A Simulation Method for Defectivity Profiling

Dan Houston, Ph.D.
The Aerospace Corporation

Computers and Software Division
April 23, 2009
Complementarity in Software Quality Management

- **Product qualities**
  - *Typically specific to a product or product line*
  - *Wide variety of metrics, for example, data display response time or usability*
  - *Analysis: regression, time series, designed experiments, simulation, etc.*
  - *Modeling supports proactive view of quality*

- **Defectivity**
  - *Generic approach and most common*
  - *Based on counts, categorization, and profiling over time*
  - *Analysis: defect profile targets, reliability growth, defect classification, leakage matrix*
  - *Most common derived metrics are defect density, defect discovery rate*
COQUALMO

- Extension of COCOMO II
  - Relates defectivity to cost and schedule
  - COCOMO II drivers are treated as quality drivers
  - Quality measured in counts of non-trivial defects (critical system function impairment or worse)

- Submodels
  - Defect introduction
  - Defect removal

© The Aerospace Corporation 2009

COCOMO II and COQUALMO were developed at the Center for Systems and Software Engineering of the University of Southern California.
Defect Introduction Submodel

• Sources of defects: Requirements, Design, and Code

\[ DI_{source} = DIR_{source,nom} \times Size^{B_{source}} \times \prod_{i=1}^{21} DefectDriver_{i,source} \]

• \( DI \) = defects introduced from each source
• \( DIR_{nom} \) = nominal defect introduction rate by source
• \( Size^{B} \) = software size raised to scale factor by source

• Defect Drivers in Quality Adjustment Factors (QAFs)
  – Example: Analyst Capability (ACAP)

• Defect driver values produced through a two-round Delphi process.

<table>
<thead>
<tr>
<th>ACAP Level</th>
<th>Requirements</th>
<th>Design</th>
<th>Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High</td>
<td>.75</td>
<td>.83</td>
<td>.90</td>
</tr>
<tr>
<td>High</td>
<td>.87</td>
<td>.91</td>
<td>.95</td>
</tr>
<tr>
<td>Nominal</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Low</td>
<td>1.15</td>
<td>1.10</td>
<td>1.05</td>
</tr>
<tr>
<td>Very Low</td>
<td>1.33</td>
<td>1.22</td>
<td>1.11</td>
</tr>
</tbody>
</table>
Defect Removal Submodel

- Defect removal activities: peer reviews, automated analysis, testing

\[ DR_{artifact} = DI_{artifact} \times \prod_{i=1}^{3} (1 - DRF_{i,artifact}) \]

- DR = defects removed from artifact
- DI = defects introduced into each artifact
- DRF = removal fraction for each activity, \( i \), applied to each artifact
- DRF assigned to quality levels of activities in 2-round Delphi
## Defect Removal Ratings

<table>
<thead>
<tr>
<th>Rating</th>
<th>Peer Reviews</th>
<th>Automated Analysis</th>
<th>Execution Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td>None</td>
<td>Simple compiler checking</td>
<td>None</td>
</tr>
<tr>
<td>Low</td>
<td>Ad hoc</td>
<td>Static module code analysis</td>
<td>Ad hoc</td>
</tr>
<tr>
<td>Nominal</td>
<td>Informal roles and procedures</td>
<td>Static code analysis; Requirements/design checking</td>
<td>Basic test process</td>
</tr>
<tr>
<td>High</td>
<td>Formal roles and procedures</td>
<td>Intermediate semantic analysis; Requirements/design checking</td>
<td>Organizational test process; Basic test coverage tools</td>
</tr>
<tr>
<td>Very High</td>
<td>Formality plus use of data</td>
<td>Temporal analysis &amp; symbolic execution</td>
<td>Advanced test tools; Quantitative test process</td>
</tr>
<tr>
<td>Extra High</td>
<td>Review process improvement</td>
<td>Formal specification and verification</td>
<td>Highly advanced tools; Model-based test management</td>
</tr>
</tbody>
</table>
Defectivity Profiling over Time

Fit distributions to defect discovery data

- In system test, reliability growth curves are used to estimate latent defects, support test decisions, support readiness for release decisions
- In earlier stages
  - Reduce cost of software quality
  - Challenge: obtaining defect data
  - Answer: software inspections provide data for defectivity profiling
  - Example: defect leakage matrix

