ARTICLES & NEWS

Testing - Systematic Code Coverage Techniques
Summary of the Fifth World Congress for Software Quality
Synopsis of ICSQ 2012

ANNOUNCEMENTS

Institute for Software Excellence, May 6-8
HIMSS and ASQ Software Division reach Memo of Understanding
ASQ Becomes a HIMSS-Approved Education Provider
Update on TheSRO.org
Volunteer Opportunities

COLUMNS

Chair’s Message
From the Regions
CSQE Quiz

QUESTIONS OR COMMENTS?

See:
Software Division Web Site

Or Contact:

Chair
Greg Zimmerman
(gregz@appliedqualitysolutions.com)

Chair-Elect
Kandy Senthilmaran
(kanthassamy@hotmail.com)

Secretary
Murali Krishnan
(Murali.Krishnan@mpiiresearch.com)

Treasurer
Brenda Fisk
(bmfisk@golden.net)

Newsletter Editor
Byron Mattingly
(byronmattingly2013@yahoo.com)

ADVERTISE HERE!
EMAIL
SWD@ASQ.ORG

How can we improve Software Quality Live?
Does this issue provide helpful information? Let us know!
Software Quality Live is returning after an unusually long break. As our communications focus has shifted from print to web to social media, we have been working to bring our members more current information more frequently. We have delivered a steady stream of info via the "ASQ Software Division" LinkedIn group, and have also been communicating webinars, conferences, and other events via email. However, as a consequence of that new focus, we have neglected our commitment to produce a quarterly newsletter. Please accept my apology for that lapse and look forward to new issues on a regular basis in the future.

At the end of October 2012, we held another successful International Conference on Software Quality (ICSQ). I think this was either our 20\textsuperscript{th} or 21\textsuperscript{st} ICSQ/WCSQ event. The 2012 event, with the theme "Preparing for Tomorrow's Software Challenges," was held in downtown Indianapolis, Indiana. We had some great talks on risk management, security, Agile, mobile, and other topics along that vein. Unfortunately, this was the same week hurricane Sandy hit the northeastern U.S. which caused travel difficulties for some of our speakers and attendees. Those not impacted by the weather, however, benefitted from highly interactive sessions. The next ICSQ will be in February 2014 in the Dallas, Texas area with the focus returning to software quality in regulated industries.

We will be back in Indianapolis in May of 2013. The Software Division is again sponsoring the Institute for Software Excellence (ISE), a co-located “conference within-a-conference” at ASQ's World Conference on Quality and Improvement (WCQI). Registration to ISE provides access to the entire software program and giveaways as well as all of the regular WCQI sessions. The preliminary brochure you have by now already received lists the ISE program as "Software Sessions" under the "Industry" heading. See the ISE flyer in this issue for additional program details and registration information, or visit the ISE conference web site.

Late in 2013, we will begin updating the Certified Software Quality Engineer (CSQE) Body of Knowledge. This process occurs approximately every five years to ensure the CSQE BoK stays current. Active CSQEs will have the opportunity to contribute to this BoK update through a series of workshops. Keep your eyes peeled for the calls for volunteers and/or contact the CSQE Certification Chair, Velinda Moore, for additional info.

Greg Zimmerman
Software Division Chair, 2011-2013
gregz@appliedqualitysolutions.com
OBITUARY FOR
WILLIAM LANSIN “BILL” TREST JR.
1942 - 2012

Bill Trest, a CSQE and Past Chair of the Software Division, succumbed to his battles with cancer this past August. Bill’s term as Chair was cut short due to his health. A few months before he stepped down, he pulled me aside and asked me to help support his then Chair-Elect when he was gone, and to consider running for the office myself. At that point, I had barely been involved with the division for a year, but Bill saw passion in me which he felt the division needed. I don’t know whether I would have eventually considered running for Chair on my own, but the seed Bill planted worked its magic and I eventually volunteered to fill the mid-term vacancy in Chair-Elect when Nicole Radziwill assumed the Chair role, starting me on the path to my current role.

Thank you, Bill, for your many years of service to the Software Division and for giving a wide-eyed new volunteer a strategic push.

Greg Zimmerman
Software Division Chair, 2011-2013
gregz@appliedqualitysolutions.com

Obituary for William Lansin "Bill" Trest Jr Published in the Star-Telegram: August 29, 2012
The ASQ Software Division is sponsoring the fifth annual Institute for Software Excellence (ISE) with presentations by today’s industry leaders. Practical, useful information will be presented along with thought-provoking ideas for your future. Join other software quality professionals to learn, share, and network.

Preliminary Program

• Influencing Without Authority – Presented by Rick Hefner
• Software Reliability Engineering Contributions to Software Security – Presented by Taz Daugherty
• Risk-based Agile Deployment of Mobile Medical Apps in Healthcare IT – Presented by Byron Mattingly
• Leveraging the CMMI to Pursue Excellence in Software Development – Presented by David Walker
• Risk-based Approach to Grading Software – Presented by David Peercy
• Risk-based Configuration Control – Balancing Flexibility With Stability – Presented by Linda Westfall
• Proving the Value of Quality Through Metrics and Communication – Presented by Carol Dekkers
• Safe and Secure Software Systems and the Role of Professional Licensure – Presented by Phillip LaPlante
• Converting Requirements Into Tests – Presented by Louise Tamres
• Plus, a Panel Discussion on “Software Is Everywhere – and How to Live With It” with Taz Daughtrey, Linda Westfall, and Phillip LaPlante

ISE is a component of the ASQ World Conference on Quality and Improvement, which draws thousands of quality professionals from around the world. A registration to ISE includes full access to the ASQ World Conference, with hundreds of presentations on world-class quality techniques.

