

Root Cause Analysis for Customer Reported Problems

Software Quality Consulting Inc.

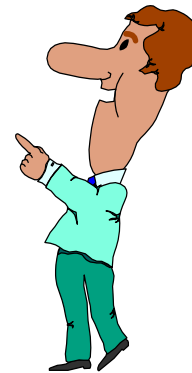
Steven R. Rakitin President	<ul style="list-style-type: none">• Consulting• Training• Auditing
Phone: 508.529.4282 Fax: 508.529.7799	www.swqual.com steve@swqual.com

Copyright © 2008 Software Quality Consulting Inc.

Slide 1

Topics

- **Introduction**
 - Motivation
 - Software Defect Costs
- **Root Cause Analysis**
 - Terminology
 - Tools and Process
- **Integrating RCA into Defect Life Cycle**
 - Triage Team
- **Summary and Action Plan**



Copyright © 2008 Software Quality Consulting Inc.

Slide 2

Motivation

- **Customer Reported Problems (CRPs) are critically important**
 - Represent gaps in knowledge of how customers use software
 - May reflect deficiencies in development and test processes
 - Often lead to disruptive, expensive, unplanned releases
- **CRPs represent opportunities:**
 - To turn potential dissatisfaction into satisfaction
 - To learn more about how customers use your software
 - To identify areas for process improvement...

Copyright © 2008 Software Quality Consulting Inc.

Slide 3

Motivation

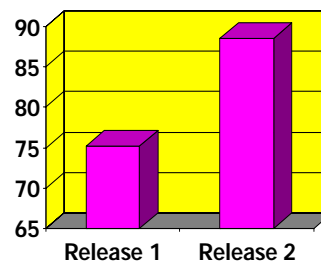
- **How successful are you at finding defects your customers are likely to find?**

$$\frac{\text{Total defects you found}}{\text{Total you found} + \text{Customer-reported defects}}$$

- based on at least n months of Customer use

- **Use of this measure...**

How good a job are we doing of
Act Like a Customer Testing™?



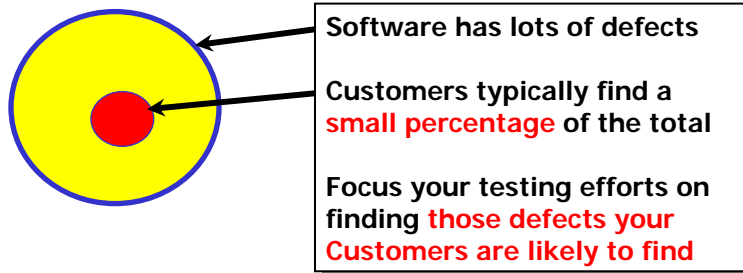
Act Like a Customer Testing is a trademark of Software Quality Consulting, Inc.
Jones, C., "Software defect-removal Efficiency", *IEEE Computer*, Vol. 29, No. 4, April 1996, pp. 94-95.

Copyright © 2008 Software Quality Consulting Inc.

Slide 4

Motivation

- **Act Like a Customer Testing™**
 - Testers need **domain knowledge** to be effective
 - Write tests based on customer use in their environment



Act Like a Customer Testing is a trademark of Software Quality Consulting, Inc.

Copyright © 2008 Software Quality Consulting Inc.

