Crystal Ball: Understanding Trends in Future Software

West Virginia University®

Tim Menzies (WVU)
Barry Bohem (USC)

8/1/08

Problem

- How to plan for the future?
- How are we to assess the benefits vs cost trade-offs of different software methods?
- How are we to make future plans for the agency, given some much change in current practices?

Approach

- Using traditional methods, there are no answers to these questions.

- The local tuning problem.
  - Software process models most accurate after local tuning
  - But, data required for local tuning is hard to obtain
    - Due to business sensitivity associated with the data
    - And differences in how the metrics are defined, collected and archived.

- New method
  - Find models that can make stable predictions
    - Despite unstable tunings

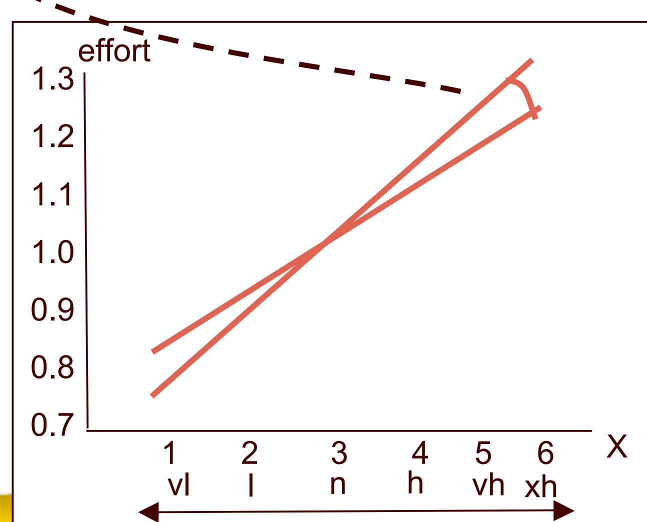
Pick your process models carefully

- There exists a class of models M of the form
  - Output= M(input,tunings)
- Such that
  - the variance in the tunings …
  - … is dominated by variance in the inputs
- For those models,
  - can make stable predictions
  - despite tuning varaince
- E.g. the USC family of
  - Effort predictors
  - Time predictors
  - Defect predictors

What to vary

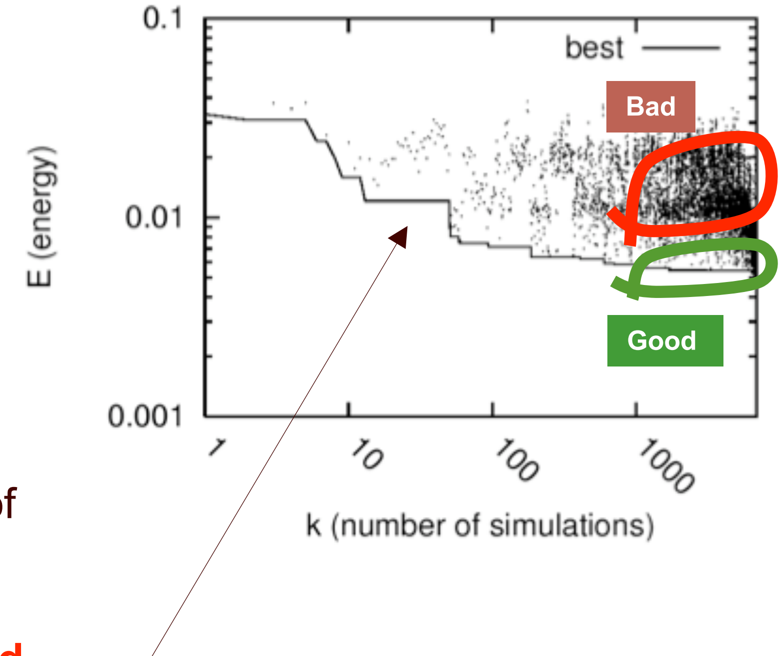
- E.g. effort = mx + b
- Two kinds of unknowns
  - Unknowns in project ranges
    - E.g. range of “x”
  - Unknowns in internal ranges
    - E.g. range of {“m”, “b”}
- Standard practice:
  - Use historical data to constrain {“m”,”b”}
- Here: Monte carlo over range of { “x” , “m”, “b” }
  - Learn values for “x” that reduce effort
  - As a side-effect, reduce variance
  - Not need for tuning data

	ranges			values	
project	feature	low	high	feature	setting
	rely	3	5	tool	2
Flight:	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		
	rely	1	4	tool	2
Ground:	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		