Visit asq.org/conferences/institute-for-software-excellence for more information.

Questions regarding this exciting program can be directed to Nancy Pasquan at npasquan@eyes-on.us
Update on TheSRO.org
In September 2012, ASQ suspended its site www.thesro.org. This difficult decision was the result of multiple factors including, but not limited to, instability in the site's platform and the outside solicitation of site users. ASQ Head Quarters is currently working on an improved and redesigned version of the site. Version 2.0 of TheSRO.org will be released concurrently with the ASQ World Conference in May 2013. This process is being managed by the SRO Design Team—a group of ASQ member leaders and global SR leaders. The new site will roll over content that users developed, but also include increased functionality and a strong focus on the user experience. All registered users of the site will be notified of the new version's release as soon as it comes online. Follow developments and make suggestions by liking TheSRO on Facebook, following TheSRO on Twitter, joining TheSRO LinkedIn group, or contacting Dave Celata, Social Responsibility Manager at dcelata@asq.org.

Thanks,

Dave Celata
Social Responsibility Program Manager
American Society for Quality

ASQ Becomes an HIMSS-Approved Education Provider

(December, 2012) Julianna Kazragys, Manager, Certification Professional Development at HIMSS has confirmed that ASQ has been approved as a HIMSS-approved education provider. Over 40,000 HIMSS members may now receive continuing education and recertification credits for some of ASQ's eligible educational and session offerings.

ASQ members can likewise count HIMSS education credits towards ASQ recertification (per the ASQ Recertification Journal guidelines).
The 5WCSQ, Fifth World Congress for Software Quality, was held October 31-November 3, 2011 in Shanghai, China. In addition to the keynote speakers, there were 13 presentations and papers from China, 16 from Europe, 3 from the USA, and 20 from Japan, for a total of 52 papers. We would like to thank the presenters representing America: Richard Levielle of Synopsis Inc., Keith Stobie of Doyenz Inc. (work created while at Microsoft), and Patricia McQuaid of California Polytechnic State University, ASQ-Software Division, and the ASTQB. Richard presented two papers, entitled “A Tailored Approach to Effective and Efficient Software Process Maturity Improvement” and “Test Strategy for High Quality EDA Software”. Keith presented a paper entitled “Testing Services in Production”. Patricia presented a Keynote speech, entitled “Software Disasters - Understanding the Past, to Improve the Future”.

The plan for the 6WCSQ, Sixth World Congress for Software Quality, is that it will rotate back to Europe and be held near the end of June, 2014 in London England. The WCSQ is sponsored by the American, European, and Japanese software quality organizations.

Best Papers – 5WCSQ
- “Process Improvement Using XDDP-Application of XDDP to the car Navigation System” by Keiji Kobata, Eiji Nakai, Takahiro Tsuda - Japan

- “Success Factors to Achieve Excellent Quality - CMMI Level 5 Organizations Research Report” by Naomi Honda, Japan

- “Design and Applications of Test Cases in the Business Process” by Zhenyu Liu, Xu Jian, Lizhi Cai, China

Best Presentations – 5WCSQ
- “Success Factors to Achieve Excellent Quality - CMMI Level 5 Organizations Research Report” by Naomi Honda, Japan (Not only did Naomi win Best Paper and Best Presentation at the 5WCSQ, but she won a Best Paper award at the 5WCSQ!)

- “Ever Been Fooled by Performance Testing Results?” by Mieke Gevers, Belgium

- “Utilization of Domain-Specific Knowledge for Quality Software Design” by Noriko Iizumi, Japan

The WCSQ has been a partnership of the Union of Japanese Scientists and Engineers (JUSE, Japan) – Software Quality Profession (SQiP), the American Society for Quality – Software Division (ASQ-SD) and the European Organization for Quality - Software Group (EOQ-SG). The first World Congress on Software Quality (1WCSQ) was held in San Francisco, USA in 1995 on the basis of 10 years of professional exchanges among the three organizations. The 2WCSQ was held in Yokohama, Japan in 2000 under the slogan, “Software Quality for the Coming New Millennium.” The 3WCSQ was held in Munich, Germany in 2005. 4WCSQ was held in Bethesda Maryland, USA in 2008 under the theme of “World-Class Software Quality Creating Competitive Advantage Worldwide.” The 5WCSQ’s theme was “United Under One Banner: Best of Best Quality”.