Slide 5

Root Cause Analysis

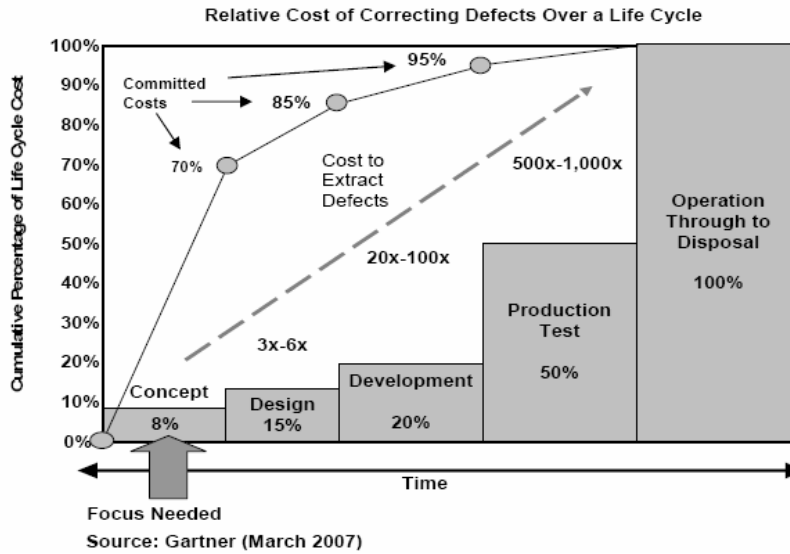
- **Root Cause Analysis (RCA)** helps:
 - understand causes of **customer dissatisfaction**
 - reduce expensive rework by **preventing recurrence**
 - identify **process weaknesses**
 - improve **customer satisfaction**
- **RCA can provide answers to:**
 - **What happened?**
 - **Why did it happen?**
 - **How did we miss it?**
 - **What can we do to prevent it?**



Copyright © 2008 Software Quality Consulting Inc.

Slide 6

Software Defect Costs



Copyright © 2008 Software Quality Consulting Inc.

Slide 7

Software Defect Costs

- A recent study commissioned by National Institute of Standards and Technology found that **defective software costs US economy \$60 billion annually**
- Are you measuring your defect costs?



The Economic Impacts of Inadequate Infrastructure for Software Testing, NIST Planning Report 02-3, May 2002

Copyright © 2008 Software Quality Consulting Inc.

Slide 8

Software Defect Costs

Pre-release Find/Fix Cycle

Cycle can take from **10-30 hours per defect**

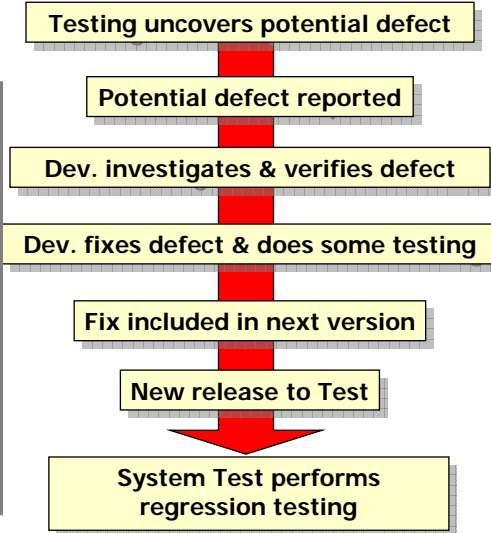
Use **\$150** as **fully loaded labor cost** for Engineering time...

Cost per defect is:

$$30 \times \$150 = \$4,500$$

For 100 defects:

$$3,000 \times \$150 = \$450,000$$



Copyright © 2008 Software Quality Consulting Inc.

Slide 9

Software Defect Costs

Post-release Find/Fix Cycle

Cycle can take **20-60 hours per defect**

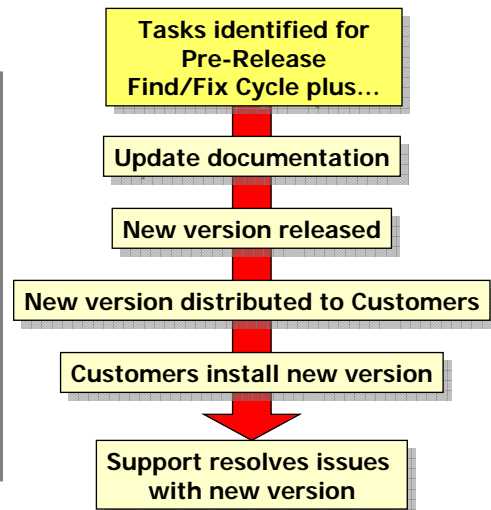
Use **\$150** **fully loaded labor cost** for Engineering time...

