Value-added Feedback - The Missing Link in Software Testing

Randall W. Rice, CTAL
www.riceconsulting.com
Becoming an Information Provider

- Traditionally, testers have been “problem finder”.
- The more positive and value-added view is to be an information provider.
What’s In This For You?

• An affordable and easy way to:
  • Build your credibility
  • Promote a positive view of testing
  • Help guide your testing projects
  • Be helpful to project management
  • Add value to projects
  • Start improving processes and methods
  • Be seen as more than a commodity
Four Ways You Can Add Value

- Ending the cycle of defects
- Showing information with dashboards
- Being the keeper of information
- Learning from failure
Development and Testing Methodologies

- Are typically not the main solution to build better software!
  - The magic is not in the methods!
  - If so, why do we keep trying different “silver bullets”?

- There is a relationship, just not the deeper solution.

- This is actually good news, because you can apply these ideas regardless of your methods.
“If people don’t know what they want, no development process - no matter how exact, how clever, or how efficient - will satisfy them.”

Gerald Weinberg
Methods can solve similar, lower complexity problems. This gives a sense of relief, but leaves difficult and complex problems which grow and are not solved by the same methods.

Source: Exploring Requirements by Weinberg and Gause
Ending the Cycle of Defects

- Defects = Rework = Lost $$
- The cycle continues until broken

Same types of defects

Find ➔ Fix
The Feedback Loop

Create → Find → Fix

New types of defects

Identify root causes and fix the process
Quick Wins

- Pick 5 high severity defects
- Identify key causes
- One by one, work to fix the process to prevent similar occurrences
- Test to make sure the improvements work
- Move on 5 more problems
The Payoff

- Rework: 25%
- Testing: 15%
- Prevention: 5%

Savings: 15%
Rework: 10%
Testing: 10%
Prevention: 10%
What’s Needed?

- Measurements and metrics
  - Where defects originate
  - Which types of defects are most prevalent
  - Why defects occur
  - Degree of improvement

- A meaningful way to show metrics
- A culture of learning and improvement
- The will and support to take action
The Main Objective of Testing Dashboards

- To provide simple, meaningful and reliable information in one place to help guide the testing effort and convey that information to our clients.
A Project is Like Taking a Trip

- You need to know your:
  - Destination
  - Current location
  - Orientation (Direction)
  - Trip progress
  - Speed
  - Resource levels (gas, oil, etc.)
  - Engine operation (temp, charge, etc.)
The Goals

- Arrive at the desired destination safely
- Stay on the road
- Make good progress
- Don’t get lost
- Don’t run out of fuel
- Only one driver at a time
Key Components

- An effective testing strategy
  - Defines the test objectives, scope and approach early in the project.

- A workable test plan
  - Defines scope, resources, schedules, risks, contingencies, etc.

- A dashboard
  - Monitors defect levels, test progress, resource levels.
Dashboards

- Dashboards are not new
  - They have been a common topic in articles and at conferences for several years.
- At the same time, testers often struggle with how to convey accurate and timely information to management.
- So...let’s explore dashboards and look at some examples.
- Then, we’ll look at the issues behind test measurement and reporting.
Example: Your Car’s Dashboard

- Car dashboards tell you current:
  - Trip progress (speedometer and odometer)
  - Resources (fuel)
  - Car status (temp, oil pressure, charging, engine performance)
  - Warnings (seat belt, open doors)
What Makes a Good Metric?

- Simple
  - Can be easily measured and understood
- Can be automated
  - So we don’t have to take readings manually
  - Also, people don’t get the chance to manipulate the numbers
- Meaningful
  - We can gain useful information to make decisions
Why Have a Testing Dashboard?

- For fast and easy reporting test results to management
- To have all of your testing information in one place
- To help guide the testing effort
- To help make good decisions
- To build project learning
  - Better estimates in the future
- To build the credibility and visibility of testing
What is Required for a Dashboard?

• Accurate and meaningful measurements and metrics
  • Plus...a clear understanding of what the metrics mean.

