Software Analytics

Structural Quality Measurement and the State of IT Software

Dr. Bill Curtis
Chief Scientist, CAST & SVP, CAST Research Labs
Director, Consortium for IT Software Quality (CISQ)
“As higher levels of assurance are demanded…testing cannot deliver the level of confidence required at a reasonable cost.”

“The cost of preventing all failures will usually be prohibitively expensive, so a dependable system will not offer uniform levels of confidence across all functions.”

“The correctness of the code is rarely the weakest link.”

Internal Quality Is Often Overlooked

**Quality**

The degree to which a product meets its specified requirements

**Problem**—Customers struggle to state functional requirements. They do not understand non-functional requirements.

“...a failure to satisfy a non-functional requirement can be critical, even catastrophic...non-functional requirements are sometimes difficult to verify. We cannot write a test case to verify a system’s reliability...The ability to associate code to non-functional properties can be a powerful weapon in a software engineer’s arsenal.”
What Is An Application

Product Catalogue
- ? hosts
- ? threads

Order Entry Application
- ? hosts
- ? threads

Credit Card Application
- ? hosts
- ? threads

Express Service Application
- ? hosts
- ? threads

Retail Website
- ? hosts
- ?K threads

GODOT’s lament:
Ever have an application hang waiting for a response that will never come?

Resource pools?
Connection pools?
Error handling?
Timeouts?

PSM 2011—Curtis—13JUL11—© CAST 2011
Revising Our Understanding of Defects

Study of defects across 1 open source and 2 large NASA applications

<table>
<thead>
<tr>
<th>Observation</th>
<th>Percent of cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixes mapping to &gt; 2 files</td>
<td>≈ 60%</td>
</tr>
<tr>
<td>Fixes mapping to &gt; 3 files</td>
<td>≈ 30-40%</td>
</tr>
<tr>
<td>Fixes mapping to &gt; 2 components</td>
<td>≈ 10-36%</td>
</tr>
<tr>
<td>Fixes mapping to &gt; 2 subsystems</td>
<td>≈ 10-20%</td>
</tr>
<tr>
<td>Narrow spread of faults</td>
<td>80% of faults in 20% of files</td>
</tr>
<tr>
<td>Types of defects</td>
<td>Requirements 33%</td>
</tr>
<tr>
<td></td>
<td>Coding 33%</td>
</tr>
<tr>
<td></td>
<td>Data 14%</td>
</tr>
</tbody>
</table>

Application Quality vs. Code Quality

Application Quality
Application quality also measures how well the individual components work together to make up the overall business system.

Component Quality
Code quality is the measure of individual components for compliance with standards and best practices in the context of a specific language.

Good component quality ≠ Good application quality
<table>
<thead>
<tr>
<th>Wave</th>
<th>Description</th>
<th>What:</th>
<th>When:</th>
<th>Why:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Languages</td>
<td>3rd &amp; 4th generation languages, structured programming</td>
<td>1965-1980</td>
<td>Give developers greater power for expressing their programs</td>
</tr>
<tr>
<td>2</td>
<td>Methods</td>
<td>Design methods, CASE tools</td>
<td>1980-1990</td>
<td>Give developers better tools and aids for constructing software systems</td>
</tr>
<tr>
<td>3</td>
<td>Process</td>
<td>CMM, ITIL, PMBOK, Agile</td>
<td>1990-2000</td>
<td>Provide a more disciplined environment for professional work incorporating best practices</td>
</tr>
<tr>
<td>4</td>
<td>Product</td>
<td>Architecture, Quality characteristics, Reuse</td>
<td>2000</td>
<td>Ensure software is constructed to standards that meet the lifetime demands placed on it</td>
</tr>
</tbody>
</table>
Software Physiology
How the Data Were Collected

CAST APPLICATION PLATFORM

LANGUAGE PARSERS

Oracle PL/SQL
Sybase T-SQL
SQL Server T-SQL
IBM SQL/PSM
C, C++, C#, Pro C
COBOL
CICS
Visual Basic
VB.Net, ASP.Net
Java, J2EE
JSP
XML
HTML
Javascript
VBScript
PHP
PowerBuilder
Oracle Forms
PeopleSoft
SAP ABAP,
Netweaver
Tibco
Business Objects
Universal Analyzer

QUALITY METRICS

Security
Performance
Changeability

APPLICATION METADATA

Presentation
Configuration Utility
JSP/Val Configuration File
Servlets/Beans
Java Tasks
Universal Analyzer
Schemas
Databases
Tables
Columns
The Computation of Quality

Quality Metrics Subset

- SQL Complexity Distribution
- Class complexity (Inher. depth)
- Class complexity (Inher. width)
- Artifacts having recursive calls
- Method complexity (control flow)
- Multiple artifacts inserting data on the same SQL table
- Coupling Distribution
- File conformity
- Dead code
- Structuredness
- Controlled data access
- Empty code
- Modularity
- Encapsulation conformity
- Inheritance
- Package naming
- Class naming
- Interface naming
- Package comment
- Class comment
- Method comment
- Package size
- Class size (methods)
- Interface size

Quality Indicators

- Complexity
- Architecture
- Programming Practices
- Naming Conventions
- Documentation