Cost per defect is:

$$60 \times \$150 = \$9,000$$

For 100 defects:

$$6,000 \times \$150 = \$900,000$$



Copyright © 2008 Software Quality Consulting Inc.

Slide 10

Software Defect Costs



Dr. Harlan Mills
IBM Fellow

“Programs do not acquire bugs as people acquire germs, by hanging around other buggy programs. Programmers must insert them.”



Copyright © 2008 Software Quality Consulting Inc.

Slide 11

Software Defect Costs

- **Reported Defect Injection Rates for a sample of 810 experienced software engineers:**

Group	Avg. no. defects injected per (KLOC)
All	120.8 KLOC = 1 defect per 8 LOC
Upper Quartile	61.9
Upper 10%	28.9
Upper 1%	11.2

- **Software is released with some known defects and a significant number of unknown defects**

Humphrey, W., "The Quality Attitude", *news@sei newsletter*, Number 3, 2004.

Copyright © 2008 Software Quality Consulting Inc.

Slide 12

Software Defect Costs

- Please try this at work:

$$\frac{\text{Defects injected} - \text{Defects found}}{\text{Estimated no. of unknown defects}}$$

where: **defects injected** = size (KLOC) X 120.8



Copyright © 2008 Software Quality Consulting Inc.

Slide 13

Software Defect Costs

- A simple example...
- One million LOC = 1,000 KLOCs
 - Avg. defect injection rate of 120 defects/KLOC
 - **120,000** defects injected
 - Assume 95% found = **114,000** defects found
- **Unknown defects** = defects injected – defects found
 - = (120,000 – 114,000)
 - = **6,000**



Copyright © 2008 Software Quality Consulting Inc.

Slide 14

Root Cause Analysis

- Used to investigate **root cause** of major disasters:
 - Airplane crashes
 - Space Shuttle accidents
 - Chemical and nuclear plant disasters
- **RCA** requires effective problem solving skills
- Finding **root cause** may be difficult because:
 - We have an incomplete problem definition
 - **Causal relationships** are unknown
 - We often focus on finding solutions and assigning blame



Gano, D., et. al., [Apollo Root Cause Analysis – A New Way Of Thinking](#), Apollonian Publications, 1999

Copyright © 2008 Software Quality Consulting Inc.

Slide 15

Terminology

- **Event**
 - Any failure of software and services (including code, documentation, installation, customization, training, etc.) that impacts customers
- **Causal Factors**
 - Factors that contribute to occurrence of an **event**
- **Causal Relationships**
 - Cause and effect sequence in which a specific action creates a condition that contributes to or results in an **event**



US Dept of Energy, [Root Cause Analysis Guidance Document](#), DOE-NE-STD-1004-92, February 1992

Copyright © 2008 Software Quality Consulting Inc.

Slide 16

Terminology

- **Corrective Action (CA)**
 - Action to eliminate **root cause** of a **reported problem**
 - **Immediate CA** is taken **soon after problem is reported to help customer recover...**
 - **workaround, hot fix, etc.**
 - **Long Term CA** taken to prevent **recurrence**
 - **results in changes to process and procedures**

Quality Management Systems – Fundamentals and Vocabulary, ISO-9000:2000

Copyright © 2008 Software Quality Consulting Inc.

Slide 17

Terminology

- **Root Cause**
 - Cause that, if corrected, prevents **recurrence** of this and similar **events**
 - **Attributes of root causes:**
 - Represent specific underlying causes of **events...**
 - Can be **reasonably identified...**
 - Can be **fixed by Management...**
 - Lead to effective **corrective actions...**



Rooney, J. and Vanden Heuvel, L., "Root Cause Analysis for Beginners", *ASQ Quality Progress*, July 2004, p. 45-53

Copyright © 2008 Software Quality Consulting Inc.

