Incorporating Bayesian Analysis to Improve the Accuracy of COCOMO II and its Quality Model Extension

Sunita Devnani-Chulani
USC-CSE

Feb 19, 1998
Outline

➤ Motivation

◆ Related Work
  – Existing Software Estimation Techniques

◆ Research Approach
  – Modeling Methodology
  – Bayesian Analysis
  – Quality Model

◆ Status
  – COCOMO II Bayesian Prototype
  – Quality Model Data Analysis
  – Contributions

◆ Plans
Quantitative Software Estimation

Examples of Popular Software Estimation Models:
❖ SPR’s Checkpoint (proprietary)
❖ Putnam’s SLIM (semi-published)
❖ COCOMO (published) - no quality estimation
COnstructive COSt MOdel (COCOMO)

- COCOMO published since 1981
- Commercial implementations of COCOMO
  - CoCoPro, CB COCOMO, COCOMOID, COSTMODL, GECOMO Plus, SECOMO, etc.
- Other models based on COCOMO
  - REVIC, Gulezian
- COCOMO II
  - Research effort started in 1994 to develop a 1990’s-2000’s software cost model
  - Address new processes and practices
  - COCOMO II.199Y/200Y
COCOMO ‘81 & COCOMO II.1997

- COCOMO ‘81: Pred (.20) = 68%
- COCOMO II.1997: Pred (.20) = 46%
  - Challenges faced in calibrating COCOMO II
    - GUI builders, COTS, 4GL’s, reuse, requirements breakage
      - Need to rethink size metrics
    - Distributed interactive applications
      - Web-based, object-oriented, event-based
      - Middleware effects
    - New process models (evolutionary, incremental, spiral)
      - Phases overlap
      - Where are cost measurement endpoints?
    - Lack of good data
      - not enough data (i.e. very little degrees of freedom)
      - lack of dispersion
      - heteroskedasticity
COCOMO II.1997 Calibration

- 83 projects
- Multiple Linear Regression
  - 10% weighted average between a-priori values and data-determined values

Develop for Reuse (RUSE)
Multiple Linear Regression Well-Suited When...

- lot of data is available
- no data items are missing
- there are no outliers
- the predictor variables are not highly correlated
- the predictor variables have an easy interpretation when used in model

most are violated by current software engineering data
Model Plans: Affiliate Priorities
October ‘97

21  ◆ Improve accuracy of COCOMO II Model
15  ◆ Cost/schedule/quality tradeoffs
12  ◆ Sizing improvements
10  ◆ COTS integration costs
  7  ◆ Activity Distribution
  3  ◆ Life cycle tradeoff models
Solutions to Top Two Priorities

- **Improve Accuracy of COCOMO II**
  - Try Bayesian technique
    - with different weighted average between data-determined and a-priori values for each parameter
  - Collect more data

Does Bayesian analysis improve the accuracy of the COCOMO model versus multiple regression analysis?

- **Cost/schedule/quality tradeoffs**
  - Develop Quality Model and integrate with existing COCOMO II
    - Use COCOMO II parameters
    - Use similar Bayesian approach
Outline

◆ Motivation

➡ Related Work
  ➔ Existing Software Estimation Techniques

◆ Research Approach
  ➔ Modeling Methodology
  ➔ Bayesian Analysis
  ➔ Quality Model

◆ Status
  ➔ COCOMO II Bayesian Prototype
  ➔ Quality Model Data Analysis
  ➔ Contributions

◆ Plans
Existing Software Estimation Techniques - 6 Categories

- Model-Based - SLIM, COCOMO
- Learning-Oriented - Neural, Case-based
- Expertise-Based - Delphi, Rule-Based
- Dynamics-Based - Abdel-Hamid-Madnick
- Regression-Based - OLS, Robust
- Composite - Bayesian
Outline

◆ Motivation

◆ Related Work
  – Existing Software Estimation Techniques

➤ Research Approach
  – Modeling Methodology
  – Bayesian Analysis
  – Quality Model

◆ Status
  – COCOMO II Bayesian Prototype
  – Quality Model Data Analysis
  – Contributions

