How Can We Advance Structural Quality Analysis with Standards and Machine Learning?

Bill Curtis, Executive Director, Consortium for IT Software Quality (CISQ), Senior VP & Chief Scientist, CAST Software; Head of CAST Research Labs

October 3 | 1:00 PM ET

- Today’s session will be recorded.
- An archive of today’s talk will be available at: www.sercuarc.org/serc-talks/
- Use the Q&A box to queue questions, reserving the chat box for comments, and questions will be answered during the last 5-10 minutes of the session.
- If you are connected via the dial-in information only, please email questions or comments to SERCtalks@stevens.edu.
- Any issues? Use the chat feature for any technical difficulties or other comments, or email SERCtalks@stevens.edu.
The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense, ASD(R&E), nor the SERC.

No Warranty. This Stevens Institute of Technology Material is furnished on an “as-is” basis. Stevens Institute of Technology makes no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material. Stevens Institute of Technology does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

This material has been approved for public release and unlimited distribution.
How Can We Advance Structural Quality Analysis with Standards and Machine Learning?

SERC Webinar
October 3, 2018

Dr. Bill Curtis
Executive Director, CISQ

CISQ
Consortium for IT Software Quality
Era of the 9-Digit Glitch

9-Digit Glitches now affect

- Board of Directors
- CEO, COO, CFO
- Business VPs
- Corporate Auditors
- CIO

accountable for

- Governance
- Financial outcomes
- Risk management
- Brand protection
- Customer experience

Structural Quality Standards to Evaluate Application Risk
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.”
Apps Are a Technology Stack

Multi-language, multi-layer Architecture

1. Unit Level
   - Code style & layout
   - Expression complexity
   - Code documentation
   - Class or program design
   - Basic coding standards
   - Developer level

2. Technology Level
   - Single language/technology layer
   - Intra-technology architecture
   - Intra-layer dependencies
   - Inter-program invocation
   - Security vulnerabilities
   - Development team level

3. System Level
   - Integration quality
   - Architectural compliance
   - Risk propagation
   - Application security
   - Resiliency checks
   - Transaction integrity
   - Function point,
   - Effort estimation
   - Data access control
   - SDK versioning
   - Calibration across technologies
   - IT organization level
<table>
<thead>
<tr>
<th>Wave</th>
<th>What</th>
<th>When</th>
<th>Why</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Language</td>
<td>1965-1980</td>
<td>Give developers greater power for expressing programs</td>
</tr>
<tr>
<td>2</td>
<td>Method</td>
<td>1980-1990</td>
<td>Give developers better aids to construct systems</td>
</tr>
<tr>
<td>3</td>
<td>Process</td>
<td>1990-2002</td>
<td>Improve software management and discipline</td>
</tr>
<tr>
<td>4</td>
<td>Product</td>
<td>2002</td>
<td>Improve engineering of software products</td>
</tr>
</tbody>
</table>

**What:** CMM, ITIL, PMBOK, Agile

**Why:** Improve software management and discipline
Study of structural quality measures and corrective maintenance effort across 20 customers in a large global system integrator.

\[ r^2 = .34 \]

TQI increase of .24 decreased corrective maintenance effort by 50%
Measuring an Agile/DevOps Transformation

Agile/DevOps transformation at Fannie Mae

### Maturity Affects Quality — Level 1 Worst

<table>
<thead>
<tr>
<th>Level</th>
<th># apps</th>
<th>Robust</th>
<th>Security</th>
<th>Perform</th>
<th>Change</th>
<th>Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMMI Level 3</td>
<td>22</td>
<td>3.40</td>
<td>3.32</td>
<td>3.28</td>
<td>3.26</td>
<td>3.08</td>
</tr>
<tr>
<td>CMMI Level 2</td>
<td>27</td>
<td>3.44</td>
<td>3.40</td>
<td>3.31</td>
<td>3.37</td>
<td>3.09</td>
</tr>
<tr>
<td>CMMI Level 1</td>
<td>23</td>
<td>3.22</td>
<td>3.06</td>
<td>3.11</td>
<td>3.14</td>
<td>2.93</td>
</tr>
</tbody>
</table>

|                | % variance explained | 28%   | 25%      | 15%     | 24%    | 12%      |

Java EE applications analyzed with CAST Application Intelligence Platform — scale is 1.0 to 4.0

Stabilizing project schedules and baselines (Level 2) allows developers to work in an orderly professional way. Standardizing processes (Level 3) did not improve structural quality and may have its benefits more on economy of scale issues.
## Method Affects Quality — Hybrid best

Java EE applications analyzed with CAST Application Intelligence Platform — scale is 1.0 to 4.0

Hybrid method’s mix of early attention to architecture combined with structural analysis in short cycle iterations produces the best structural quality outcomes.

