An Enterprise Framework for Evaluating and Improving Software Quality

Philip Lew
Be Agile...

GO SLOWER TO GO FASTER
Outline

• Introduce quality concepts and quality modeling
• Enterprise quality framework for building quality software products
• Using the framework
Software Everywhere

• Software more and more ubiquitous in our everyday lives.

• Shorter development cycles put pressure on software product quality.

• Business models have changed
  – Pay as you go, pay by subscription

• Cloud and mobile converge

• Behavior and expectations have changed
Software – Drive for Quality

• Abundant research and standards development in the areas of software quality & quality processes.

• Yet, a gap exists in the area of measuring and evaluating the end user’s view on quality as influenced by the ENTIRE organization.

• Development models such as Agile, Spiral, V-model, and others begin at requirements, end with acceptance testing.

• Other parts of an organization and its processes, prior to requirements and after acceptance testing, can significantly influence a customer’s perception of software quality.
Enterprise Quality in Use (EQinU) framework

• Goal is to understand, evaluate, and then improve.

• Flexible framework that can be used in any organization.

• Based on concepts where the outputs of one phase of quality influences the quality at the next phase.
What is Software Quality?

• The meaning of the quality term is not simple and atomic, but a multidimensional and abstract concept.
• Quality can not be measured directly — at least not in a very trivial way
• Common practice assesses quality by means of the quantification of lower abstraction concepts, such as attributes of entities
• Given the complexity that a quality concept involves, **a model** is used in order to specify the quality requirements and understand it better.
Understanding Quality
Quality Models

• Quality models are used assist → to help us to understand and evaluate quality

• One such model that is well known is ISO 25010 - an international standard for software product quality, and system-in-use quality
  – A nice product in the laboratory does not necessarily translate to a high quality product in the eyes of the beholder – the end user (Quality-in-Use)
ISO 25010 – Product Quality

- Breaking down into components and then abstraction to lower levels helps us to understand
- Understanding leads to measurement and evaluation
Quality in Use
Real Situation

- Effectiveness
  - Effectiveness
- Efficiency
  - Efficiency
- Satisfaction
  - Usefulness
  - Trust
  - Pleasure
  - Comfort
- Freedom from risk
  - Economic risk mitigation
  - Health and safety risk mitigation
  - Environmental risk mitigation
- Context coverage
  - Context completeness
  - Flexibility
ISO 25010 Quality Relationship

- Process Quality influences Product Quality
- Process Quality depends on In-Use Measures
- Product Quality influences Quality in Use
- Product Quality depends on In-Use Measures

CMMI

ISO 25010
Previous Work

• Utilizing relationships in different phases to develop quality requirements
• Built relationships

Previous research (2010-2011)
Sample Product Quality Evaluation

<table>
<thead>
<tr>
<th>Usability</th>
<th>Measure</th>
<th>Elementary</th>
<th>Partial</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Understandability</td>
<td></td>
<td></td>
<td>75%</td>
</tr>
<tr>
<td>1.1 Label/icon recognizable</td>
<td>2</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>1.2 Function-task recognizable</td>
<td>4</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>2. Navigation</td>
<td></td>
<td></td>
<td>63%</td>
</tr>
<tr>
<td>2.1 Back-Forth</td>
<td>3</td>
<td>75%</td>
<td></td>
</tr>
<tr>
<td>2.2 Full-mobile</td>
<td>2</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>3. Swiping</td>
<td></td>
<td></td>
<td>63%</td>
</tr>
<tr>
<td>3.1 Consistency</td>
<td>3</td>
<td>75%</td>
<td></td>
</tr>
<tr>
<td>3.2 User Awareness</td>
<td>2</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>4 Learnability</td>
<td></td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td>4.1 Error prevention</td>
<td>1</td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td>4.2 Error detection</td>
<td>1</td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>58%</strong></td>
<td></td>
</tr>
</tbody>
</table>

Through defining the model, then benchmarking with heuristic evaluation, we were able to determine weak areas and make recommendations for significant improvements.
Sample Quality in Use Evaluation

