Measuring the Reliability and Value of a Checklist

Dan Houston, Ph.D.
July 26, 2007
Problem

• Software developers rely heavily on natural language instruments.
  - Task names and descriptions
  - Checklists and questionnaires
  - Development procedures
  - Defect categories

• Have you ever ...
  - spent more time than necessary charging your time because task names didn’t cover your work?
  - had to skip a checklist item because you couldn’t figure out what it was asking for?
  - been unable to perform a procedure the way it was written?
  - argued over defect classification because the category descriptions weren’t clear?

• Questions
  - Can we measure the reliability of natural language instruments such as a checklist?
  - How can the problematic items be identified accurately?
Outline

• Specific problem: defects leaking through test phase due to poor test specifications
• Context: a process improvement project
• The process analysis
• Process improvements
  - Producing a checklist
  - Measuring checklist reliability
  - Identifying checklist items to be improved
  - Improving checklist
• Checklist validation and project savings
• Other software development applications of subjective measurement system evaluation
DMAIC: Process Improvement

• Analyze and measure process for variation
  - Uses qualitative and quantitative, especially statistical, tools.
    ◆ Subjective measurement system evaluation (MSE)
  - Categorize inputs to process steps
  - Statistically characterize variation in process outputs
• Identify improvement opportunities
• Implement improvements and measure savings

<table>
<thead>
<tr>
<th>Phase</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define</td>
<td>Identify an opportunity and define a project to address it.</td>
</tr>
<tr>
<td>Measure</td>
<td>Analyze the current process and specify the desired outcome.</td>
</tr>
<tr>
<td>Analyze</td>
<td>Identify root causes and proposed solutions.</td>
</tr>
<tr>
<td>Improve</td>
<td>Prioritize solutions; select, plan, validate, and implement solution.</td>
</tr>
<tr>
<td>Control</td>
<td>Develop a plan for measuring progress and maintaining gains.</td>
</tr>
</tbody>
</table>
Test Specifications Project

- Context: Fagan-style inspections of all work products
- System testers realized the need for guidance in reviewing test specifications.
  - Lack of content guidance caused concern about specification incompleteness.
  - Were defects passing through the system test phase?
- Project focus: test specification process
  - Emphasis on the quality of test specification content.
  - No savings were anticipated, but as the project progressed, the project team saw an opportunity to measure savings from use of the checklist.

<table>
<thead>
<tr>
<th>Define</th>
<th>Measure</th>
<th>Analyze</th>
<th>Improve</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem statement</td>
<td>As-is process map</td>
<td>Failure modes and effects analysis (FMEA)</td>
<td>To be process map</td>
<td>Control plan Results</td>
</tr>
<tr>
<td>SIPOC (supplier, inputs, process, outputs, customers)</td>
<td></td>
<td></td>
<td>Checklist drafted and reliability measured</td>
<td></td>
</tr>
</tbody>
</table>
SIPOC Diagram

**Suppliers** (Providers of the required resources)

- Product architect
- Development teams
- SSEPG
- Test engineers

**Inputs** (Resources required by the process)

- Requirements
- Functional concepts
- Functional specification
- Participation in Reviews
- Documented practices and/or application notes

**Process** (Top Level description of activity)

- Testable requirements
- Identification of primary intended usage
- Clear & complete description of user interfaces
- Description of limitations & expected error conditions
- Detailed knowledge of intended behavior
- Sufficient time spent
- Template and guidance for efficient & effective development of test specs
- Allocation of requirements to test specifications & teams
- Ability to write clearly & precisely
- Technical knowledge of how to test

**Outputs** (Deliverables from the process)

- Functional specification baselined
- Test specification base lined
- Expected test results are clearly & accurately documented
- Prerequisites for running the test are documented
- 100% coverage of all allocated requirements
- 100% coverage of all interfaces defined in functional specification

**Customers** (Anyone who receives a deliverable from the process)

- Testers
- Management
- Development teams

**Requirements**

- 100% coverage of all interfaces defined in functional specification

**Outputs**

- Test specification
Test Specification Process Map

Test Specification

Log of Review Findings

Revised Test Specification

Baselined Test Specification

Write Test Specification → Inspect Test Specification → Revise based on Inspection → Baseline

• n- Functional prototype
• nx- DFS
• nx- Author skill and allotted time
• s- Framework Template for Test Specification
• n- Test Plan (allocation of requirements & functionality to this Test Spec.)
• nx- FWS part 2
• n- FWS part 3