Health Factors

- Performance
- Robustness
- Security
- Transferability
- Changeability
- Maintainability

Application Quality

Immediate Impact

On-Going Impact

Over 800+ architectural and language-specific code checks
Uses of Structural Quality Measures

IT Executives

Vendor Managers

Deliverables insight

App / Project Managers

Application insight

Developers

Remedial insight

Portfolio insight
## The Study Sample

<table>
<thead>
<tr>
<th>Industry</th>
<th>No. of Companies</th>
<th>No. of Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy &amp; Utilities</td>
<td>4</td>
<td>23</td>
</tr>
<tr>
<td>Financial Services</td>
<td>10</td>
<td>54</td>
</tr>
<tr>
<td>Insurance</td>
<td>12</td>
<td>25</td>
</tr>
<tr>
<td>IT and Business Consulting</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>11</td>
<td>25</td>
</tr>
<tr>
<td>Other</td>
<td>11</td>
<td>21</td>
</tr>
<tr>
<td>Public Administration</td>
<td>10</td>
<td>63</td>
</tr>
<tr>
<td>Software ISV</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>Telecommunications</td>
<td>5</td>
<td>51</td>
</tr>
<tr>
<td><strong>Grand Total</strong></td>
<td><strong>74</strong></td>
<td><strong>288</strong></td>
</tr>
</tbody>
</table>
Distribution of Application Size in KLOC
### Industry Segments & Technologies

<table>
<thead>
<tr>
<th>Industry</th>
<th>.NET</th>
<th>ABAP</th>
<th>C/C++</th>
<th>COBOL</th>
<th>Java EE</th>
<th>Mixed Technologies</th>
<th>Oracle 4GL</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy &amp; Utilities</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td>17</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Financial Services</td>
<td>1</td>
<td></td>
<td>6</td>
<td>18</td>
<td>26</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Insurance</td>
<td>3</td>
<td>1</td>
<td></td>
<td>7</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IT and Business Consulting</td>
<td></td>
<td>2</td>
<td>1</td>
<td></td>
<td>6</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>3</td>
<td>10</td>
<td>2</td>
<td>7</td>
<td></td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Public Administration</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>2</td>
<td>35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software ISV</td>
<td>1</td>
<td>2</td>
<td></td>
<td>8</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Telecommunications</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>35</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Grand Total</td>
<td>14</td>
<td>23</td>
<td>9</td>
<td>27</td>
<td>123</td>
<td>45</td>
<td>40</td>
<td>7</td>
</tr>
<tr>
<td>Language</td>
<td>Violations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Java EE</td>
<td>Use of accessors to Private Fields&lt;br&gt;Artifacts with <strong>High Fan-Out</strong>&lt;br&gt;Unreferenced Methods&lt;br&gt;Unreferenced Fields</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.NET</td>
<td>Declaring Public Class Fields&lt;br&gt;Artifacts with <strong>High Fan-Out</strong>&lt;br&gt;Classes with a High Lack of Cohesion&lt;br&gt;Artifacts with High Fan-In</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Artifacts with <strong>High Fan-Out</strong>&lt;br&gt;Large Functions - too many Lines of Code&lt;br&gt;Functions with SQL statement including Subqueries&lt;br&gt;Artifacts with High Cyclomatic Complexity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C++</td>
<td>Data Members that are not Private&lt;br&gt;Artifacts with <strong>High Fan-Out</strong>&lt;br&gt;Included files including other files&lt;br&gt;Artifacts with High Cyclomatic Complexity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABAP</td>
<td>Artifacts with a Complex SELECT Clause&lt;br&gt;Artifacts with <strong>High Fan-Out</strong>&lt;br&gt;Artifacts with High Cyclomatic Complexity&lt;br&gt;Artifacts with High Essential Complexity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Distribution of Security Scores

Frequency

1.00 1.50 2.00 2.50 3.00 3.50 4.00
Security Distributions by Language

![Box plot showing security distributions for various languages]

- .NET
- C/C++
- COBOL
- Java EE
- Oracle 4GL

Security values range from 1 to 4.
Distribution of Performance Scores
Performance Distributions by Language

The diagram shows performance scores for different languages, including .NET, C/C++, COBOL, Java EE, and Oracle 4GL. The scores range from 2.2 to 4.0, with COBOL having the highest average score and Oracle 4GL having the lowest.
Distribution of Changeability Scores
Changeability Distribution by Industry

![Box plot showing changeability scores by industry](image_url)
Does Size Matter?

Total Quality Index vs K Lines of Code
$R^2$ – TQI with Size for COBOL Apps

$R^2 = .45$

Total Quality Index

K Lines of Code
% Complex Components by Language

Technologies

% of High Complex Objects

.NET  C/C++  COBOL  Java EE  Oracle 4GL
Use of PERFORM ... THROUGH | THRU
Missing WHEN OTHER when using EVALUATE
Components with High Cyclomatic Complexity
CISQ — An Industry Response

Co-sponsorship

IT Executives

CISQ

Technical experts

Carnegie Mellon Software Engineering Institute

OMG

IT Executives

CISQ

Technical experts