• A culture of trust and openness

• Non-intrusive ways to measure
  • Ideally, the measures should come from activities already being tracked.
    • Defect tracking systems
    • Project management software
What is Required for a Dashboard? (2)

- A way to display the information in ways that are:
  - Understandable
  - Easy to Build and Maintain
  - Accessible
  - Integrated with tools
    - Issue tracking
    - Test management
    - Project management
What is Shown on a Typical Testing Dashboard?

- Test Coverage
  - Requirements
  - Functional
  - Test case
  - Code

- Test Status
  - Testing
  - Defect resolution
  - Readiness for deployment
  - Pass/Fail

- Progress
  - Based on test goals and objectives
  - Blockages

- Risk
  - Technical
  - Business
  - Project

- Defects
  - Categories
  - Trends
  - Detection Percentage
  - Resolution Status

- Testware
  - Completion %
  - Automation %

- Resources
What Should You Show?

• Ask your customers!
  • What information do they value?
  • How do they need it shown?
  • When do they need it?
  • How timely must it be?

• However, just like in obtaining user requirements, people often don’t know what they want or need until they actually see it.
  • Start with a prototype
Things to Consider When Designing Dashboard Items

- Purpose of the Dashboard
- Chart or Graphic Types
- Colors
- Positions
- Brightness
- Orientation
- Sizes
- Shapes
A “Not So Good” Example
Percentage of Analytic Computer Usage by Type

- **Extended Enterprise User**
  (Extranets/B2B/B2C and mobile/wireless)

- **Casual User**
  (Dashboards and enterprise reporting)

- **Business User**
  (Scorecards, performance mgmt, business reporting, and packaged apps)

- **Power User**
  (Statistical analysis, analytical reporting and OLAP)

- **IT User**
  (Application development, data mining and meta data design)

Source: Giga Research, a wholly-owned subsidiary of Forrester Research, Inc.

http://www.perceptualedge.com
### Sample Project Quality Dashboard

#### Post-Implementation Defects By System

<table>
<thead>
<tr>
<th>System</th>
<th>Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting</td>
<td>28%</td>
</tr>
<tr>
<td>Sales</td>
<td>17%</td>
</tr>
<tr>
<td>Payroll</td>
<td>7%</td>
</tr>
<tr>
<td>HR</td>
<td>10%</td>
</tr>
<tr>
<td>Inventory</td>
<td>6%</td>
</tr>
<tr>
<td>CRM</td>
<td>6%</td>
</tr>
<tr>
<td>ERP</td>
<td>6%</td>
</tr>
</tbody>
</table>

#### Top Eight Outstanding Risks

<table>
<thead>
<tr>
<th>Risk, System</th>
<th>Risk Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 User Training, Accounting</td>
<td>4</td>
</tr>
<tr>
<td>2 Usability, Web Site</td>
<td>4</td>
</tr>
<tr>
<td>3 Security, Web Site</td>
<td>4</td>
</tr>
<tr>
<td>4 User Involvement, Inventory</td>
<td>4</td>
</tr>
<tr>
<td>5 Lack of Resources, HR</td>
<td>3</td>
</tr>
<tr>
<td>6 Inadequate User Requirements, Sales</td>
<td>3</td>
</tr>
<tr>
<td>7 Insufficient test schedule, Finance</td>
<td>3</td>
</tr>
<tr>
<td>8 Vendor dispute, CRM</td>
<td>3</td>
</tr>
</tbody>
</table>

Data Current As Of: 2/19/2009 20:59

---

### Quality Attribute Levels

#### Test Completion % by

<table>
<thead>
<tr>
<th>System</th>
<th>Completion %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing</td>
<td>93</td>
</tr>
<tr>
<td>Sales</td>
<td>81</td>
</tr>
<tr>
<td>Payroll</td>
<td>52</td>
</tr>
<tr>
<td>HR</td>
<td>31</td>
</tr>
<tr>
<td>Inventory</td>
<td>22</td>
</tr>
</tbody>
</table>