◆ Plans
The Seven-Step Modeling Methodology

1. Analyze Existing literature
2. Perform Behavioral Analysis
3. Identify Relative Significance
4. Perform Expert-Judgment, Delphi Assessment
5. Gather Project Data
6. Determine Bayesian A-Posteriori Update
7. Gather more data; refine model

A-PRIORI MODEL + SAMPLING DATA = A-POSTERIORI MODEL
The Bayesian Approach

- **Simple Software Cost Model**

\[
Effort = A \cdot Size^B \cdot \varepsilon
\]

- **Linearizing, we get**

\[
\ln(Effort) = A_1 + B \cdot \ln(Size) + \ln(\varepsilon)
\]

where \(A_1 = \ln(A)\)

- **Sampling Data**

<table>
<thead>
<tr>
<th>Effort (PM)</th>
<th>Size (SLOC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>4500</td>
</tr>
<tr>
<td>7</td>
<td>5200</td>
</tr>
<tr>
<td>4.8</td>
<td>3200</td>
</tr>
<tr>
<td>9</td>
<td>7000</td>
</tr>
<tr>
<td>8</td>
<td>6000</td>
</tr>
</tbody>
</table>

PM=Person Months; SLOC=Source L

PM=Person Months; SLOC=Source L
No Prior Information - Complete Prior Uncertainty

\[ f(A_1) = 1 \quad -\infty < A_1 < +\infty \]

\[ f(B) = 1 \quad -\infty < B < +\infty \]
## Modeling Under Complete Prior Uncertainty - Statistical Analysis

<table>
<thead>
<tr>
<th>Label</th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>0.330574</td>
<td>0.189132</td>
<td>1.748</td>
</tr>
<tr>
<td>log[SIZE]</td>
<td>0.987199</td>
<td>0.115833</td>
<td>8.523</td>
</tr>
</tbody>
</table>

R Squared: 0.960336

\[
\ln(\text{Effort}) = 0.33 + 0.99 \ln(\text{Size})
\]

\[
\text{Effort} = 1.4 \cdot \text{Size}^{0.99}
\]

where \(0.39 < B < 1.58\)

\(0.95 < A < 2.03\)

\[1.4 = e^{0.33}\]
Posterior Density Functions - Noninformative Prior

\[ f \left( \frac{A_1}{\ln(\text{Effort})} \right) \]

\[ f \left( \frac{B}{\ln(\text{Effort})} \right) \]

But, can \( B < 1 \); i.e. Economies of Scale?
Some Prior Knowledge on B

- Experience indicates that Software exhibits diseconomies of scale

[Banker94, Gulledge93]*.

- We had some prior knowledge, since we are dissatisfied with results shown on previous two slides
- \( f(B) = 1 \) if \( B > 1 \)
- \( f(B) = 0 \) if \( B = 1 \) or \( B < 1 \)


Post Sample Density Functions - Inclusion of Prior Information

51% of area under curve

$f_N(B/\ln(Effort))$

$f_{TN}(B/\ln(Effort))$

Mean$_{TN}(B) = 1.15$
The Bayesian Approach

A - Priori Information + Sampling Data = A - Posteriori Model

- Expertise-based
- Model-based
- Learning-oriented
- Dynamics-based
- Regression-based
Determining A-Priori Information: Modeling Methodology

1. Analyze Existing literature
2. Perform Behavioral Analysis
3. Identify Relative Significance
4. Perform Expert-Judgment, Delphi Assessment
Literature, Behavioral Analysis (Steps 1-3)

Productivity Range = Highest Rating / Lowest Rating

Literature, behavioral analysis
Results of Delphi (Step 4)

Productivity Range = \frac{\text{Highest Rating}}{\text{Lowest Rating}}

A-priori
Experts’ Delphi

1.42

1.54

Literature,
behavioral analysis
Determining A-Posteriori Information: Modeling Methodology

1. Gather Project Data
2. Determine Bayesian A-Posteriori Update
Results of Sampling Data (Step 5)

Noisy data analysis

1.28

A-priori Experts’ Delphi

1.42

Literature, behavioral analysis

1.54

Productivity Range = Highest Rating / Lowest Rating
Results of Bayesian Update: Using Prior and Sampling Information (Step 6)