• yx- Updated Test Specification
• n- DFS
• n- Test Plan
• s- FrameWork review process
• nx- Reviewers’ skill and allotted time
• c- Moderator’s skill

• y- Test Specification
• y- Log of review findings
• n- Author’s skill and allotted time
• cx- Moderator’s skill and allotted time
• c- Moderator’s skill

INPUTS KEY
s - standard operating procedure
n - noise
c - controllable
y – output from previous step
* - Not implemented in current process
Failure Modes and Effects Analysis

• For each process step or step output, list potential failure modes
• For each failure mode,
  - list potential failure effects,
  - rate the severity of each failure effect, and
  - list the causes of each failure mode.
• Rate the likelihood of each failure mode, effect, and cause combination occurring.
• Assess current controls on each combination.
• Recommend actions for highest risks.
• Select improvements.
• Re-rate risks after improvements.
Test Specification FMEA

• Identified 39 failure modes
• Recommended actions for 28 failure modes
• Majority of the risks controlled by applying prior experience to ensure specification completeness.
  - Distill experience in a checklist.
  - Use different types of experts to review specific parts of a test specification.
• Identified five desirable attributes for test specification authors:
  - Analytical skills (identifying completeness of coverage with minimal redundancy)
  - Communications skills (clarity of instructions)
  - Customer usage knowledge
  - Technical systems knowledge (the architecture and interaction of components)
  - Testing experience
Improvements

• Process revisions
  - Specifications could be written incrementally and a draft could be inspected prior to baselining.
  - Test specifications can be revised and reviewed after execution.

• Checklist with type of expertise required for each item.
  - Needed to ensure the reliability of checklist
    ◆ Is each item interpreted consistently?

  - Measure consistency of checklist usage
    ◆ Have different raters use the checklist on the same specification: independently indicate whether the specification conformed to each item in the checklist.
# Sample of the Test Specification Checklist Items

## General

<table>
<thead>
<tr>
<th>Reviewer</th>
<th>Checklist Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architect</td>
<td>1.1. Does the scope clearly specify the boundaries of the testing covered by this document?</td>
</tr>
<tr>
<td>Test expert</td>
<td>1.3. Does this spec tell the tester where to find all the files necessary to run each test?</td>
</tr>
</tbody>
</table>

## Overall coverage

<table>
<thead>
<tr>
<th>Reviewer</th>
<th>Checklist item</th>
</tr>
</thead>
<tbody>
<tr>
<td>any</td>
<td>2.1 Are all requirements allocated by the test plan to this test team covered by this set of test cases?</td>
</tr>
<tr>
<td>(technical) Domain expert</td>
<td>2.4 Are there test cases with loads to stress the functionality to at least the level of the maximum realistic customer usage?</td>
</tr>
</tbody>
</table>

## Individual test cases

<table>
<thead>
<tr>
<th>Reviewer</th>
<th>Checklist Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test expert</td>
<td>3.1 Are the required files/databases and their location identified?</td>
</tr>
</tbody>
</table>
Nominal classification reliability*

κ (kappa) is defined as the proportion of agreement between raters after agreement by chance has been removed. The formula for κ, with two raters, is:

\[
κ = \frac{P_{\text{observed}} - P_{\text{chance}}}{1 - P_{\text{chance}}}
\]

Where

\( P_{\text{observed}} \) is the proportion of units in which the raters agreed.

\( P_{\text{chance}} \) is the proportion of units in which agreement by chance is expected.

* Reliability estimates the interchangeability of judges by removing random measurement error variance.
Nominal classification reliability

For more than two raters,

\[
K_{overall} = 1 - \frac{nm^2 - \sum_{i=1}^{n} \sum_{j=1}^{k} x_{ij}^2}{nm(m-1) \sum_{j=1}^{k} \overline{p_j q_j}}
\]

Where

- \( x_{ij} \) is the number of ratings of the \( i^{th} \) unit in the \( j^{th} \) category
- \( n \) is the number of units
- \( m \) is the number of raters
- \( k \) is the number of categories
- \( \overline{p} \) = ratings within a category / \( (n \times m) \)
- \( \overline{q} = 1 - \overline{p} \)
### k calculation, first checklist