SUMMARY OF THE FIFTH WORLD CONGRESS FOR SOFTWARE QUALITY
In conjunction with HIMSS Professional Development and Informatics Sections, the Cleveland Clinic hosted a one-day, in-person educational event on 31 May 2012 exploring how nursing informatics can impact daily practice:

**Session 1: Informatics Nurses: What? Where? And how many?**
Joyce Sensmeier, MS, RN, BC, FAAN, CPHIMS, FHIMSS
- Vice President, Informatics, HIMSS

**Session 2: The Future of Nursing and the Role of Informatics**
Dana Alexander, RN, MSN, MBA, FHIMSS, FAAN, VP & Chief Nursing Officer, GE Health Care IT Solutions

**Session 3: Dialogue with Professional Panel**

**Session 4: Closing Keynote: Understand the MU Alphabet Soup**
Judy Murphy, RN, FACMI, FHIMSS, FAAN, Deputy National Coordinator for Programs & Policy, Office of the National Coordinator for Health IT

The sessions were given by key opinion leaders in nursing informatics (NI), and all stressed the importance of Software Quality and the need for it in NI, some even mentioning ASQ by name. In fact, the last slide of the morning presentation introducing the various presenters for the day was dedicated to ASQ and the relevance of Software Quality!

According to Helen Figge, Senior Director, Professional Development, Career Services, HIMSS, “Participants Learned from the nation’s leaders in Nursing Informatics while building upon career portfolio and enhancement of knowledge of the role of nurses in implementing information technology to enhance nursing workflow, promote patient safety and yield outcomes that improve patient care.”

The ASQ Software Division also had a display table at the event. Over 200 people attended and many of them came over to the ASQ Software Division table. They asked lots of good questions and took the materials from the table as fast as they could be replaced. The typical visitor was an IS / IT Manager / Director of Nursing Informatics for their hospital / healthcare provider network.
The Software Division Wants You!

As a volunteer led organization, the Software Division is always in search of passionate individuals looking for a way to give back to the profession. Whether your interest is for personal or professional growth, your help is needed to improve the practice of software quality. Opportunities include:

- Speak at a section meeting
- Conference planning, including local / regional marketing and coordination
- Create / manage social media or other online content
- Write an article for the Software Quality Professional journal
- Speak at a national or regional conference
- CSQE exam development workshops (question writing, exam reviews, and BoK updates)
- Write a software related article for QP magazine
- Present a webinar on a relevant quality topic or practice
- Craft and coordinate division communications
- Staff the Software Division booth at an event

Contact the Software Division Chair or anyone else on the division leadership team to express your interest.

Software Community in India

Dear Fellow ASQ Software Division Members,

Over the last decade the Indian Information Technology (IT) industry has become the key contributor for operating, transforming and innovating IT services. It has helped organizations improve efficiency and effectiveness of not only IT functions, but also key business processes. Many of you have been part of this evolution. We have gained a lot being part of this industry. It is imperative for us that we come together to consolidate our learning and share it with industry.

I invite you to join the 'ASQ Software Division - India' community on LinkedIn. Objectives of the group are:

- Enable learning and share knowledge about role of Quality in Information Technology with professional members
- Spread reach and improve awareness about ASQ Software Division in India
- Improve contributions in ASQ Software Division from India

Let’s be the force.

Best wishes,
Haresh Amre

SQP Journal Seeks Volunteers

Software Quality Professional (SQP) is an ASQ journal supported by the ASQ Software Division. Each issue has 3 to 4 peer-reviewed papers, book reviews, and a Quality Nugget. If you would like to write a paper for the journal – see the author guidelines. If you would like to write a Quality Nugget - practical tips that can be easily used by our readers - submit an article of around 1000 words to Rupa Mahanti, rupa.mahanti@tcs.com. If you would like to volunteer to review articles or books, or write a letter to the editor, contact Mark Paulk, mark@paulk123.com. In each quarterly issue there is a paper that is available to non-subscribers. The abstract of that article and a link to the current article are in the Columns section of this newsletter.
1. A quality action team needs to generate lots of improvement suggestions in a short period of time. Which of the following tools would the team’s facilitator be most likely to recommend?
   A. Brainstorming
   B. Affinity diagram
   C. Story board
   D. Force field analysis

2. Root causes of process problems in the process being incrementally improved are determined during which of the following steps in the Six Sigma DMAIC process?
   A. Define
   B. Measure
   C. Analyze
   D. Improve

3. Based on the Yourdon/Demarco notation for data flow diagrams the rectangle labeled Element 1 in the diagram above is:
   A. an entity external to the system
   B. a data item
   C. a function inside the system that transforms data
   D. a repository for data (e.g., buffer, queue, data file, database)

4. A project manager estimates that an activity will take one software engineer 20 hours of effort to complete. However, due to lack of sufficient personnel, the engineer performing the task can only work on it part time. This will result in:
   A. a decrease in the activity’s estimated duration.
   B. an increase in the activity’s estimated duration.
   C. a decrease in the activity’s estimated effort.
   D. an increase in the activity’s estimated effort.

5. Which of the following is an example of a derived measure?
   A. Hours of effort expended in preparing for a peer review
   B. Number of defects found during the peer review
   C. Number of pages in the document being peer reviewed
   D. Defect density of the document being peer reviewed

6. Which of the following would be an example of an appropriate test suspension criterion to define in a software system test plan?
   A. More than 20% of the system test cases are blocked
   B. The arrival rate of problems reported by testers is increasing
   C. 95% of the planned test cases have been executed and passed
   D. No more than 10 non-closed majors exist, all with workarounds

7. Combining two or more versions of a software product to create a single new version is called:
   A. building
   B. merging
   C. compiling
   D. linking
In 2012, the Software Division council launched a revamped regional engagement focused program led by Territory Councilors (TC). Each Territory will have a Councilor and they are responsible for collaborating with the regional ASQ sections and other Quality events to deliver Software Quality sessions. The primary focus is to provide opportunities for Software Division members to get access to the latest updates and learning in software quality and networking without having to travel too far from where they live.