A-posteriori Bayesian update

A-priori

Experts’ Delphi

Noisy data analysis

Productivity Range = Highest Rating / Lowest Rating

Literature, behavioral analysis
COCOMO II Calibration Approaches

COCOMO II.1997
10% weighted-average approach

Bayesian approach - weight determined by data and prior significance
Quality Model

Defect Introduction pipes

Code Defects 15/KDSI

Design Defects 25/KDSI

Requirements Defects 5/KDSI*

Defect Removal pipes

Residual Software Defects

*Baseline rates of the late 1970s based on studies done by Jones, Thayer et al and Boehm
### An Example Defect Introduction Rate Driver

<table>
<thead>
<tr>
<th>AEXP (Applications Experience) level</th>
<th>Requirements</th>
<th>Design</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very High</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fewer Requirements defects due to less learning and fewer false starts</td>
<td>Fewer Design defects due to less learning and fewer false starts</td>
<td>Fewer Coding defects due to less learning</td>
<td>0.81</td>
</tr>
<tr>
<td>Fewer Requirements understanding defects</td>
<td>Fewer Requirements traceability defects</td>
<td>Fewer Coding defects due to requirements, design shortfalls</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fewer defects introduced in fixing requirements, preliminary design fixes</td>
<td></td>
<td>0.82</td>
</tr>
<tr>
<td>Nominal</td>
<td>Nominal level of defect introduction</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td><strong>Very Low</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>More Requirements defects due to extensive learning and more false starts</td>
<td>More Design defects due to less learning and fewer false starts</td>
<td>More Coding defects due to extensive learning</td>
<td>1.24</td>
</tr>
<tr>
<td>More Requirements understanding defects</td>
<td>More Requirements traceability defects</td>
<td>More Coding defects due to requirements, design shortfalls</td>
<td></td>
</tr>
<tr>
<td></td>
<td>More defects introduced in fixing requirements, preliminary design fixes</td>
<td></td>
<td>1.22</td>
</tr>
<tr>
<td>Behavioural analysis; relative significance</td>
<td>1.56</td>
<td>1.56</td>
<td>1.32</td>
</tr>
<tr>
<td>Expert-judgment</td>
<td>Range</td>
<td>1.4-1.65</td>
<td>1.3-1.56</td>
</tr>
<tr>
<td>Delphi Round 1</td>
<td>Median</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>Expert-judgment</td>
<td>Range</td>
<td>1.5-1.6</td>
<td>1.4-1.56</td>
</tr>
<tr>
<td>Delphi Round 2</td>
<td>Median</td>
<td>1.53</td>
<td>1.5</td>
</tr>
</tbody>
</table>
# An Example Defect Removal Rate Driver

<table>
<thead>
<tr>
<th>RELY (Required Software Reliability) level</th>
<th>Impact on Defect Removal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High</td>
<td>More Requirements defects removed due to more consistency, detailed verification, QA, CM, standards, CDR, documentation, IV&amp;V interface, test plans, procedures</td>
</tr>
<tr>
<td>Nominal</td>
<td>Nominal level of defect removal</td>
</tr>
<tr>
<td>Very Low</td>
<td>Fewer Requirements defects removed due to lower consistency, minimal verification, QA, CM, standards, SSR, documentation, IV&amp;V interface, test plans, procedures</td>
</tr>
<tr>
<td>Quality Range</td>
<td>10-90%</td>
</tr>
</tbody>
</table>
Outline

◆ Motivation

◆ Related Work
  – Existing Software Estimation Techniques

◆ Research Approach
  – Modeling Methodology
  – Bayesian Analysis
  – Quality Model

► Status
  – COCOMO II Bayesian Prototype
  – Quality Model Data Analysis
  – Contributions

◆ Plans
Bayesian Prototype Model

\[ \text{Effort} = A \cdot (\text{Size})^B \cdot RCPX \cdot PDIF \cdot PERS \]

Linearizing, we get,

\[ \ln(\text{Effort}) = \ln(A) + B \ln(\text{Size}) + \ln(RCPX) + \ln(PDIF) + \ln(PERS) \]

