NASA software: vision for the future

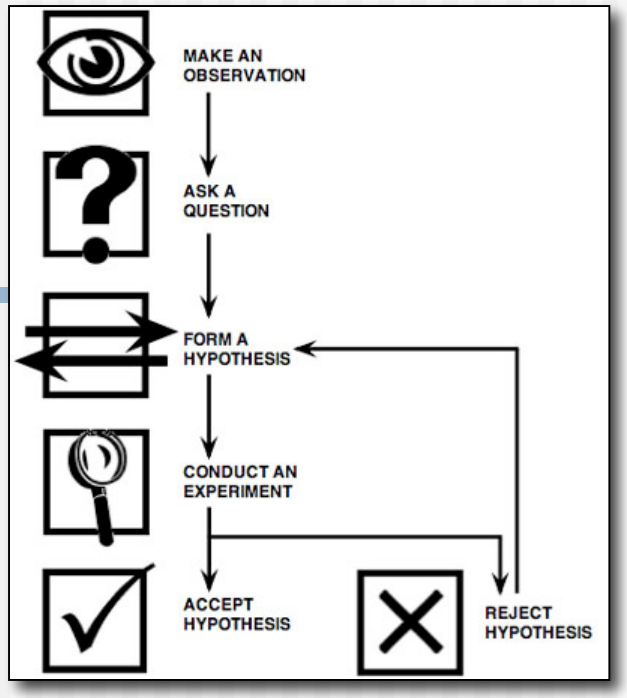
Tim Menzies
Barry Boehm
Jairus Hihn

July 21, 2008

Background

- Possibility
  - Maybe, can use AI to better plan software projects
    - Monte Carlo, simulated annealing, Bayesian feature selection
    - Result: better peeking into the future
  - E.g. Menzies, Boehm, Hihn, Lum ASE 2007
    - AI search methods on software process models
    - Some stable predictions in a huge space of options
    - http://menzies.us/pdf/07casease.pdf
- To test that possibility
  - Need constraints representing current & future NASA environment

Methodology



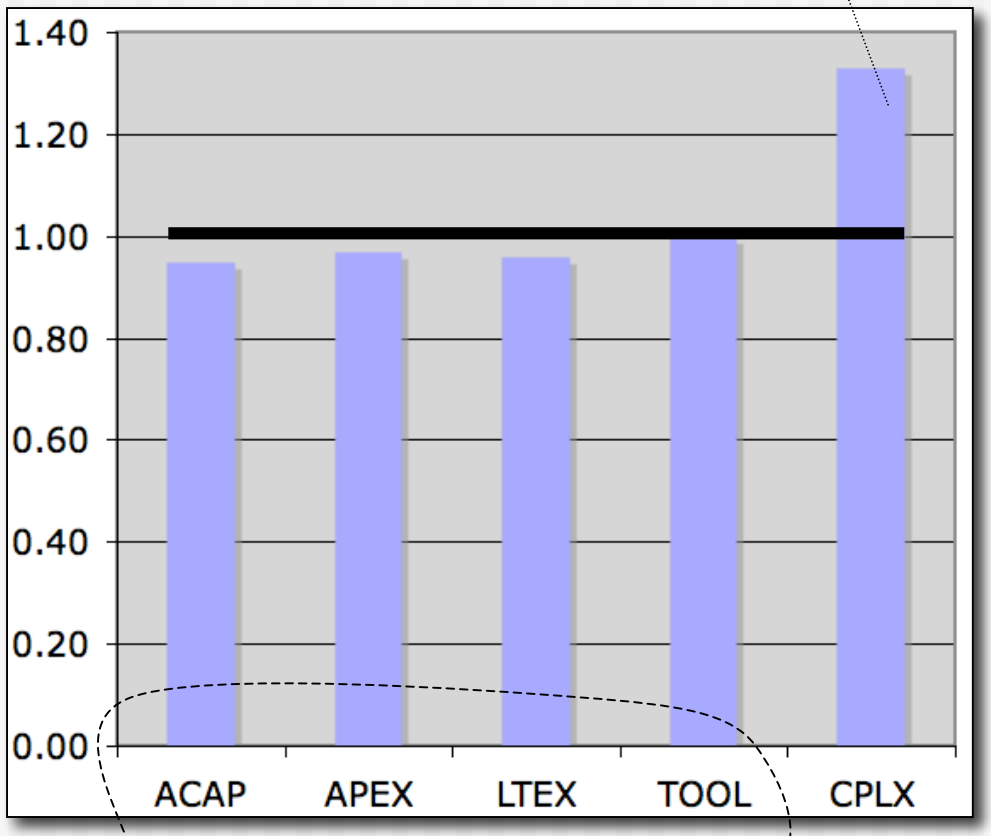
- 3 USC software process models
  - 1 WVU AI search engine
  - Multiple case studies
  - Sensitivty analysis
    - Found nine key factors
- Manual exploration of those factors with experienced software experts
  - May 2008: SE resaerch leaders
  - July 2008: JPL experts (that's you)
  - Nov 2008: Experts at other NASA centers

Focus

- What:
  - JPL flight software
  - Planetary missions (e.g. rovers)
- When:
  - -10 years (to calibrate historical data)
  - +10 years (to make predictions)
- What:
  - Projects expressed in the COCOMO ontology
- Why:
  - Please anwser :
    - A: significantly better
    - B: somewhat better
    - C: no change
    - D : somewhat worse
    - E : significantly worse

Factors (1)

- Acap (analyst capability)
  - Worse: worst 15%
  - Middle: 55%
  - Best: best 10%
- Apex (applications experience)
  - Worst: 2 months
  - Middle: 1 year
  - Best: 6 years
- Ltex (language and toolset experience)
  - Worst: 2 months
  - Middle: 1 year
  - Best: 6 years
- Tool (use of software tools)
  - Worst: edit,code,debug
  - Best: integrated with life cycle
- Cplx (product complexity)
  - Worse: e.g. simple read/write statements
  - Middle: e.g. use of simple interface widgets
  - Best: e.g. performance-critical em- bedded systems



Factors (2)

- Resl (architecture or risk resolution)
  - Worst: few interfaces defined or few risk eliminated
  - Middle: most interfaces defined or most risks eliminated
  - Best: all interfaces defined or all risks eliminated
- Pmat process maturity
  - Worst: CMM level 1
  - Middle: CMM level 3
  - Best: CMM level 5
- Prec precedentedness
  - Worst: we have never built this kind of software before
  - Middle: somewhat new
  - Best: thoroughly familiar
- Flex (development flexibility)
  - Worst: development process rigorously defined
  - Middle; some guidelines, which can be relaxed
  - Best; only general goals defined