Slide 18

Terminology

- **Root Causes represent specific underlying causes of events...**
 - Goal is to identify **specific underlying causes**
 - More specific investigation is about why an **event** occurred, easier it is to recommend changes that prevent recurrence
- **RCA Process needs to be reasonable...**
 - Investigation must be **cost-effective**
 - Good **RCA Process** helps keep **ROI high**

Rooney, J. and Vanden Heuvel, L., "Root Cause Analysis for Beginners", *ASQ Quality Progress*, July 2004, p. 45-53

Copyright © 2008 Software Quality Consulting Inc.

Slide 19

Terminology

- **Root Causes can be fixed by Management**
 - Vague classifications such as "operator error", "hardware failure", or "external factors" are not helpful
 - We need to know exactly **why** an **event** occurred before effective **CA** can be taken to **prevent recurrence**
- **Root Causes lead to effective CA**
 - Corrective Actions should directly address identified **root causes**
 - If recommendations are vague -- specific **root cause** was probably not found

Copyright © 2008 Software Quality Consulting Inc.

Slide 20

Terminology

- **Root Cause Analysis (RCA)**
 - Process of investigating, understanding, categorizing **root causes**
 - Performed by **small cross-functional team** as part of **Triage Process**
 - Analysis based on **factual information** obtained from:
 - **Documents and records**
 - **Interviews**
 - **Brainstorming sessions**
 - Use tools such as:
 - **Why Tree**
 - **Pareto Analysis**



Copyright © 2008 Software Quality Consulting Inc.

Slide 21

Tools – Why Tree

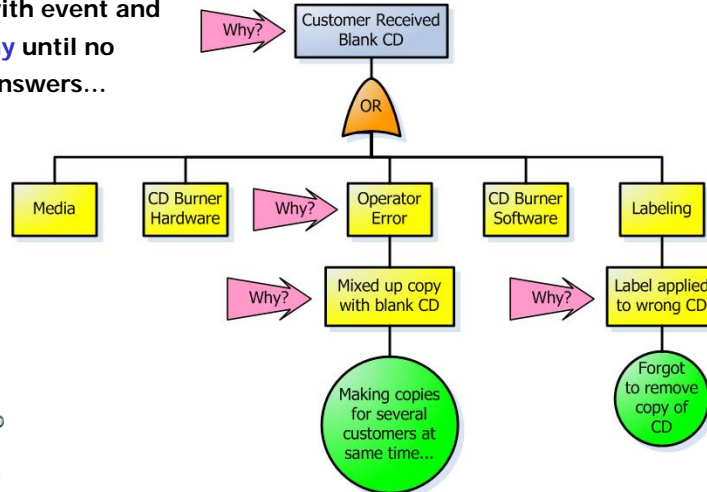
- **Can help identify an appropriate CA...**
 - **What** should be done **immediately** to resolve this **CRP...**
 - **What** should be done **long term** to **prevent recurrence...**
 - What is it about **way we work** that allowed this event to occur?
 - Most **root causes** are found in **way we work...**
- **Start with a specific event** and ask **Why did this happen?**

Copyright © 2008 Software Quality Consulting Inc.

Slide 22

Tools – Why Tree

- Start with event and ask **Why** until no more answers...



Copyright © 2008 Software Quality Consulting Inc.

Slide 23

Immediate Corrective Action

- Use **Why Tree** to help develop an **Immediate CA**
 - Workaround, hot fix, patch, new CD, config changes, ...
 - Implement **CA**
 - Collect data to determine effectiveness with Customer
 - Document **Immediate CA** in **Bug Tracking System**
 - Add results of **RCA** as attachment to **CRP**...
 - Identify effectiveness measures...
 - determines if **CA** resolves problem...
 - ensures that real **root cause** found



Copyright © 2008 Software Quality Consulting Inc.