Approach (details)

- Implement USC software process models
  - COCOMO time / effort estimation,
  - COQUALMO defect prediction
  - MADACHY threats model
- Using
  - historical data, define space of past tunings
  - NASA experts, define standard project types
- Using simualted annealing, Monte Carlo simulation/optionation across intersection of
  - A particular project type
  - Space of possible tunings
- Rank options by frequency in good, not bad
- Test top ranked options for their median and variance effect. Smile if
  - Reduced median and variance in defects/ efforts/ time/ threats

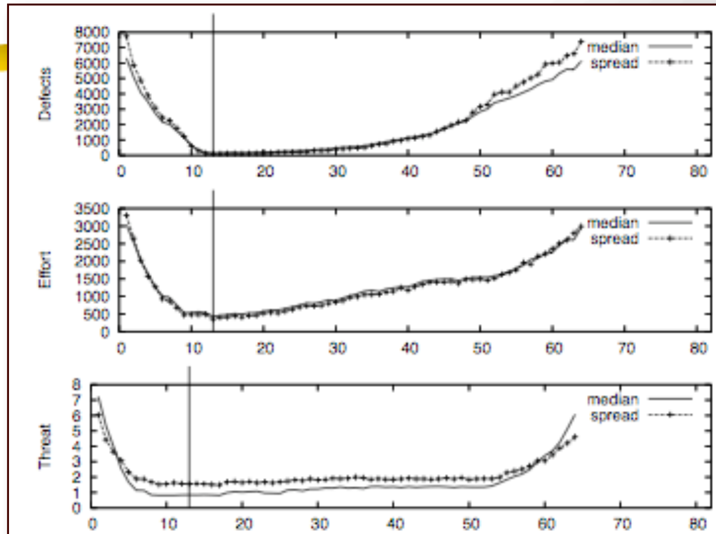


Sample run
(after 10,000 runs, little improvement)

Accomplishments

- After extensive interviews with…
  - SE research gurus
  - Experienced NASA developers/managers
- … clear evidence of variance in NASA software processes

- In numerous case studies…
- … massic reduction in
  - Defects/ effort/ time/ threats
  - Both median and variance
- …. options required to reach minimum defects/ effort /time /threats
  - Are a small subset of all options



X	feature = range	
1	pmat = 5	
2	acap = 5	
3	pcap = 5	
4	apex = 5	
5	ltex = 5	
6	prec = 4	
7	plex = 5	
8	site = 6	
9	peer reviews = 6	
10	execution testing and tools = 6	▲▲
11	automated analysis = 6	▲▲
12	pcon = 5	
13	pcap = 4	

Workshops (1 of 3): ICSE’08

Method 1 (at Workshop 1)

- Target application picked
  - A mission critical, real-time system;
  - Built by contractors (not in-house)
  - That has an operational life of 5 to 10 years (since have invested much effort into a mission critical system, an organization is most likely to use it for many years to come).
- For each COCOMO input variable
  - Boehm defines each variable
  - 5 minutes “open comments”
  - Vote. Record majoirty view

Results: Workshop 1 (part1)

- No Consensus :
  - data (database size per LOC)
  - ett (execution-based testing and tools)
  - reuse (level of design of the current product for future reuse)
  - time (% CPU requirements)
- No Change :
  - docu (required levels of documentation)
  - ltex (language and tools experience)
  - pcap (programmer capability)
  - plex (platform experience)
  - team (team cohesion)

- Increasing :
  - aa (automated analysis): i.e. use of more automated analysis;
  - acap (analyst capability): due to better SE education, broader international talent pool;
  - cplx (product complexity): i.e. due to more ambitious projects;
  - flex (development flexibility): due to more agile-style development;
  - peer (peer reviews).
  - pmat (process maturity)
  - pvol (platform volatility)
  - rely (reliability)
  - resl (architecture and risk analysis)
  - tool (use of software tools)