<table>
<thead>
<tr>
<th>Defect Injection Phase</th>
<th>Requirements</th>
<th>Design</th>
<th>Code &amp; Unit Test</th>
<th>SW Integration Test</th>
<th>System Test</th>
<th>Operational Test</th>
<th>Post-Release</th>
<th>Uncorrected</th>
<th>Leakage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>17</td>
<td>11</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>69%</td>
</tr>
<tr>
<td>Design</td>
<td>71</td>
<td>56</td>
<td>21</td>
<td>11</td>
<td>11</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>58%</td>
</tr>
<tr>
<td>Code &amp; Unit Test</td>
<td>201</td>
<td>87</td>
<td>67</td>
<td>5</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>45%</td>
<td></td>
</tr>
<tr>
<td>SW Integration Test</td>
<td>11</td>
<td>11</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>54%</td>
</tr>
<tr>
<td>System Test</td>
<td>14</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td>Operational Test</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>Post Release</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>Total New Defects</td>
<td>17</td>
<td>82</td>
<td>264</td>
<td>124</td>
<td>108</td>
<td>25</td>
<td>20</td>
<td>5</td>
<td>50%</td>
</tr>
</tbody>
</table>
Purpose of this Effort

• Advance defectivity profiling
  – Utilize the quantitative relationships developed and refined in COCOMO II and COQUALMO.
  – Fit a non-parametric curve composed of multiple curves to defect data
Model Description: Decompositions

- To accommodate significant project changes, provide for changes in defect driver and DRF values by project interval.
- To accommodate variation in quality of practice, use a profile (set of weighted values) for DRFs.
- To accommodate artifact types, use separate DRF profile for each artifact.
- To support reliability growth project, use two testing phases.
Model Description: Defect Flows

- Three inflows, one each for requirements, design, code
- Outflow for each review type, automated analysis, and testing phase
- Flows arrayed by interval
Model Description: Spreadsheet Inputs

- Estimated job size in KSLOC.
- Interval durations
- Estimated phase durations and degrees of phase concurrency such that they sum to the project duration.
- Delay from start of phase for starting peer reviews in each phase
- Relative effectiveness estimates:
  - Relative effectiveness of requirements, design, and code reviews in finding requirements defects.
  - Relative effectiveness of design and code reviews in finding design defects.
  - Relative effectiveness of the two test phases in finding defects (requires definition of the differences between the two phases).
- For each interval:
  - Settings for defect drivers (COCOMO II factors), including effort multipliers and scale factors.
  - Usage profile of quality levels for each defect removal activity.
Model Outputs: Project Defect Profile

![Graph showing project defect profile through testing 1 and testing 2 defect removal.](image-url)
Model Outputs: Defects Introduced

Interval 2 began in Month 9.
Model Outputs: Defects Removed

- Requirements Defects Removed by Requirements Reviews
- Design Defects Removed by Design Reviews
- Design Defects Removed by Automated Analysis
- Code Defects Removed by Automated Analysis
- Code Defects Removed by Code Reviews
- Code Defects Removed by Testing 1
- Code Defects Removed by Testing 2

© The Aerospace Corporation 2009
Model Testing and Usage

• Sensitivity
  – *Product size dominates*
  – *Next, nominal defect introduction values*
  – *QAFs (lognormally distributed)*

• Replication
  – *Two space system flight software projects*
  – *Project A: 68 KSLOC (Ada)*
    • Revised in its 8th year during testing.
    • Average QAF change from 3.2 to 1.5
  – *Project C: 99 KSLOC (50 Ada and 49 assembly)*
    • Redesigned during its third year.
    • Average QAF change from 11.4 to .31
    • Had better use of peer reviews, matured sooner.
Major Defect Discovery Profiles for Projects A & C, actual and modeled
Lessons Learned

• COQUALMO values for nominal defects introduced (10, 20, and 30 defects /KSLOC for requirements, design, and code) appear to be high.
  – Values between .5 (Project C requirements) and 6.1 (Project C code) were used to produce the modeled curves.
• The need to adjust the usage profiles suggests that either COQUALMO’s DRF values require adjustment, or the usage of defect removal activities was reported inaccurately, or both.
• Software development projects seem to have characteristic defect discovery profiles.
  – Dynamic COQUALMO can replicate a discovery profile and, by inference, produce a realistic defect profile for use in managing quality effort in an organization’s future projects.
Defect Removal Submodel