### # apps & Metrics

<table>
<thead>
<tr>
<th>Method</th>
<th># apps</th>
<th>Robust</th>
<th>Security</th>
<th>Perform</th>
<th>Change</th>
<th>Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybrid mix</td>
<td>51</td>
<td>3.36</td>
<td>3.27</td>
<td>3.37</td>
<td>3.27</td>
<td>3.08</td>
</tr>
<tr>
<td>Agile</td>
<td>77</td>
<td>3.26</td>
<td>3.14</td>
<td>3.22</td>
<td>3.12</td>
<td>3.00</td>
</tr>
<tr>
<td>Waterfall</td>
<td>60</td>
<td>3.24</td>
<td>3.21</td>
<td>3.16</td>
<td>3.13</td>
<td>2.95</td>
</tr>
<tr>
<td>None</td>
<td>20</td>
<td>3.13</td>
<td>3.15</td>
<td>3.14</td>
<td>3.00</td>
<td>2.92</td>
</tr>
</tbody>
</table>

### % variance explained

- Robust: 10%
- Security: n.s.
- Perform: 6%
- Change: 14%
- Transfer: 6%
# Machine Learning

## Machine Learning
- Creating and using models learned from data
- Learning without being explicitly programmed
- Performance improves as more data is analyzed

## Methods
- Statistical → model fitting → Predict
- Unsupervised → model discovery → Suggest
- Supervised → model training → Categorize
- Reinforcement → outcome training → Predict

## Some Use Cases
- Defect prediction → regression methods
- Behavior prediction → supervised learning
- Micro-service discovery → clustering, training
Comparison of Machine Learning Methods

- Reinforcement
- Supervised
- Unsupervised
- Statistical

Notes from the AI frontier: Applications and value of deep learning
M. Chui, J. Manyika, M. Miremadi, N. Henke, R. Chung, P. Nel, & S. Malhotra
McKinsey Global Institute
Does Machine Learning Add Capability?

Would a machine learning application add significantly better prediction than a simpler measure?

The power of machine learning may be in identifying issues long before they are evident from simpler measures.
CISQ is chartered to define automatable measures of software size and quality that can be measured in the source code, and promote them to become Approved Specifications of the OMG®.
CISQ/OMG Standards Process

- Automated Function Points
- Reliability
- Performance Efficiency
- Security
- Maintainability

Approved Standards

OMG

ISO Fasttrack

Deployment Workshops
### CISQ Structural Quality Characteristic Measures

<table>
<thead>
<tr>
<th>Category</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>22 weaknesses (Top 25 CWEs)</td>
</tr>
<tr>
<td>Reliability</td>
<td>29 weaknesses</td>
</tr>
<tr>
<td>Performance Efficiency</td>
<td>15 weaknesses</td>
</tr>
<tr>
<td>Maintainability</td>
<td>20 weaknesses</td>
</tr>
</tbody>
</table>

#### Example architectural and coding violations included

- SQL injection
- Cross-site scripting
- Buffer overflow
- Empty exception block
- Unreleased resources
- Poor error handling
- Expensive loop operation
- Un-indexed data access
- Unreleased memory
- Excessive coupling
- Dead code
- Hard-coded literals
CISQ Automated Technical Debt Measure

Sum of all efforts-to-fix for all weaknesses in each CISQ Structural Quality Measure

- **Reliability weaknesses**
- **Security weaknesses**
- **Performance weaknesses**
- **Maintainability weaknesses**

Automated Technical Debt

- Predict effort for corrective maintenance
- Predict cost of corrective maintenance

Sum of efforts-to-fix for all instances of each weakness

Weighted effort-to-fix for each instance of a weakness
CISQ and ISO 25000 Standards

- ISO 25010 defines quality characteristics and sub-characteristics
- CISQ conforms to ISO 25010 quality characteristic definitions
- ISO 25023 defines measures, but not at the source code level
- CISQ supplements ISO 25023 with source code level measures

CISQ defined automatable measures for quality characteristics highlighted in blue
As a greater portion of mission, business, and safety critical functionality is committed to software, we hold the following propositions as paramount:

1. Engineering discipline in product and process
2. Quality assurance to risk tolerance thresholds
3. Traceable provenance of all system components
4. Proactive defense of the system and its data
5. Resilient and safe operations
CISQ Membership is Free  www.it-cisq.org

Consortium for IT Software Quality

The Consortium for IT Software Quality™ (CISQ™) is an IT industry leadership group comprised of IT executives from the Global 2000, system integrators, outsourced service providers, and software technology vendors committed to introducing computable metrics standards for measuring software quality & size. CISQ is a neutral, open forum in which customers and suppliers of IT application software can develop an industry-wide agenda of actions for improving IT application quality to reduce cost and risk.

Visit the Cyber Resilience Summit Knowledge Repository

CISQ Sponsors

CYBER RESILIENCE SUMMIT

Strategies to Modernize & Secure Government IT

MARCH 20, 2018 IN RESTON, VA U.S.A.

RSVP TODAY!
Upcoming Events
UPCOMING TALKS:
“Systems and Software Qualities Tradespace Analysis”
Series

“Why Are Ontologies and Languages for Software Quality Increasingly Important?”
Xavier Franch, Full Professor, Polytechnic University of Catalonia (BarcelonaTech)
December 11 | 1:00 PM ET

Please visit the SERC Talks page for more information and updates.
SERC Annual Events | November 7-8

Recognizing a decade of contributions to systems engineering research

SERCUARC.ORG/EVENT/SDSF-2018
SERCUARC.ORG/EVENT/SSRR-2018

Events are free to Academia and Government participants; Industry attendees are charged a nominal fee.

Deadline to Register: October 26
We invite you to visit the collections of research reviews online: https://sercuarc.org/research-reviews/

FOR MORE INFORMATION:
Ms. Monica Brito
mbrito@stevens.edu

LOCATION: FHI360 CONFERENCE CENTER
1825 CONNECTICUT AVE NW, 8TH FLOOR,
WASHINGTON, DC 20009
Thank you for joining us!

Please check back on the SERC website for today’s recording and future SERC Talks information!

Subscribe and follow SERC on these channels:

CONTACT
Editor-in-Chief: Dr. Barry Boehm, University of Southern California – boehm@usc.edu
Webinar Coordinator: Ms. Mimi Marcus, Stevens Institute of Technology – SERCtalks@stevens.edu