Managing Uncertainty in Value-based SE

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

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

- Come to PROMISE ‘09
- Value-based SE:
  - not even wrong?
- Data drought leading to conclusion uncertainty
  - Seek stability over samples
- On sampling some systems, we see
  - Value does not cost more
  - Value takes not take more time
  - Value (is, isn’t) harder to control
  - More value = more defects
- Community challenge:
  - when does 1,2,3,4 hold?
PROMISE ‘09

- www.promisedata.org/2009
- Reproducible SE results
- Papers:
  - and the data used to generate those papers
  - www.promisedata.org/data
- Keynote speaker:
  - Barry Boehm, USC
- Motto:
  - Repeatable, refutable, improvable
  - Put up or shut up
Value-based Software Engineering

The future of SE?
Q: what is SE
   - A: The application of science and mathematics by which the properties of software are made useful to people

Most SE techniques are “value-neutral”
   - Boehm, ASE 2004
   - Euphuism for “useless”?

Value-based SE makes a difference
   - Yeah? Really?
Risk Exposure (RE) = Software Quality Investment RE (REq) + Market Share Erosion RE (REm)

- Software Quality Investment RE
- Many defects and/or Critical defects
- Few rivals and/or Weak rivals
- Sweet Spot

- Market Share Erosion RE
- Many rivals and/or Strong rivals
- Few defects and/or minor defects
- Time to Ship (amount of testing)
The History of Computing Naturally Leads to Value-based SE
Value-based SE

Not even wrong?
Is the value-thesis not even wrong?

Wolfgang Pauli

The "conscience of physics",
- the critic to whom his colleagues were accountable.

Scathing in his dismissal of poor theories
- often labeling it *ganz falsch*, utterly false.

But "*ganz falsch*” was not his most severe criticism,
- He hated theories so unclearly presented as to be
  - untestable
  - unevaluatable
- Worse than wrong because they could not be proven wrong.
- Not properly belonging within the realm of science
  - even though posing as such.
- Famously, he wrote of a such unclear paper:
  - "That's not right. It's not even wrong."
So is the value thesis refutable?

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

Find a local value proposition
- A variant of USC Ph.D. thesis
  - [Huang 2006]: Software Quality Analysis: a Value-Based Approach
- “value”

Use them in a what-if scenario
Any difference in the conclusions?

(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)))
Aside

- Not really [Huang06]
  - But some variant Huang06
- Had to use some “engineering judgment”
  - a.k.a. guesses
- Apologies to Dr. Huang
Tools

Four USC models
- COCOMO effort prediction: staff months
- COCOMO schedule predictor: calendar months
- COQUALMO defect predictor: defects/KLOC
- THREATS: “how many dumb things are you doing right now?”

Monte Carlo simulator

AI search engine
- Search for the least number of project changes ...
- … that most improves the “target”
- “Target” is either
  - [Ase07]’s “energy” function
  - [Huang06]’s “value” proposition
“Energy” [Ase07]

- Euclidean Distance between best point and weighted normalized model scores (time, effort, defects, threats)
- Best point is origin because desired time, effort, detects, threats are to be as small as possible.
- Weights on scores can be tweaks to bias results. (i.e. BFC)
- Energy = \(\sqrt{\frac{a \cdot \text{square}(\text{normalized}(\text{Time})) + b \cdot \text{square}(\text{normalized}(\text{Effort})) + c \cdot \text{square}(\text{normalized}(\text{Defects})) + d \cdot \text{square}(\text{normalized}(\text{Threats}))}{\sqrt{a+b+c+d}}}\)
Value [Huang06]

- Value based evaluation method designed to minimize risk exposure based on 'rely'
- Balances beating everyone to market with more/worse bugs and being last to market with few/minor bugs.
- Based on NASA/USC Inspector SCRover project described in [Huang06]
Software Quality Investment Risk Exposure (REq) from [Huang06]

\[ \text{REq} = Pq(L) \times Sq(L) \]