This is a rough map of the territories – each ASQ region maps to a TC, along with separate regions for Europe, China, India and Latin America.

EMEA - Europe, Middle East and Africa; LATAM - Latin America
Here is the latest list of the TCs:

<table>
<thead>
<tr>
<th>Region</th>
<th>Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Northeast</td>
<td>Scott Ankrum</td>
<td><a href="mailto:ankrums@mitre.org">ankrums@mitre.org</a></td>
</tr>
<tr>
<td>North central</td>
<td>Wil Gutierrez</td>
<td><a href="mailto:joevolley@gmail.com">joevolley@gmail.com</a></td>
</tr>
<tr>
<td>Northwest</td>
<td>Ashish Soni</td>
<td><a href="mailto:Ashish.Soni@microsoft.com">Ashish.Soni@microsoft.com</a></td>
</tr>
<tr>
<td>Southeast</td>
<td>Kristal Ray</td>
<td><a href="mailto:kristal.ray@oracle.com">kristal.ray@oracle.com</a></td>
</tr>
<tr>
<td>South central</td>
<td>David Peercy</td>
<td><a href="mailto:depeerc@sandia.gov">depeerc@sandia.gov</a></td>
</tr>
<tr>
<td>Southwest</td>
<td>Nancy Pasquan</td>
<td><a href="mailto:npasquan@eyes-on.us">npasquan@eyes-on.us</a></td>
</tr>
<tr>
<td>EMEA</td>
<td>Zigmund Bluvband</td>
<td><a href="mailto:zigmund@ald.co.il">zigmund@ald.co.il</a></td>
</tr>
<tr>
<td>India</td>
<td>Haresh Amre</td>
<td><a href="mailto:haresh1712@yahoo.co.in">haresh1712@yahoo.co.in</a></td>
</tr>
<tr>
<td>China</td>
<td>TBD</td>
<td></td>
</tr>
<tr>
<td>LATAM</td>
<td>TBD</td>
<td></td>
</tr>
</tbody>
</table>

We are still looking for member leaders for the China and LATAM TC roles. If you know anyone who may be interested in these roles, please ask them to write to kanthassamy@hotmail.com

We are successfully piloted this model with ASQ Southern California Quality Conference (SCQC) for 2011 and 2012, with a separate software quality track in the conference. We are collaborating with other regional conferences to do a similar engagement or to have software quality topics sponsored by the division in the conference.

Please reach out to individual TCs if you are interested to know more about activities planned for the region or if you any opportunities to engage with a regional conference.

Thank you,

Kandy Senthilmaran
ASQ Software Division Chair Elect and TC Coordinator
When performing structural testing of the code, the term paths refer to control flow sequences through the internal structure of the software. There are typically many possible paths between the entry and exit of a typical software application. Every if-then or if-then-else decision doubles the number of potential paths, every case statement multiplies the number of potential paths by the number of cases, and every loop multiplies the number of potential paths by the number of different iteration values possible for the loop. For example, a software unit with a loop that can be iterated from one to 100 times has 100 possible paths (once through the loop, twice through the loop, and so on, up to 100 times through the loop). Add an if-then-else statement inside that loop and that increases the number of paths to 200. Add a case statement with four possible choices inside the loop as well and there are 800 possible paths. Moving from the individual unit to the integration level, if this unit with 800 paths is integrated with a unit that only has two sequential if-then-else statements (four paths), there are now 3200 paths through these two units in combination. Since there are rarely enough resources to test every path through a complex software application or even a complex individual unit, a tester can use white-box logic coverage techniques to systematically select the tests that are the most likely to help identify the yet undiscovered, relevant defects.

To demonstrate the different white-box logic coverage techniques of statement, decision, and condition coverage, the piece of nonsense code shown in Figure 1 will be used.

```
A = 300
if B > 40 and C < 100 then A = 1000
if B < 60 and C < 20 then A = 10
print A
```

**Figure 1: Code with the Input Variables B and C**

**Statement Coverage:** A statement, also called a line, is an instruction or a series of instructions that a computer carries out. Statement coverage, also called line coverage, is the extent that a given software unit/component’s statements are exercised by a set of tests. Statement coverage is the least rigorous type of code coverage technique. To have complete statement coverage, each statement must be executed at least once. Table 1 illustrates that it only takes one test case to have statement coverage of the code in Figure 1. As long as input variables B and C are selected so that both decisions in this code are true, every statement is executed.
### Decision Coverage

A decision determines the branch path that the code takes. To have decision coverage, also called branch coverage, each statement is executed at least once and each decision takes all possible outcomes at least once. For example, if the decision is a Boolean expression, decision coverage requires test cases for both the true and false branches. If the decision is a case statement, decision coverage requires test cases that take each case branch. If decision coverage exists, then statement coverage also exists. Table 2 illustrates the test cases needed to have decision coverage of the code in Figure 1. As illustrated in Figure 2, test case #2 results in the first decision being true and the second decision being false. Test case #3 results in the first decision being false and the second decision being true. Thus, these two test cases in combination provide decision coverage because the true and false paths are taken out of each decision.

