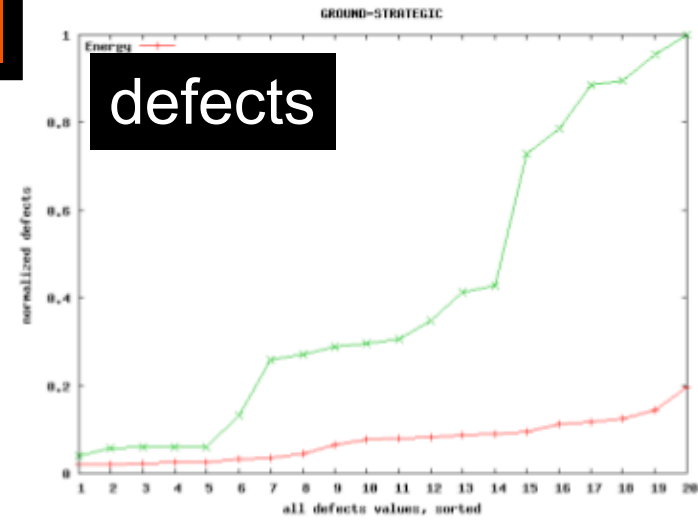
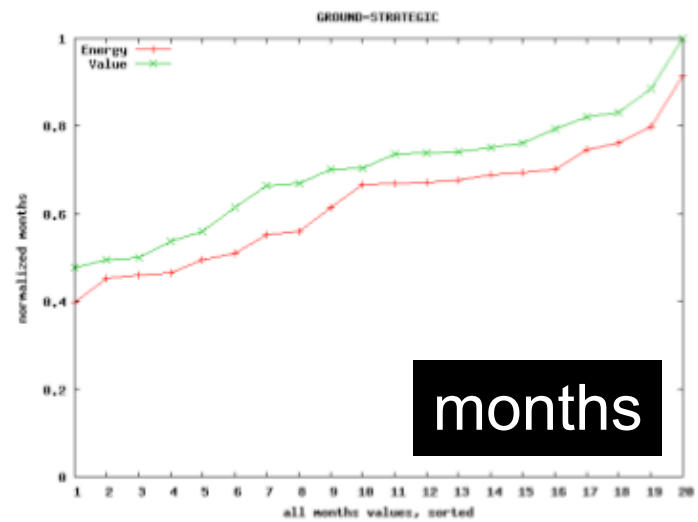
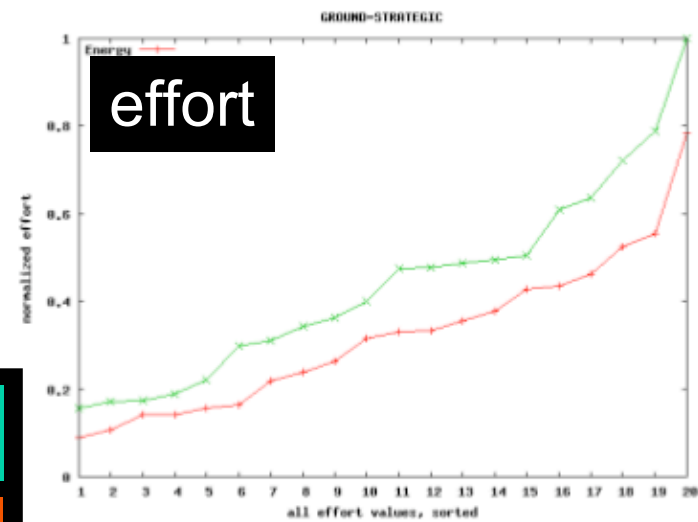
value  
energy