• Defect Drivers in COQUALMO QAFs

  - Required Software Reliability (RELY)
  - Required Reusability (RUSE)
  - Process Maturity (PMAT)
  - Main Storage Constraint (STOR)
  - Analyst Capability (ACAP)
  - Applications Experience (AEXP)
  - Language and Tool Experience (LTEX)
  - Use of Software Tools (TOOL)
  - Development Schedule (SCED)
  - Precedentedness (PREC)
  - Team Cohesion (TEAM)
  - Documentation Match to Life-Cycle Needs (DOCU)
  - Data Base Size (DATA)
  - Product Complexity (CPLX)
  - Execution Time Constraint (TIME)
  - Platform Volatility (PVOL)
  - Programmer Capability (PCAP)
  - Platform Experience (PEXP)
  - Personnel Continuity (PCON)
  - Multisite Development (SITE)
  - Disciplined Methods (DISC)
  - Development Flexibility (FLEX)
  - Architecture/Risk Resolution (RESL)
COQUALMO and Simulation Models

- In a model of software development project, add COQUALMO-based defect co-flows of artifact development
  - Quality focus on residual defect density
  - Advantage: quality factors reflect dynamic project environment
  - Disadvantage: doesn’t relate artifact defects to downstream activities
  - Choi and Bae (2006) developed a COCOMO II-based model
  - Tawileh et al. (2007) reused Abdel-Hamid and Madnick’s model
- Model only defect flows using COQUALMO, estimated durations, and Rayleigh curves
  - Quality focus on defect management
  - Advantage: simulates dynamic project environment and defectivity profiling
  - Disadvantage: requires calibration with defect datasets
  - Madachy and Boehm (2008): Defectivity profile composed of generation and detection rates for each ODC type
  - This work: Defectivity profile composed of generation and detection rates for each activity
Composing Defect Profiles with Rayleigh Curves

- Rayleigh distributions for project effort loading (Norden, Putnam)
- For a given set of project conditions,
  - defect generation $\propto$ development effort
  - defect discovery $\propto$ defect generation

$\therefore$ use Rayleigh distributions represent defect discovery

- Phase level: Kan (2003), Modroiu and Schieferdecker (2006)
- Lower levels: Madachy and Boehm (2008)
- Activity level
  - Intuitive appeal of shape
  - Easy to implement as function of amount flowing and time
  - Assumptions often satisfied “in the small”
Rayleigh Curve Implementation

\[
\text{rate} = (\text{total amount to be processed} - \text{amount processed}) \times \text{time} \times \text{buildup parameter}
\]

- COQUALMO provides \textit{total amount to be processed}
- Stock accumulates \textit{amount processed}
- \textit{buildup parameter} = \textit{(coefficient \times fractional duration\textsuperscript{exponent}}) / \textit{planned development duration}

- Generate Rayleigh curves for
  - 3 months < \textit{planned development duration} < 60 months
  - .05 < \textit{fractional duration} < 1.0
- Fit curves to results to obtain exponents and coefficients
- Fit curves to exponents and to coefficients
  \[
  \text{exponent} = -0.01 \times \ln(\text{planned development duration}) - 2.0377 \quad (R^2=.62)
  \]
  \[
  \text{coefficient} = 6.3889 \times \text{planned development duration}^{-1.0564} \quad (R^2=.99)
  \]
References

- Trachtenberg, M.: Discovering how to ensure software reliability. RCA Engineer (Jan-Feb 1982) 53-57.
Acronyms

- ACAP: Analyst Capability
- COCOMO II: COnstructive COst MOdel II
- COQUALMO: COnstructive QUALity MOdel
- DC: Dynamic COQUALMO
- DI: number of defects introduced
- DIR_{nom}: nominal defect introduction rate
- DR: number of defect removed
- DRF: defect removal fraction
- KSLOC: thousand source lines of code
- ODC: orthogonal defect classification
- QAF: quality adjustment factor
- Size^B: software size raised to a scale factor
A Simulation Method for Defectivity Profiling

Dan Houston, Ph.D.
The Aerospace Corporation

Computers and Software Division
April 23, 2009
Daniel.X.Houston@aero.org
310-336-0732