- Decreasing :
  - apex (analyst experience): due to a wider variety of new tools and an increasing number of novel application areas
  - pcon (personnel continuity): i.e. the panel expects more turn over in the industry1 .
  - sced (time to deliver product): i.e. more products expected to be delivered faster .
  - site (single site development): i.e. less work at one location, more development at multiple distributed locations;

Results: workshop1 (part2)

- Uncertainties on
  - prec (have we built this kind of thing before?)
  - stor (%CPU RAM)
  - time (%CPU cycles)
- 3 binary options (increase decrease)
  - 2*2*2 = 8 studies,
  - repeated 3 times for
  - KLOC= small, medium, or large

- In 8 studies, for small, medium, large, very stable conclusions:

small	medium	large	
8	8	8	acap=4,4.5
8	8	8	apex=4, 4.55.5
8	8	8	site=4.5
8	8	8	pmat=4.5,5
8	8	5	prec=4,5
8	8	6	pcon=2,3
1	2	3	peer=6
0	3	3	aa=5.5,6
0	2	3	ett = 5.5,6

Good news: stable conclusions

But is this result NASA relevant?

Workshops (2 of 3): JPL

- Acknowledge:
  - Much help from Dr. Jairus Hihn, JPL
- Preliminaries:
  - For two days, one-on-one interviews * 5
- Final day:
  - Morning session: everyone reviews results from preliminary sessions

Not a recommended method

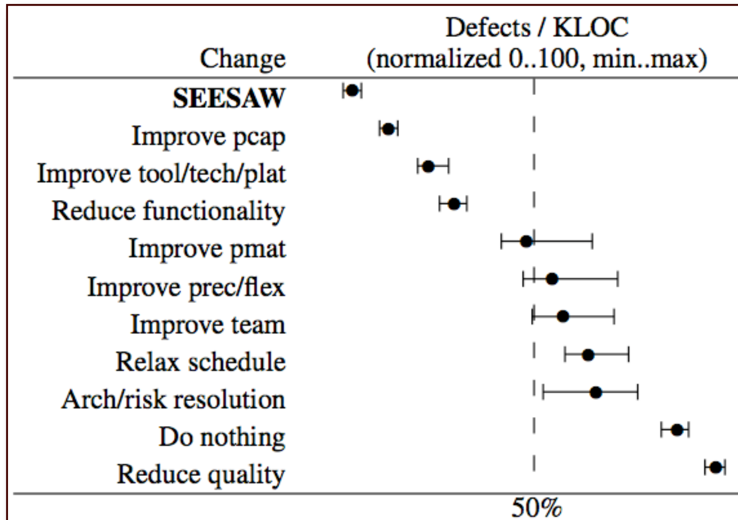
- Issue found with model
  - Need an extra feedback loop
- Little consensus on
  - Future of JPL
    - Development house?
    - Managers of external development
  - Future JPL missions
    - Much uncertainty there
- Also, ability to explicate local consensus
  - Requires elaborate domain knowledge

In future: recommend single focused panel session

Workshop (3 of 3)

- Where?
  - Headquarters?
  - JPL?
- Next time:
  - More focused panel
  - Better initial choice
  - Better range of future policies

Drastic change	Effects on Figure 4
1 Improve personnel	acap = 5; pcap = 5; pcon = 5 apex = 5 ; plex = 5 ; ltex = 5
2 Improve tools, techniques, or development platform	time = 3; stor = 3 pvol = 2; tool = 5 site = 6
3 Improve precedentness / development flexibility	prec = 5; flex = 5
4 Increase architectural analysis / risk resolution	resl = 5
5 Relax schedule	sced = 5
6 Improve process maturity	pmat = 5
7 Reduce functionality	data = 2; kloc * 0.5
8 Improve the team	team = 5
9 Reduce quality	rely = 1 ; docu = 1 time = 3 ; cplx = 1



Relevance to NASA

- NASA's software methods are rapidly evolving
  - NASA IV&V is the use of early lifecycle model-based validation.
  - Agile process,
  - Assertion-based analysis,
  - Eclipse-based programming,
  - Matlab-based automatic code generation,
  - Simulation-oriented development cycles,
  - etc.
- Any stability in all that chaos?
  - Can we make any plans for the future?

Next Steps

- Required: more NASA software gurus
  – Wanted: volunteers from SAS
- More simulation studies
  – To confirm / refute stability hypothesis
- Generation of recommendations
  – For different NASA project types