### Condition Coverage

A condition is a state that a decision is based on. To have condition coverage, each statement is executed at least once and each condition in a decision takes all possible outcomes at least once. If decision coverage exists, then statement coverage also exists. For the code in Figure 1 there are three conditions that the input variable B can have:

1. $B \leq 40$
2. $40 < B < 60$
3. $B \geq 60$.

There are also three conditions that input variable C can have:

1. $C < 20$
2. $20 \leq C < 100$
3. $C \geq 100$. 

---

**Table 1: Statement Coverage - Example**

<table>
<thead>
<tr>
<th>Test Case #</th>
<th>Inputs</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$B &gt; 40 \text{ and } C &lt; 60$</td>
<td>$C &lt; 20$</td>
</tr>
</tbody>
</table>

**Table 2: Decision Coverage - Example**

<table>
<thead>
<tr>
<th>Test Case #</th>
<th>Inputs</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$B \geq 60$</td>
<td>$C &lt; 20$</td>
</tr>
<tr>
<td>3</td>
<td>$B \leq 40$</td>
<td>$C &lt; 20$</td>
</tr>
</tbody>
</table>

**A = 300**

if $B > 40$ and $C < 100$ then $A = 1000$

if $B < 60$ and $C < 20$ then $A = 10$

print $A$
Table 3 illustrates one choice of test cases that combines these into condition coverage of the code in Figure 1.

<table>
<thead>
<tr>
<th>Test Case #</th>
<th>Inputs</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>B &gt;= 60, C &lt;20</td>
<td>A = 1000</td>
</tr>
<tr>
<td>5</td>
<td>B &lt;= 40, C &gt;= 20 and &lt;100</td>
<td>A = 300</td>
</tr>
<tr>
<td>6</td>
<td>B &gt; 40 and &lt; 60, C &gt;= 100</td>
<td>A = 300</td>
</tr>
</tbody>
</table>

Table 3: Condition Coverage - Example

Note that condition coverage does not always imply decision coverage. For example, as illustrated in Figure 3, in the set of test cases in Table 3, test case 4 results in the first decision being true and the second decision being false. Test cases 5 and 6 result in both decisions being false. Therefore, decision coverage has not been achieved because the true path has not been taken out of the second decision.

Figure 3: Condition Coverage Test Case Results – Test Cases 4, 5 & 6

**Condition and Decision Coverage:** The next level of rigor is to have condition and decision coverage where each statement is executed at least once, each decision takes all possible outcomes at least once, and each condition in a decision takes all possible outcomes at least once. If decision/condition coverage exists, condition coverage, decision coverage, and statement coverage also all exist. Table 4 and Figure 4 illustrate one choice of test cases that provides decision/condition coverage of the code in Figure 1.

<table>
<thead>
<tr>
<th>Test Case #</th>
<th>Inputs</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>B &gt;= 60, C &gt;= 20 and &lt;100</td>
<td>A = 1000</td>
</tr>
<tr>
<td>8</td>
<td>B &gt; 40 and &lt; 60, C &lt;20</td>
<td>A = 10</td>
</tr>
<tr>
<td>9</td>
<td>B &lt;= 40, C &gt;= 100</td>
<td>A = 300</td>
</tr>
</tbody>
</table>

Table 4: Condition & Decision Coverage - Example

Figure 4: Condition & Decision Coverage Test Case Results – Test Cases 7, 8 & 9
Multiple Condition Coverage: To have *multiple condition coverage*, each statement is executed at least once and all possible combinations of condition outcomes in each decision occur at least once. Multiple condition coverage always results in condition, decision, and statement coverage as well. Multiple condition coverage is the most rigorous type of structural coverage testing. Table 5 illustrates a choice of test cases that provides multiple condition coverage of the code in Figure 1.

<table>
<thead>
<tr>
<th>Test Case #</th>
<th>Inputs</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>1</td>
<td>&lt;= 40</td>
<td>&lt;20</td>
</tr>
<tr>
<td>2</td>
<td>&lt;= 40</td>
<td>&gt;= 20 and &lt;100</td>
</tr>
<tr>
<td>3</td>
<td>&lt;= 40</td>
<td>&gt;= 100</td>
</tr>
<tr>
<td>4</td>
<td>&gt; 40 and &lt;60</td>
<td>&lt;20</td>
</tr>
<tr>
<td>5</td>
<td>&gt; 40 and &lt;60</td>
<td>&gt;= 20 and &lt;100</td>
</tr>
<tr>
<td>6</td>
<td>&gt; 40 and &lt;60</td>
<td>&gt;= 100</td>
</tr>
<tr>
<td>7</td>
<td>&gt; 60</td>
<td>&lt;20</td>
</tr>
<tr>
<td>8</td>
<td>&gt; 60</td>
<td>&gt;= 20 and &lt;100</td>
</tr>
<tr>
<td>9</td>
<td>&gt; 60</td>
<td>&gt;= 100</td>
</tr>
</tbody>
</table>

Which Type of Code Coverage to Choose?: To answer to this question, the tester must consider the risk of the code under test. When selecting the level of coverage, the tester should consider the probability that the code under test has one or more yet undiscovered defects that will escape testing and cause field failures, and the cost of those field failures (e.g., cost to the customer/user, cost to correct the defect, cost of propagating the correction back into the field, cost of potential penalty or litigation, cost of lost reputation, cost of lost sale and so on). The higher the probability that defects will escape and cause expensive field failures, the more rigorous the level of coverage that should be used. Of course, the tradeoff is the more rigorous the coverage, the longer it will take and the more it will cost to perform the testing.