<table>
<thead>
<tr>
<th>Prototype Parameter</th>
<th>Combination of*:</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCPX (Product Reliability and Complexity)</td>
<td>RELY, DATA, CPLX, DOCU</td>
</tr>
<tr>
<td>PDIF (Platform Difficulty)</td>
<td>TIME, STOR, PVOL</td>
</tr>
<tr>
<td>PERS (Personnel Capability)</td>
<td>ACAP, PCAP, PCON</td>
</tr>
</tbody>
</table>

*See Glossary - last slide
### Bayesian Prototype A-Priori Information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>VL</th>
<th>L</th>
<th>N</th>
<th>H</th>
<th>VH</th>
<th>XH</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREC</td>
<td>0.05</td>
<td>0.04</td>
<td>0.03</td>
<td>0.02</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>FLEX</td>
<td>0.05</td>
<td>0.04</td>
<td>0.03</td>
<td>0.02</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>RESL</td>
<td>0.05</td>
<td>0.04</td>
<td>0.03</td>
<td>0.02</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>TEAM</td>
<td>0.05</td>
<td>0.04</td>
<td>0.03</td>
<td>0.02</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>PMAT</td>
<td>0.05</td>
<td>0.04</td>
<td>0.03</td>
<td>0.02</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>RELY</td>
<td>0.75</td>
<td>0.88</td>
<td>1</td>
<td>1.15</td>
<td>1.4</td>
<td>0.00</td>
</tr>
<tr>
<td>DATA</td>
<td>0.94</td>
<td>1</td>
<td>1.08</td>
<td>1.16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPLX</td>
<td>0.75</td>
<td>0.88</td>
<td>1</td>
<td>1.15</td>
<td>1.3</td>
<td>1.65</td>
</tr>
<tr>
<td>DOCU</td>
<td>0.85</td>
<td>0.93</td>
<td>1</td>
<td>1.08</td>
<td>1.17</td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td></td>
<td></td>
<td>1</td>
<td>1.11</td>
<td>1.3</td>
<td>1.66</td>
</tr>
<tr>
<td>STOR</td>
<td></td>
<td></td>
<td>1</td>
<td>1.06</td>
<td>1.21</td>
<td>1.56</td>
</tr>
<tr>
<td>PVOL</td>
<td>0.87</td>
<td>1</td>
<td>1.15</td>
<td>1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACAP</td>
<td>1.5</td>
<td>1.22</td>
<td>1</td>
<td>0.83</td>
<td>0.67</td>
<td></td>
</tr>
<tr>
<td>PCAP</td>
<td>1.37</td>
<td>1.16</td>
<td>1</td>
<td>0.87</td>
<td>0.74</td>
<td></td>
</tr>
<tr>
<td>PCON</td>
<td>1.26</td>
<td>1.11</td>
<td>1</td>
<td>0.91</td>
<td>0.83</td>
<td></td>
</tr>
</tbody>
</table>

- New Prior Information being gathered - Delphi process is ongoing.
- Hence, for the prototype model, the above point estimates were used and the variance for each parameter was assumed
## Bayesian Prototype Sampling Data

**Subset of Data (112 observations)**

<table>
<thead>
<tr>
<th>EFFORT</th>
<th>SIZE</th>
<th>B</th>
<th>PERS</th>
<th>RCPX</th>
<th>PDIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>520.0</td>
<td>123.</td>
<td>1.110</td>
<td>0.66</td>
<td>1.66</td>
<td>0.92</td>
</tr>
<tr>
<td>169.0</td>
<td>36.</td>
<td>1.120</td>
<td>0.66</td>
<td>1.44</td>
<td>0.87</td>
</tr>
<tr>
<td>544.0</td>
<td>282.</td>
<td>1.170</td>
<td>0.94</td>
<td>1.24</td>
<td>1.2</td>
</tr>
<tr>
<td>198.0</td>
<td>179.</td>
<td>1.130</td>
<td>0.92</td>
<td>1.33</td>
<td>1.2</td>
</tr>
<tr>
<td>75.0</td>
<td>35.6</td>
<td>1.110</td>
<td>0.69</td>
<td>1.43</td>
<td>0.87</td>
</tr>
<tr>
<td>418.0</td>
<td>118.</td>
<td>1.090</td>
<td>0.53</td>
<td>1.52</td>
<td>0.94</td>
</tr>
<tr>
<td>631.0</td>
<td>142.8</td>
<td>1.170</td>
<td>0.95</td>
<td>1.27</td>
<td>0.87</td>
</tr>
<tr>
<td>418.0</td>
<td>140.8</td>
<td>1.130</td>
<td>0.87</td>
<td>1.0</td>
<td>0.87</td>
</tr>
<tr>
<td>499.0</td>
<td>127.05</td>
<td>1.160</td>
<td>0.9</td>
<td>1.19</td>
<td>1.15</td>
</tr>
<tr>
<td>128.0</td>
<td>28.11</td>
<td>1.180</td>
<td>0.51</td>
<td>1.02</td>
<td>1.0</td>
</tr>
<tr>
<td>58.0</td>
<td>10.23</td>
<td>1.180</td>
<td>0.9</td>
<td>1.12</td>
<td>1.08</td>
</tr>
<tr>
<td>131.0</td>
<td>71.</td>
<td>1.110</td>
<td>0.53</td>
<td>0.94</td>
<td>0.87</td>
</tr>
</tbody>
</table>
Bayesian Prototype Results