#### DDP History

<table>
<thead>
<tr>
<th>Release</th>
<th>Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>76</td>
</tr>
<tr>
<td>2</td>
<td>71</td>
</tr>
<tr>
<td>3</td>
<td>69</td>
</tr>
<tr>
<td>4</td>
<td>78</td>
</tr>
<tr>
<td>5</td>
<td>83</td>
</tr>
</tbody>
</table>

#### Defect Types

- Faults: 76, 71, 69, 78, 83
- Defects: 40, 22, 12, 8, 0
- Documentation: 70, 40, 22, 12, 8
- Usability: 80, 60, 40, 20, 0

#### Quality Attribute Levels
What Would it Mean...

• To your project managers to have access to this type of information at any point in time?

• To the senior management in your company to see overall software quality information?

• To your career to be seen as the keeper of this kind of information?
Words of Warning

- Too many items on a dashboard can be distracting and confusing.
  - Unless you are flying a plane!
- Metrics can be abused.
  - If people don’t understand human behavior, more harm than good can result.
- Stuff happens.
  - Things not shown on your dashboard can derail your test.
Words of Warning (2)

• With dashboards, everyone can see the same information at the same time.
  • This may be an issue if you don’t want to show someone information until you have had a chance to see it first.

• Manual input to the dashboard gets overwhelming.
  • You want to automate the data capture as much as possible.

• Dashboards may be too general for some managers.
Keeping the Process Working

• The dashboard tells you about vehicle (process) malfunctions.
  • In testing, the process is the engine.
    • The process might not be documented.
    • How you perform the process determines whether or not you reach the intended destination.
Learning from Failure

- The greatest value of a defect is to learn from it.
- This requires a culture of trust and openness.
- Methods include measurements, retrospectives and root cause analysis.
In testing, failure is not an option - it’s an objective!
Learning from Active Feedback and Failure

- Learn
- Fail
- Adjust
- Risk
Process Improvement Without Making a Big Deal Out of It

• Most people who improve do so without a major model or push.

• All it takes is asking and acting on a few questions to your customers and team.
Questions for Customers

• Are we meeting your expectations?
  • How or why not?

• What else can we do to improve our service to you?

• Is there a particular team member that has been especially helpful to you?
Questions for Team Members

- What do we do best?
- Which things are broken?
- Which things are declining?
- What can I do better as a leader to fix the problem?
- What can we address and improve in the short-term?
- What will take longer?
- Which additional resources will we need?
Becoming an Agent for Change in Your Organization

- Starting Small – The “Skunkworks” Approach
- “Planned Organic”
- Celebrating and Rewarding Success
- Creating Your Action Plan
  - Identify Your Goals
  - Short-Term
  - Mid-Term
  - Long-Term
Final Thoughts

- A key purpose of testing is to provide meaningful information to management to make informed decisions.
- This is a positive value-added view of testing.
- Dashboards are one more tool to help you guide your testing project, but they don’t drive the car!
Final Thoughts (2)

- Good dashboards have:
  - Good design
  - Current information
  - The right metrics for your situation
    - They should reflect the job at hand
    - They should be understandable
  - Meaning and value for the readers
  - Interpretation and guidance
    - Annotations are helpful
Bio - Randall W. Rice

- Over 35 years experience in building and testing information systems in a variety of industries and technical environments
- Certified Software Quality Analyst
- Certified Software Tester
- ASTQB Certified Tester - Foundation level, Advanced level (Full)
- Director, American Software Testing Qualification Board (ASTQB)
- Chairperson, 1995 - 2000 QAI’s annual software testing conference
- Co-author with William E. Perry, *Surviving the Top Ten Challenges of Software Testing* and *Testing Dirty Systems*
- Principal Consultant and Trainer, Rice Consulting Services, Inc.
Contact Information

Randall W. Rice, CTAL
Rice Consulting Services, Inc.
P.O. Box 892003
Oklahoma City, OK 73170
Ph: 405-691-8075
Fax: 405-691-1441
Web site: www.riceconsulting.com
e-mail: rrice@riceconsulting.com