Slide 24

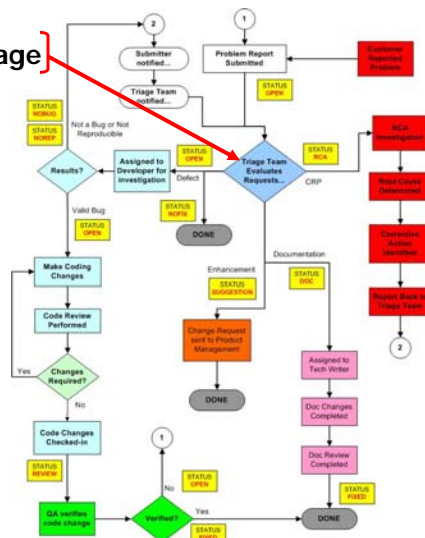
Long Term Corrective Action

- Use **Why Tree** to develop **Long Term CA**
 - Review existing business processes and procedures
 - Identify **process weaknesses** directly related to **root cause**
 - Identify recommendations to **prevent recurrence**
 - Some changes may require Management review and approval...
 - Identify **effectiveness measures**...
 - determine if **long term CA** prevents recurrence
 - **Implement** recommendations...
 - Collect data to determine **effectiveness**...
 - Document **Long Term CA** in **Bug Tracking System**...



RCA Process

- **RCA Process** occurs as part of Triage
- **Triage Team** reviews all **CRPs**
- Consider **RCA** for all **CRPs**
- Triage Team appoints **RCA Team** to investigate...
 - **Support, SQA, Dev**
- Report back to Triage Team...



RCA Process

- **Step 1 - Data Collection**

- Majority of time analyzing events is spent gathering data and information
- Complete information and thorough understanding of events required to identify **causal factors** and **real root causes**
- Begin with accurate statement of what happened **in Customer's own words**
 - Descriptions of **events** in Customer's language is sometimes "filtered"...
 - Data collection will initially be sketchy – use **Why Tree** to identify additional data to collect...

Rooney, J. and Vanden Heuvel, L., "Root Cause Analysis for Beginners", *ASQ Quality Progress*, July 2004, p. 45-53

Copyright © 2008 Software Quality Consulting Inc.

Slide 27

RCA Process

- **Step 1 - Data Collection**

- **Collect general information about Customer:**
 - Is Customer power user or novice?
 - Has Customer received training?
 - Is Customer's use and/or data unique?
 - **More questions?**
- **Collect information about Customer's environment:**
 - Standard release or customer-specific release?
 - Platform/database/operating system releases?
 - Received hot fixes recently? Installed?
 - **More questions?**
- **Customer Support staff can help gather this information**
 - Use checklist of questions to ask when Customer calls

Copyright © 2008 Software Quality Consulting Inc.

Slide 28

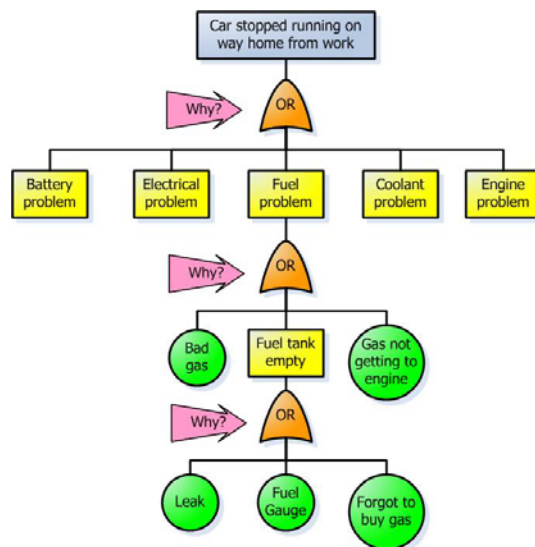
RCA Process

- **Step 2 – Determine What Happened**
 - Start creating a **Why Tree...**
 - Begin with **event** in **Customer's language...**
 - "Application crashed on startup..."
 - Then ask **Why?**
 - Continue asking **Why** until there are no more answers
 - **Process will identify additional information to collect...**
 - Was feature defined in Requirements Spec?
 - Was feature tested? If so, how?
 - Was user training effective?
 - Are there metadata, platform, or configuration issues?
 - **Other questions?**

Copyright © 2008 Software Quality Consulting Inc.