<table>
<thead>
<tr>
<th>Item</th>
<th>PF</th>
<th>ES</th>
<th>KM</th>
<th>DS</th>
<th>1</th>
<th>0</th>
<th>$\sum_{i=1}^{2} \chi_i^2$</th>
<th>$\bar{\chi}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>1.4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>1.6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>1.7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2.6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>2.7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>

sum: 54 26 244

$n = 20$  
$m = 4$  
$k = 2$

<table>
<thead>
<tr>
<th></th>
<th>$\bar{P}$</th>
<th>$\bar{q}$</th>
<th>$\bar{Pq}$</th>
<th>sum</th>
<th>$K$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.68</td>
<td>0.33</td>
<td>0.22</td>
<td>0.44</td>
<td>0.28</td>
</tr>
</tbody>
</table>
### \( k \) Calculation, Revised Checklist

<table>
<thead>
<tr>
<th>Item</th>
<th>PF</th>
<th>ES</th>
<th>KM</th>
<th>DS</th>
<th>1</th>
<th>0</th>
<th>( \sum_{i=1}^{2} x_i^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>1.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>1.3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>1.4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>1.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>1.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>1.7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>2.1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>2.3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>2.4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>2.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2.6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>2.7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>2.8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>3.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>3.3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>3.4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>3.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>sum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>70</td>
<td>10</td>
<td>308</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n</th>
<th>( \bar{p} )</th>
<th>( \bar{q} )</th>
<th>( \sum )</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.88</td>
<td>0.13</td>
<td>0.22</td>
<td>0.77</td>
</tr>
<tr>
<td>4</td>
<td>0.13</td>
<td>0.88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.11</td>
<td>0.11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Validation and Savings

• Validation
  - Used revised checklist to inspect a test specification that had already been used for testing.
  - Found and corrected specification deficiencies.
  - Used the revised test specification to run additional tests and found three high-priority defects.

• Savings
  - Estimated additional costs to fix defects found in the field.
  - For three defects, additional cost of leaked defects was estimated at $10,100.

• Further validation
  - Used the Test Specification Checklist to re-inspect another test specification.
  - Additional testing with the second revised test specification discovered two more defects at the same time they were being discovered by customers in beta testing.
Components of Savings Calculation

- **Defect management costs**
  - \( \# \text{ defects found} \times (\text{total defect effort} / \# \text{ defects}) \times \text{burdened rate} \)

- **Rework costs**
  - Effort to fix defects found \( \times \text{burdened rate} \)

- **Release costs**
  - **Cost of release \( \times \text{Probability of release due to a high priority defect} \)**
    - Cost of release
      - Management at project and program levels
      - Release management
      - Software configuration management
      - Product and system tests (planning, testing, analysis, and reporting)
      - Media verification and documentation
      - Installation documentation
    - Probability of release is calculated from problem database and release records

- **Less improvement project cost**

- **Additional unmeasured costs avoided**
  - Schedule impact
  - Customer dissatisfaction
  - Contracted customer support
Other uses of $\kappa$ in software development

- Process improvement: classification of process inputs
  - Mapped process across sites, then independently classified inputs to each step. Low $\kappa$ value for critical inputs. Discovered differing perspectives between sites on criticality of inputs (product knowledge and resolution options) to a software rework process.

- Process capability assessment instruments*

- Project planning: test a project risk classification scheme

- Project tracking: test activity/task labels for time charging

- SQA: test a project’s application of documented software processes

- Process reliability: test multiple projects’ interpretation and use of a procedure (projects’ usages are the raters)

- Usability: test usability questionnaire

- Defect management: test defect classification schemes

Conclusions

• Time and energy can be wasted using unreliable instruments due to:
  - Missing or incomplete items
  - Ambiguous items
  - Unclear or meaningless items

• Measuring the reliability of assessment instruments, questionnaires, and nominal categories prior to widespread usage ...
  - can identify problems items in the instrument,
  - provides a basis for improving the instrument,
  - engenders confidence in and encourages use of the instrument, and
  - avoids rework, frustration, and wasted time.
Resources

• David Futrell. 1995. When quality is a matter of taste, use reliability indexes. *Quality Progress* 28: 5 (May), 81-86.
  - This article is a practical guide for applying both the kappa and the intraclass correlation techniques.

The following articles are recommended for further study of $\kappa$ and other interrater agreement measures.

  - Presents the kappa coefficient and discusses its statistical characteristics.

  - Reviews and critiques various approaches to the study of interrater agreement, for which the relevant data comprise either nominal or ordinal categorical ratings from multiple raters.