Without at least statement coverage, part of the code is not tested at all. If that untested code contains a defect, there is no opportunity to identify that defect during test. Many testers would consider decision coverage the minimum level of coverage necessary and test all code to at least that level of rigor. Testers would then use the higher levels of coverage rigor as appropriate based on the code under test having a higher level of risk.

About the Author: Linda Westfall is president of Westfall Team, Inc. and has more than 35 years of experience in software engineering, software quality and software project management. She is the author of the *Certified Software Quality Engineer Handbook*. Visit her company’s web site at [www.westfallteam.com](http://www.westfallteam.com) or send comments on this article to lwestfall@westfallteam.com.
The conference was heavily focused on AGILE/EXTREME/SCRUM software development strategy and methodology.
Evren is a member of GE’s Software Council and co-lead of the GEHC Software Leadership team. Evren is a graduate of the University of Tennessee, where he earned a master’s and doctorate degree in Nuclear Engineering. He holds over 50 US patents.

IN 50 WORDS OR LESS: There is a seismic shift in healthcare that is fueling growth in healthcare IT. International growth rates range from 2% (Italy) to 17% (China). The shift is from episodic, single-patient, provider centric, fee for service, to continuous population management, patient-centric, payment for value and systems vs. silos. Agile/SCRUM/XP, and LEAN drive satisfaction, friendly solutions, replacing unreliable, slow build processes and stale skill sets. Agile delivers desired functionality and no more, delighting the customer while improving Quality, Cycle time, Productivity and Predictability.

Evren explained how the healthcare enterprise is made up of numerous functional “silos,” that engage patients – Doctor’s office, pathology, Radiology, Imaging center, OR, Outpatient center, Labor and Delivery and Administration. A rapidly changing environment requires decision makers to rethink business models and how we deliver software. We must deliver the desired functionality, but no more. He quoted a Standish* study that found that only 20% of features are often or always used and 64% of functionality is rarely or never used. The “AGILE” approach uses four “pillars” –

UX – User Experience
SCRUM – Team structure and project management
XP – Extreme Programming
LEAN – Value stream mapping and continuous improvement

Product categories traditionally distinct, with wide geographic variation are quickly shifting due to market need for connected, intelligent workflows with context aware leverage of data proliferation. Strategic tenets suggest the adoption of Agile and learning by nature (practices). We need to

Continue to discover and develop visual and data relationships as well as adjacencies that drive decisions for patient health and at all levels of enterprise management.

- Make greater use of made-in-GE technology by connecting devices with the Cloud strategy.
- Expand strategic relationships to reach across the global healthcare eco-system.

Evren delivered numerous examples and statistics for changing trends and strategies on topics including quality metrics, bed management, asset utilization in complex care, integrated care, chronic disease management, cloud services, specialist productivity.

Evren described the concept of “technical debt” – the notion of operating with non-lean methodologies of the past – lack of automated tests, poorly factored code, obsolete technology, unreliable, incomplete and slow build processes, stale skill sets. We need to “pays-as-we-go,” not let the debt build up.

*Standish is a think tank/research organization that publishes software development data on cost and schedule overruns, limited functionality and other issues of interest to the software industry.
Agile Software Development for Medical Devices  
Andy Sorrentino, Director Quality Engineering  
GE Healthcare Information Technologies

IN 50 WORDS OR LESS:

Agile software development practices have been around for a long time. Healthcare Information Technology developers have been slow to adopt Agile development practices due to concerns that they could not be used in a manner that would demonstrate compliance with medical device regulations. This presentation will provide understanding of Agile software development practices, FDA Design Controls, and IEC62304. It will provide pragmatic insight and detailed understanding of how Agile software development methodologies can be used to meet the requirements of both Design Controls and IEC62304.

GE Healthcare is a large enterprise comprised of Healthcare Systems, Healthcare IT, Medical Diagnostics, Surgery, Life Sciences, and Performance Solutions. Andy discusses how “Agile” replaced “waterfall” in the IT segment and does so with regulatory oversight.