### Calibration Dataset

**Point Estimates and Variances**

<table>
<thead>
<tr>
<th>A B PERS RCPX PDIF</th>
<th>Prior</th>
<th>Sample</th>
<th>Prior+Sample = Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Point Estimates</strong></td>
<td>(2.45 1 1 1 1)</td>
<td>(2.64 0.95 1.06 1.07 0.52)</td>
<td>(2.64 0.95 1.02 1.01 0.82)</td>
</tr>
<tr>
<td><strong>Variance</strong></td>
<td>(0.025 0.025 0.025 0.09 0.04)</td>
<td>(0.023 0.001 0.04 0.04 0.08)</td>
<td>(0.011 0.0007 0.015 0.026 0.026)</td>
</tr>
</tbody>
</table>

### Validation Dataset

**Improvement in Prediction Accuracies using the Bayesian technique**

<table>
<thead>
<tr>
<th>Prediction Accuracy of Prototype Model</th>
<th>Prior</th>
<th>Sample</th>
<th>Prior+Sample = Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRED(.20)</td>
<td>13/60 = 22%</td>
<td>17/60 = 28%</td>
<td>17/60 = 28%</td>
</tr>
<tr>
<td>PRED(.30)</td>
<td>20/60 = 33%</td>
<td>23/60 = 38%</td>
<td>25/60 = 42%</td>
</tr>
</tbody>
</table>
# Defect Data Reporting Scheme

<table>
<thead>
<tr>
<th>Type of Artifact</th>
<th>Requirements</th>
<th>Design</th>
<th>Code &amp; Unit Test</th>
<th>SW Integ. and Test</th>
<th>SW Acceptance Test</th>
<th>System Implementation and Test</th>
<th>Post-Operational</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reqts</td>
<td>50/30</td>
<td>20+20/</td>
<td>10+20/</td>
<td>2+15/10</td>
<td>1+7/6/4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>/2</td>
<td>20/.5</td>
<td>15/1.0</td>
<td>/1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td>55/25</td>
<td>15+30/</td>
<td>25/2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The values represent the Introduced + Unresolved / Resolved in Activity/Cost To Resolve by Activity.
## Results of initial-project data analysis for Quality Model

<table>
<thead>
<tr>
<th>Type of Artifact</th>
<th>1970’s Baseline DIRs</th>
<th>Quality Adjustment Factor</th>
<th>Predicted DIR</th>
<th>Actual DIR</th>
<th>Calibrated Constant (A)</th>
<th>1990’s Baseline DIRs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>5</td>
<td>0.5</td>
<td>2.5</td>
<td>4.5</td>
<td>1.8</td>
<td>9</td>
</tr>
<tr>
<td>Design</td>
<td>25</td>
<td>0.44</td>
<td>11</td>
<td>8.4</td>
<td>0.77</td>
<td>19</td>
</tr>
<tr>
<td>Code</td>
<td>15</td>
<td>0.5</td>
<td>7.5</td>
<td>16.6</td>
<td>2.21</td>
<td>33</td>
</tr>
</tbody>
</table>

DIR = Defect Introduction Rate
Contributions

- Developed a 7-step modeling methodology
  - can be used to develop software estimation models in the lack of abundance of complete data

- Developed a Bayesian prototype in an attempt to improve the accuracy of COCOMO II
  - a formal process of how to incorporate experience-based prior information to sampling data was demonstrated
  - explicitly demonstrated what prior density functions were used
    - others can repeat calibration process
    - other prior density functions can be easily used
Contributions (contd.)