Note: no calibration

# JPL Ground systems: Strategic

20 times, find the fewest decision that lead to min {effort, months, defects}

Policy: STRATEGIC  
Case Study: GROUND



value  
energy



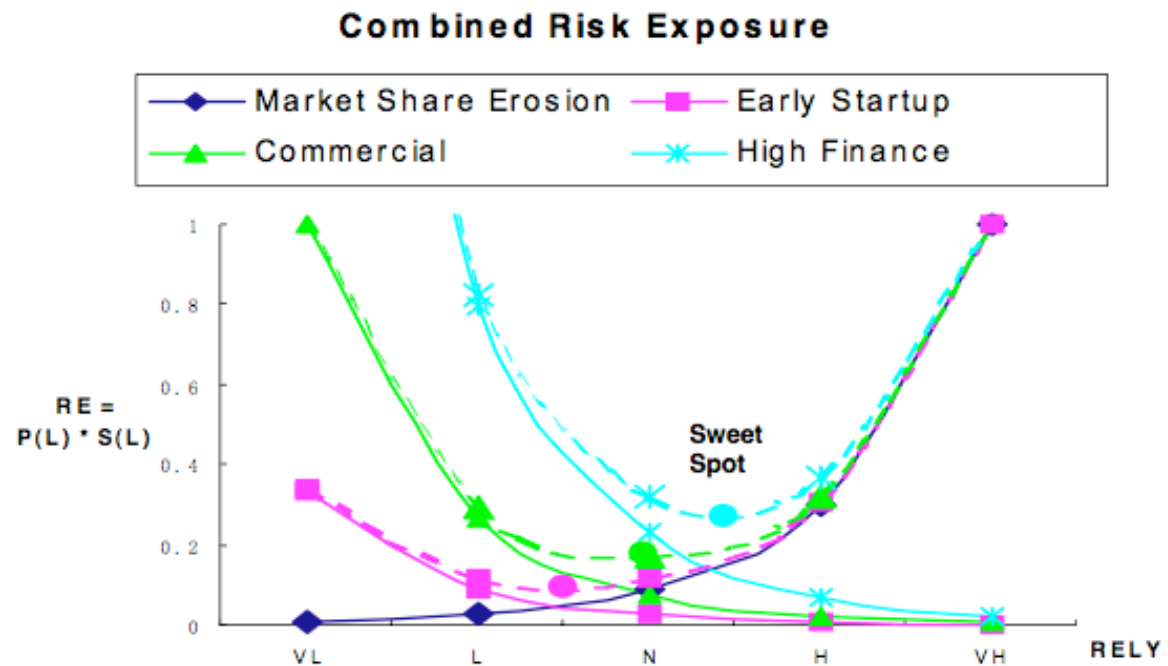
# Patterns

- ➔ With value-based  
(compared to value-neutral energy)
  - effort and months:
    - same, same, same, (a little) more
  - Decisions:
    - more, less, same, less
  - Defects:
    - more, more, more, more

# Note: we are not the first to say value $\neq$ defects

→ From [Huang06]

→ Infinitely increasing software reliability is not necessarily the best plan



Huang06: analysis across one dimension  
Here: analysis across 25 dimensions




# Conclusions



# An End to Calibration?

- ➔ No
- ➔ If the data is available
  - And if calibration results in precise tunings
    - Low variance
  - Then use calibration
- ➔ Else
  - You can still make rank different process options
  - So we still decide without data
  - (But better data = better decisions)



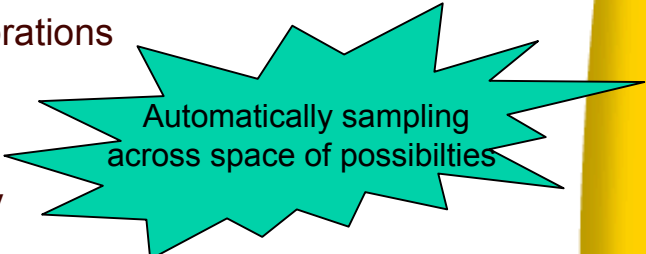
# How big is “too big” for a process model?

- The Goldilocks principle: limits to modeling
- This model is too small
  - Trite conclusions that are insensitive to most project details
- This model is too big
  - Cannot do anything with it unless it is calibrated
  - Estimate = **projectDetails** \* **modelCalibration**
- But COCOMO/COQUALMO/ THREAT is just right
  - Can use them for decision making, without calibration
  - Estimate = **projectDetails** \* **modelCalibration**



# Problems, and Solutions?

- ➔ “I need data. I want I want I want . We keep saying this and we don’t get it. So what do we do?”
  - Stop calibrating our models (ish)
  - Automatically sample across space of possible calibrations
- ➔ “Need more trade studies”
  - Automatically sample across space of possibilities
  - Days to define goals, seconds to run the trade study
- ➔ “Death to point estimates”
  - Report results from an automatic sample across a space of possibilities.
- ➔ “Cost is not enough”
  - Search space of possibilities for methods to improve a value function
- ➔ “Need more models of different types”
  - Generate skeletons of expert intuitions
  - Sample across space of possibilities within the space of possibilities.



Automatically sampling  
across space of possibilities





<http://unbox.org/wisp/tags/star>