- **Pq(L)**
  - [Huang06] calculated from COQUALMO estimate of delivered defect density
  - To incorporate COQUALMO model: defects/defects-with-vl-rely

- **Sq(L)**
  - [Huang06] used base values from a Pareto distribution and modified it with a coefficient based on a factor of 3 depending if the project was for early startup (1/3), commercial (1), high finance (3)
  - We used the same values for the distribution, but instead of defining 3 different functions, we use a function based on cplx to determine the coefficient \((3^{((cplx-3)/2)})\) (range is [0.333 … 5.196])
  - The values used in [Huang06] were normalized
    - We weighted them with PM.
Market Share Erosion Risk Exposure (REm)

- [Huang06]
  - used a simple exponential distribution for Rem
  - REm was normalized

- We weight it with PM
The details

Using AI to find stable conclusions in a space of options
Problem: local tuning

Problem

- Models need calibration
- Calibration needs data
- Usually, data incomplete (the “data drought”)

Our thesis:

- Precise tunings not required
- Space of possible tunings is well-defined
- Find and set the collars
  - Reveal policies that reduce effort/defects months
  - That are stable across the entire space
Run Delphi Sessions to Gather Project Ranges (e.g. ICSE 2008)

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 majority view
Sampling

- 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
Search for stable conclusions

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

Sample run
(after 10,000 runs, little improvement)
JPL flight systems (GNC)

<table>
<thead>
<tr>
<th>Project</th>
<th>Feature</th>
<th>Low</th>
<th>High</th>
<th>Feature</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flight</td>
<td>rely</td>
<td>2</td>
<td>5</td>
<td>tool</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>data</td>
<td>2</td>
<td>3</td>
<td>seed</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>cplx</td>
<td>3</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>stor</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>pvol</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>acap</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>apex</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>pcap</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>flex</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ltx</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>pmat</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ksloc</td>
<td>7</td>
<td>418</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ground</th>
<th>feature</th>
<th>range</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rely</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>data</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>cplx</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>stor</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>pvol</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>acap</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>apex</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>pcap</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>flex</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>ltx</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>pmat</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Ksloc</td>
<td>11</td>
<td>392</td>
</tr>
</tbody>
</table>
JPL ground systems (GNC)
Assessment criteria

- Minimal values found for:
  - Defects
  - Months
  - Effort

- Number of decisions required to find those minimums
  - In this case, 10 (ruse appears twice)
Results

And the winner is…
JPL Flight systems: Tactical

Policy: TACTICAL
Case Study: FLIGHT

- effort
- months
- value
- energy
- defects
- decisions
JPL Flight systems: Strategic

Policy: STRATEGIC
Case Study: FLIGHT

- **value**
- **energy**
- **effort**
- **months**
- **defects**
- **decisions**
JPL Ground systems: Tactical

Policy: TACTICAL
Case Study: GROUND

effort

value
energy
defects

months
decisions
JPL Ground systems: Strategic

Policy: STRATEGiC
Case Study: GROUND

- effort
- value
- energy
- months
- defects
- decisions
Patterns

Warm-up.

With value-based (compared to 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
Conclusion

So what?
Conclusion

- Is value-based SE “ganz falsch”? (not even wrong)
  - Hard to tell, if we have a data drought
  - So seek stability in samples of the possibilities
- On sample, using 2 target functions and 2 systems:
  - Value does not cost more
  - Value does not take more time
  - Value (is, isn’t) harder to control
  - More value = more defects
- Clearly, not true for all value propositions
  - But are there classes of systems with repeated patterns of value propositions?
  - For those “value patterns”:
    - Under what conditions do 1,2,3,4 apply
Sound bites

- Come to PROMISE ‘09
- Value-based SE:
  - not even wrong?
- Data drought leading to conclusion uncertainty
  - Seek stability over samples
- On sampling some systems, we see
  - Value does not cost more
  - Value takes not take more time
  - Value (is, isn’t) harder to control
  - More value = more defects
- Community challenge:
  - when does 1,2,3,4 hold?
PROMISE ‘09

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