Medical device regulators include the FDA in the US, Medical Device Directive in the EU, and emerging standards in Latin America and the Middle East. Some of the governing regulations are:

- ISO 13485 – Medical Devices Quality Management Systems
- ISO 14971 - Risk Management of Medical Devices
- IEC 62304 - Medical Device Software -life cycle processes
- IEC 62366 - Usability Engineering for Medical Devices

Andy contrasted traditional waterfall involving various iterations of formal design releases with the iterative builds of the agile process. He demonstrated how Regulatory Filings take place throughout the agile process as opposed to at the end of the waterfall process. Some of the pertinent concepts used to address Regulatory issues are:

- Objective evidence that incremental work does not affect safety
- Approved plan prior to execution required
- Architecture, design, requirements reviewed and approved
- Test cased with expected results reviewed and approved
- Objective evidence of complete test coverage.
- All requirements verified
- Validated to meet intended use and fulfill user needs.
- Test results document with objective evidence of pass/fail

Andy described how Design reviews differ in Agile from Design review in waterfall. Basically Design inputs are developed incrementally through “Epics” User stories’ and “sprints”. Stories morph into requirements. In waterfall, test cases are developed and approved and conducted after development of the code. In Agile, requirements and test cases are approved prior to execution. The beauty of agile is the efficiency. By developing many frequent incremental builds with strong customer on-the-spot collaboration and feedback, enormous saving in time and effort are realized.
In 50 words or less: SCRUM is defined as a pre-defined development lifecycle based on AGILE principles. SCRUM can be mapped to CMMI Level 2 and 3 practices, but not completely. A group can easily use SCRUM and CMMI together. An organization at level 2 or 3 could adopt SCRUM as an additional life cycle choice.

CMMI is a collection of practices that an organization can adopt to improve its performance. Maturity level two and three can be mapped to SCRUM, but not completely. SCRUM develops software in 2-4 week sprints, demanding/expecting certain functionality to emerge from each sprint. Each “Sprint” can enact some form of the R-D-C-I life cycle (Requirements, Design, Code, Integrate).

AGILE/SCRUM Principles include:

- Early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development.
- Delivering working software frequently
- Business people and developers must work together daily.
- Build projects around motivated individuals.
- Face-to-face communication is essential.
- Working software is the primary goal.
- Simplicity is achieved through efficient use of resources pair-programming, close customer collaboration.

In CMMI, Level 1 is an amorphous entity. Visibility into the project's process is limited. Management controls are at defined intervals. SCRUM covers approximately 47% of level 2 practices.

Neil provided a detailed map of SCRUM vs. CMMI practices. Burn-down charts in SCRUM can be very similar to those in CMMI or waterfall, instead of measuring releases, milestones or blocks, you measure “sprints”.

The following Level 3 practices are not readily implemented in SCRUM:

--Organizational Process Focus
– Organizational Process Definition
– Organizational Training
– Integrated Project Management
– Risk Management
– Decision Analysis and Resolution
– Generic Goal 3 (i.e., using an organization-wide and tailored process with measurements and lessons-learned)

The Pros and Cons of SCRUM are:

PRO:

- 2-4 week sprint cycles create “Momentum”
- SCRUM can be learned and used in 2 days.

CON:

- Speed can be mistaken for progress
- No architecture; you could implement yourself into a corner.
- Applying SCRUM to large projects takes extra work.

In summary, Scrum is a good implementation for many of the practices in Level 2. A group can easily use Scrum and CMMI together. All the remaining practices in Levels 2 and 3 can be implemented while using Scrum. An organization at Level 2 or 3 could adopt Scrum as an additional lifecycle choice.
Software Assurance is the level of confidence that software is free from vulnerabilities, either intentionally designed into software or accidentally inserted at anytime during its lifecycle and that the software functions in the intended manner. Vulnerability breaches continue to rise with a peak of over 8,000 breaches in 2010. (IBM X-Force).

Andy defined four qualities or attributes needed for software quality and security assurance:

**Reliable** – Software functions as intended with reduced need to update due to lost application availability or data integrity issues.

**Resilient** - Applications are flexible and elastic, able to handle change with demand quickly and without user noticeable impact ensuring system availability.

**Robust** – Applications are able to respond to high user demand, scheduled and unscheduled outages, and malicious attack such as Distributed Denial of Service.

**Secure** - Features and functions that protect the Confidentiality, Integrity and Availability of the data are built into the entire application.

Sources of software defects from data supplied by IBM, SRP Corp., TRW, MITRE, and Nippon Electric Corp., show design errors in the 45-64% range and coding errors in the 25 – 30% range. Andy postulates that there are several key reasons why secure software development is still an issue.

- Market Incentives – Rewards for being first to market, not most secure.
- Complexity and Code Heritage – Building off of legacy versions results in lack of developer insight into complete code base and greater complexity.
- Lack of Tools adoption - Tools to aid developers in examining security of the code only now beginning to enter market consciousness.
- Lack of code quality metrics

There is a persistent gap between customer demands for quality code and secure code. Customers view quality code different from secure code. Common Weakness Enumeration (CWE) is a standardized list of common weaknesses in software architecture, design,
code or implementation that can lead to exploitable security vulnerabilities. The list of CW is comprised of over 900 entries with 681 listed as current weaknesses.

An example of CWE:

Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the back-end database, possibly including execution of system commands.