Slide 29

Simple Example



Copyright © 2008 Software Quality Consulting Inc.

Slide 30

RCA Process

- **Step 3 – Identify Immediate Corrective Action**

- Based on info collected, RCA Team identifies an **immediate CA** to resolve customer's immediate problem
- RCA Team also identifies effectiveness checks...
 - Determines if **immediate CA** resolves customer's problem
- **Immediate CA** is implemented...
- RCA Team:
 - collects data from customer to verify **effectiveness**
 - **immediate CA** and other relevant info attached to CRP in **Bug Tracking System...**
 - reports back to **Triage Team** with results and effectiveness...

Copyright © 2008 Software Quality Consulting Inc.

Slide 31

RCA Process

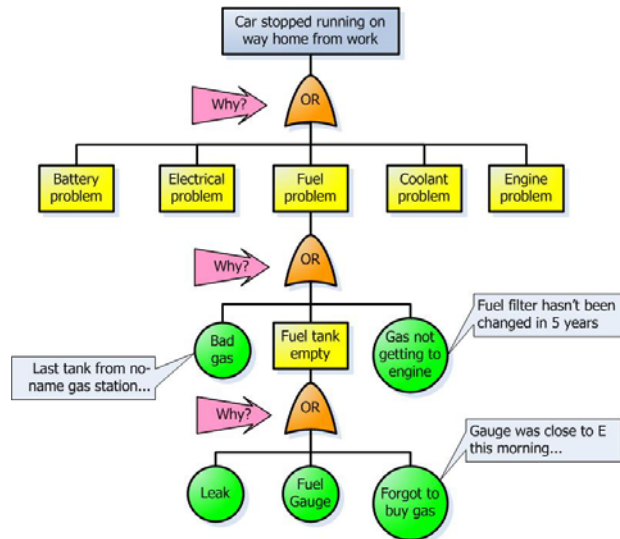
- **Step 4 - Root Cause Identification**

- Based on **Why Tree and supporting information**
- **RCA Team** reviews info and identifies most probable **root causes...**
- Ensure that most probable **root causes** meet criteria:
 - Represent specific underlying causes of **events...**
 - Can be **reasonably identified...**
 - Can be **fixed by Management...**
 - Can lead to effective **corrective actions...**
- **Root cause** documented and results attached to **CRP** in **Bug Tracking System...**

Copyright © 2008 Software Quality Consulting Inc.

Slide 32

Simple Example



Copyright © 2008 Software Quality Consulting Inc.

Slide 33

RCA Process

- **Step 5 – Long Term Corrective Action**
 - Most **root causes** found in **way you work**...
 - Review process and procedures...
 - Long term CA often results in changes to **way you work**
 - Are procedures written? Followed?
 - Unwritten procedures result in inconsistent results
 - Are existing procedures/training ineffective?
 - Are additional procedures/training required?
 - **Effectiveness Measures**...
 - How will you know that **root cause** has been eliminated?

Copyright © 2008 Software Quality Consulting Inc.

Slide 34

RCA Tools – Pareto Analysis

- 80% problems result from 20% causes
- Can help determine what problems to address
- As **root causes** are identified, add them to list...

RC	Root Cause Description
1	Feature was defined but not tested
2	Feature was tested but the test was inadequate
3	Feature was not defined in Functional Spec
4	Feature was defined in Functional Spec but not in Use Cases
5	Design was inadequate/inappropriate - Design review not held
6	Design was inadequate/inappropriate - Design review didn't catch it
7	Coding was inadequate/incorrect - Code review not held
8	Coding was inadequate/incorrect - Code review didn't catch it
9	Installation / configuration issues...
10	Metadata issues...
11	Environment / Version compatibility issues...
12	User training issues...

