Root Cause Analysis for Customer Reported Problems

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
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...

Motivation

- **How successful are you at finding defects your customers are likely to find?**
  
  \[
  \text{Total defects you found} = \frac{\text{Total you found + Customer-reported defects}}{n} \]

  - 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 TM?**

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

Motivation

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

Software has lots of defects
Customers typically find a **small percentage** of the total
Focus your testing efforts on finding **those defects** your Customers are likely to find

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

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?
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
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 x $150 = $4,500
- For 100 defects:
  - 3,000 x $150 = $450,000

**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 x $150 = $9,000
- For 100 defects:
  - 6,000 x $150 = $900,000

Copyright © 2008 Software Quality Consulting Inc.
Software Defect Costs

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

Dr. Harlan Mills
IBM Fellow

Software Defect Costs

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

<table>
<thead>
<tr>
<th>Group</th>
<th>Avg. no. defects injected per (KLOC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>120.8 KLOC = 1 defect per 8 LOC</td>
</tr>
<tr>
<td>Upper Quartile</td>
<td>61.9</td>
</tr>
<tr>
<td>Upper 10%</td>
<td>28.9</td>
</tr>
<tr>
<td>Upper 1%</td>
<td>11.2</td>
</tr>
</tbody>
</table>

• 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.
Software Defect Costs

• Please try this at work:

\[
\begin{align*}
\text{Defects injected} &= \text{size (KLOC)} \times 120.8 \\
- \text{Defects found} &= \text{Estimated no. of unknown defects}
\end{align*}
\]

where: \( \text{defects injected} = \text{size (KLOC)} \times 120.8 \)

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
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


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

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*

---

Terminology

• **Root Cause**
  - Cause that, if corrected, prevents *reurrence* 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*...
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


---

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
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

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?**
Tools - Why Tree

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

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
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, SOA, 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...

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?

Simple Example

- Car stopped running on way home from work
  - Battery problem
  - Electrical problem
  - Fuel problem
  - Coolant problem
  - Engine problem
- Battery is dead
- Fuel tank empty
- Gas not getting to engine
- Leak
- Fuel gauge
- Forgot to buy gas
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...

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...
**Simple Example**

![Simple Example Diagram]

**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?
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...

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

RCA Tools – Pareto Analysis

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

<table>
<thead>
<tr>
<th>CRP #</th>
<th>RC #1</th>
<th>RC #2</th>
<th>RC #3</th>
<th>RC #4</th>
<th>Etc...</th>
</tr>
</thead>
<tbody>
<tr>
<td>403</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>508</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>990</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1112</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>1133</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1244</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1456</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1789</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2134</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2367</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td><strong>2</strong></td>
<td><strong>7</strong></td>
<td><strong>2</strong></td>
<td><strong>2</strong></td>
<td></td>
</tr>
</tbody>
</table>
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
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](http://www.swqual.com)*

---

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™
- Peer Reviews and Inspections
- Improving the Effectiveness of Testing
- Risk Management for Embedded Software Development

*For more information, please visit [www.swqual.com](http://www.swqual.com)*

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

• If you have questions, please call or e-mail...

• Subscribe to my e-newsletter...

• For a free subscription and to view past newsletters, visit www.swqual.com

Software Quality Consulting Inc.

Steven R. Rakitin        Consulting
President                  Training
Phone: 508.529.4282         Auditing
Fax: 508.529.7799          www.swqual.com

steve@swqual.com