Other areas of weakness include:

- Coding Standards Violation
- Improper Access of Indexable Resource ('Range Error')
- Improper Check or Handling of Exception Conditions
- Improper Control of a Resource Through its Lifetime
- Improper Enforcement of Message or Data Structure
- Incorrect Calculation
- Insufficient Comparison
- Insufficient Control Flow Management
- Protection Mechanism Failure
- Use of Insufficiently Random Values

Things you can do to mitigate risks:

- Establish and maintain control over all of your inputs and outputs
- Lock down your environment
- Assume that external components can be subverted, and your code can be read by anyone.
- Use industry-accepted security features instead of inventing your own.

Andy concluded with the following surprising statistic:

Of 681 Current Weaknesses from CWE studies*

*Source unspecified.

- Functional 467 (69%)
- Security 214 (31%).

About the Author:

Mike Kress is an Associate Technical Fellow within Boeing Commercial Airplanes Global Partners Procurement QA. He has over 37 years experience in military and commercial aviation hardware and software. He has written guidebooks for the U.S. Air Force on trainer and simulator software. He holds a Bachelor’s degree in Electrical Engineering is a Fellow member of ASQ and holds ASQ CQE and CSQE certifications and is a Registered Professional Engineer. He has led several Boeing and industry advisory groups that have written or contributed toward software standards, most notably RTCA/DO-178B and AS9006. He was chair of the AAQG Project 60 team drafting AS9115, the new international aerospace software quality system standard. He is a former ASQ Software Division Regional Councilor and past chair of the ASQ Software Division. He is a member of the U.S. Technical Advisory Group to ISO/IEC TC176 SC7 and is co-editor of ISO standards on COTS software and Data Quality. He is an RAB/QSA registered QMS and Aerospace Industry Experience Auditor.
1. **Answer A is correct.** Brainstorming is a team tool for generating lots of creative ideas or suggestions in a short period of time. The affinity diagram tool is used to organize ideas or items into significant categories. Storyboards are used to in requirements engineering to describe user interfaces through pictorial sequences. The force field analysis tool is used to identify driving forces that help move towards reaching a goal and restraining forces inhibiting movement towards a goal. **Reference Page(s):** Certified Software Quality Engineering Handbook by Linda Westfall – page 55. **CSQE Body of Knowledge Area:** I.E.2

2. **Answer C is correct.** During the analyze step in the DMAIC model, statistical tools are used to Analyze the data from the measure step and the current process to fully understand the influences that each input variable has on the process and its results. Gap analysis is performed to determine the differences between the current performance of the process and the desired performance. Based on these evaluations the root cause(s) of the problem and/or variation in the process are determined and validated. The objective of the analyze step is to understand the process well enough that it is possible to identify alternative improvement actions during the improve step. **Reference Page(s):** The Certified Software Quality Engineering Handbook by Linda Westfall – page 95. **CSQE Body of Knowledge Area:** II.B.2

3. **Answer A is correct.** In the Yourdon/DeMarco notation for data flow diagrams, a rectangle is the symbol used for an entity external to the system. In this example diagram, the function (labeled Element 3) received the data (labeled Element 2) from the external entity (labeled Element 1) and communicates two data items. One data item (labeled Element 4) is stored in a data repository (labeled Element 5). The other data item (labeled Element 6) is transferred to another process (labeled Element 7). That process (labeled Element 7) then stores a third data item into the data repository (labeled Element 5). **Reference Page(s):** Certified Software Quality Engineering Handbook by Linda Westfall – pages 173-174. **CSQE Body of Knowledge Area:** III.C.3

4. **Answer B is correct.** Since the engineer can only work on the task part time, it will increase the estimated calendar time (duration) that it will take to do the activity. The estimated effort it takes to do the activity would not be affected. **Reference:** The Certified Software Quality Engineering Handbook by Linda Westfall – page 233. **CSQE Body of Knowledge Area:** IV.A.1

5. **Answer D is correct.** Derived measures are measures that are calculated as a function of two or more values of other explicit or derived measures. For example, the defect density of a document could be calculated by a function that divides the number of defects found by the size of the document in pages. Therefore, defect density is a derived measure. Explicit measures, also called base measures or metric primitives, are measurements that are taken directly. For example, we can directly measure effort hours, the number of defects and the number of pages in a document so those are explicit measures. **Reference:** The Certified Software Quality Engineering Handbook by Linda Westfall – pages 313-314. **CSQE Body of Knowledge Area:** V.A.1

6. **Answer A is correct.** Suspension criteria are the specific measurable criteria that when met indicates that the testing activities should be stopped until the resumption criteria are met. For example, if a certain number of critical defects are discovered or if a certain percentage of test cases are blocked, it may no longer be considered cost effective to continue the testing effort until
development corrects the software. The arrival rate of problems reported by testers would be expected to increase, especially during the early stages of system testing. Having “95% of the planned test cases executed and passed” and having “no more than 10 non-closed majors all with workarounds” are examples of possible exit criteria from system testing not a suspension criteria. In addition, if the 10 major problems all have workarounds, then testing can probably continue without significant impact. Reference: Certified Software Quality Engineering Handbook by Linda Westfall – page 404. CSQE Body of Knowledge Area: VI.B.2

7. Answer B is correct. Merging is the configuration management process of taking two or more versions or revisions of a product and combining them into a single new version or revision. Reference Page(s): The Certified Software Quality Engineering Handbook by Linda Westfall – pages 498-499. CSQE Body of Knowledge Area: VII.A.3