Copyright © 2008 Software Quality Consulting Inc.

Slide 35

RCA Tools – Pareto Analysis

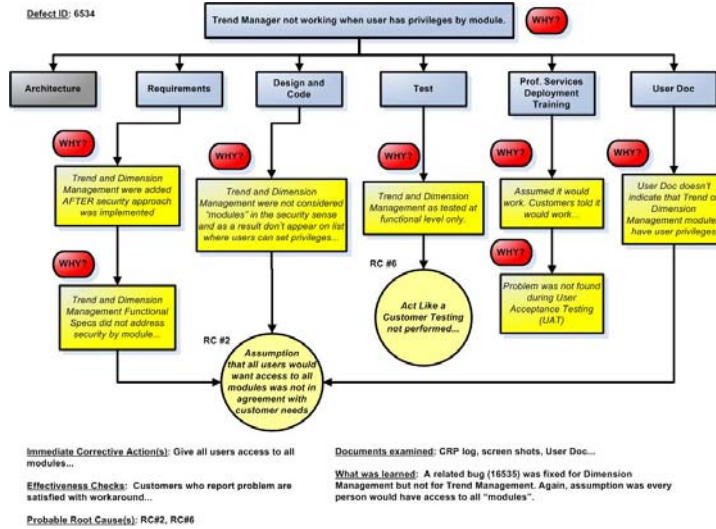
Use Pareto Analysis to identify root causes that warrant long term corrective action

CRP #	RC #1	RC #2	RC #3	RC #4	Etc...
403	✓		✓		
508		✓	✓	✓	
990		✓			
1112	✓	✓		✓	
1133		✓			
1244					
1456		✓			
1789		✓			
2134					
2367		✓			
Totals	2	7	2	2	

Copyright © 2008 Software Quality Consulting Inc.

Slide 36

Real Example



Copyright © 2008 Software Quality Consulting Inc.

Slide 37

Summary

- **RCA Process...**
 - Can be very effective at discovering **real root causes**
 - Helps identify **WHAT, WHY, and HOW**
 - Leads to **immediate CA and long term CA**
 - Improves Customer Satisfaction
 - Reduces rework and eliminates unplanned releases
 - Fits within **typical Defect Life Cycle Process**
 - Performed by **Triage Team** with support from staff
 - Includes **effectiveness measures** to determine if **CA** is effective

Copyright © 2008 Software Quality Consulting Inc.

Slide 38

Additional Workshops

- Software Development for Medical Device Manufacturers
- Peer Reviews and Inspections
- Computer System Validation
- Risk Management
- Writing and Reviewing Requirements for Software
- Software Verification & Validation
- 21 CFR Part 11: Electronic Records and Electronic Signatures
- Process Validation
- **For more information, please visit www.swqual.com**

Copyright © 2008 Software Quality Consulting Inc.

Slide 39

Additional Workshops

- Project Retrospectives
- Root Cause Analysis for Customer Reported Problems
- Writing Software Requirements
- Estimating and Scheduling Best Practices
- Software Verification & Validation for Practitioners and Managers
- Accurate Schedules Using the Yellow Sticky Method
- Predictable Software Development TM
- Peer Reviews and Inspections
- Improving the Effectiveness of Testing
- Risk Management for Embedded Software Development
- **For more information, please visit www.swqual.com**

Predictable Software Development is a trademark of Software Quality Consulting, Inc.

Copyright © 2008 Software Quality Consulting Inc.

Slide 40

Thank you...

- If you have questions, please call or e-mail...
- **Subscribe to my e-newsletter...**  The logo for 'Food for Thought' features the text in a red, handwritten-style font, flanked by two circular icons containing a stylized '@' symbol.
- For a free subscription and to view past newsletters, visit www.swqual.com

Software Quality Consulting Inc.	
Steven R. Rakitin President	<ul style="list-style-type: none">• Consulting• Training• Auditing
Phone: 508.529.4282 Fax: 508.529.7799	www.swqual.com steve@swqual.com