◆ Developed a Quality model extension to COCOMO II to facilitate cost/schedule/quality tradeoffs
  ❖ provides insights on determining ship-time
  ❖ enables “what-if” analyses to demonstrate the effects of personnel, project, product, platform characteristics on software quality
Outline

◆ Motivation

◆ Related Work
  – Existing Software Estimation Techniques

◆ Research Approach
  – Modeling Methodology
  – Bayesian Analysis
  – Quality Model

◆ Status
  – COCOMO II Bayesian Prototype
  – Quality Model Data Analysis
  – Contributions

➡ Plans
Plans

◆ COCOMO

❖ Collect more data (ongoing process).
  – Should grow to ~200 datapoints by Oct ‘98

❖ Complete the Delphi process (by Mar ‘98).
  – will determine a more defensible set of a-priori model parameters
  – Round 1 completed

❖ Perform Bayesian analysis on the complete COCOMO II model (by Oct ‘98 - by the 13th International Forum on COCOMO).
Plans (contd.)

◆ Quality Model

❖ Formulate the complete Defect Removal model (by Mar ‘98).

❖ Collect more data (ongoing process).
  – Not as much as the COCOMO II database
  – ~15 datapoints

❖ Use the same bayesian approach used for COCOMO II to calibrate the model (by Nov ‘98).
  – Bayesian approach well-suited for updating model with new data as and when available

❖ Integrate the quality model to the COCOMO II model (by Nov ‘98).

◆ Defend my research (in Spr ‘99).
## Glossary

<table>
<thead>
<tr>
<th>Abrev.</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREC</td>
<td>Precendentedness</td>
</tr>
<tr>
<td>FLEX</td>
<td>Development Flexibility</td>
</tr>
<tr>
<td>RESL</td>
<td>Architecture and Risk Resolution</td>
</tr>
<tr>
<td>TEAM</td>
<td>Team cohesion</td>
</tr>
<tr>
<td>PMAT</td>
<td>Process Maturity</td>
</tr>
<tr>
<td>RELY</td>
<td>Required Software Reliability</td>
</tr>
<tr>
<td>DATA</td>
<td>Data Base Size</td>
</tr>
<tr>
<td>CPLX</td>
<td>Product Complexity</td>
</tr>
<tr>
<td>RUSE</td>
<td>Develop for Reuse</td>
</tr>
<tr>
<td>DOCU</td>
<td>Documentation Match to Life-cycle Needs</td>
</tr>
<tr>
<td>TIME</td>
<td>Time Constraint</td>
</tr>
<tr>
<td>STOR</td>
<td>Storage Constraint</td>
</tr>
<tr>
<td>PVOL</td>
<td>Platform Volatility</td>
</tr>
<tr>
<td>ACAP</td>
<td>Analyst Capability</td>
</tr>
<tr>
<td>PCAP</td>
<td>Programmer Capability</td>
</tr>
<tr>
<td>AEXP</td>
<td>Applications Experience</td>
</tr>
<tr>
<td>PEXP</td>
<td>Platform Experience</td>
</tr>
<tr>
<td>LTEX</td>
<td>Language and Tool Experience</td>
</tr>
<tr>
<td>PCON</td>
<td>Personnel Continuity</td>
</tr>
<tr>
<td>TOOL</td>
<td>Use of Software Tools</td>
</tr>
<tr>
<td>SITE</td>
<td>Multi-Site Development</td>
</tr>
<tr>
<td>SCED</td>
<td>Required Development Schedule</td>
</tr>
</tbody>
</table>

### Rating Scale

- **VL** = Very Low
- **L** = Low
- **N** = Nominal
- **H** = High
- **VH** = Very High
- **XH** = Extra High