<table>
<thead>
<tr>
<th>Global Evaluation</th>
<th>EI</th>
<th>PI</th>
<th>GI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1. Actual Usability</td>
<td></td>
<td></td>
<td>53.3%</td>
</tr>
<tr>
<td>1.1.1. Effectiveness in use</td>
<td></td>
<td></td>
<td>73.2%</td>
</tr>
<tr>
<td>1.1.1.1. Sub-Task Correctness</td>
<td></td>
<td></td>
<td>86.4%</td>
</tr>
<tr>
<td>1.1.1.2. Sub-Task Completeness</td>
<td></td>
<td></td>
<td>87.9%</td>
</tr>
<tr>
<td>1.1.1.3. Task Successfulness</td>
<td></td>
<td></td>
<td>45.5%</td>
</tr>
<tr>
<td>1.1.1.2. Efficiency in use</td>
<td></td>
<td></td>
<td>29.3%</td>
</tr>
<tr>
<td>1.1.1.2.1. Sub-Task Correctness Efficiency</td>
<td></td>
<td></td>
<td>37.4%</td>
</tr>
<tr>
<td>1.1.1.2.2. Sub-Task Completeness Efficiency</td>
<td></td>
<td></td>
<td>37.5%</td>
</tr>
<tr>
<td>1.1.1.2.3. Task Successfulness Efficiency</td>
<td></td>
<td></td>
<td>13.1%</td>
</tr>
<tr>
<td>1.1.1.3. Learnability in use</td>
<td></td>
<td></td>
<td>57.3%</td>
</tr>
<tr>
<td>1.1.1.3.1. Sub-Task Correctness Learnability</td>
<td></td>
<td></td>
<td>78.8%</td>
</tr>
<tr>
<td>1.1.1.3.2. Sub-Task Completeness Learnability</td>
<td></td>
<td></td>
<td>26.4%</td>
</tr>
<tr>
<td>1.1.1.3.3. Task Successfulness Learnability</td>
<td></td>
<td></td>
<td>66.7%</td>
</tr>
</tbody>
</table>
# 2Q2U – Quality, Quality in Use, Usability and User Experience

<table>
<thead>
<tr>
<th>Related Quality in Use Attribute</th>
<th>Related External quality Related Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learnability in use: Sub-task completeness learnability</td>
<td>1.1.2.2 Learnability.Helpfulness.HelpCompleteness</td>
</tr>
<tr>
<td>Effectiveness in use: Task Successfulness</td>
<td>2.1.1 Information quality.InfoSuitability.Consistency</td>
</tr>
<tr>
<td>Efficiency in use: Sub-task completeness efficiency</td>
<td>1.2.1.2 Ease of use.Controllability.StabilityofMainControls</td>
</tr>
<tr>
<td></td>
<td>1.1.2.2 Learnability.Feedback Suitability.Task Progress Feedback Appropriateness</td>
</tr>
<tr>
<td></td>
<td>1.1.1.3 Learnability.Feedback Suitability.Entry Form Feedback Awareness</td>
</tr>
<tr>
<td></td>
<td>1.1.2.1 Learnability.Helpfulness.Context-sensitve help availability</td>
</tr>
<tr>
<td></td>
<td>1.2.3.1 Ease of use.Data Entry Ease.Defaults</td>
</tr>
<tr>
<td></td>
<td>1.2.3.2 Ease of use.DataEntryEase.MandatoryEntry</td>
</tr>
<tr>
<td></td>
<td>1.2.3.3 EaseofUse.DataEntryEase.ControlAppropriateness</td>
</tr>
<tr>
<td>Effectiveness in use: Sub-task completeness</td>
<td>2.1.1 Information quality.InfoSuitability.Consistency</td>
</tr>
<tr>
<td>Effectiveness in use: Sub-task correctness</td>
<td>1.1.2.2 Learnability.Helpfulness.HelpCompleteness</td>
</tr>
<tr>
<td></td>
<td>1.2.2.1 Ease of use.Error Mgmt.Error Prevention</td>
</tr>
<tr>
<td></td>
<td>1.2.3.1 Ease of use.Data Entry Ease.Defaults</td>
</tr>
<tr>
<td></td>
<td>1.2.3.3 EaseofUse.DataEntryEase.ControlAppropriateness</td>
</tr>
<tr>
<td></td>
<td>2.1.2.2 Information quality.InfoSuitability.InfoCoverage.Completeness</td>
</tr>
<tr>
<td></td>
<td>2.1.2.1 Information quality.InfoSuitability.InfoCoverage.Appropriateness</td>
</tr>
</tbody>
</table>

Previous research (2010-2011)
ISO 9000

• The ISO 9000 [2] family of standards was developed to assist organizations with implementing and operating effective quality management systems.

• ISO 9000 is founded upon eight quality management principles. The principles most applicable to our quality modeling work include:
  – Process approach
  – System approach to management
  – Continual improvement

• ISO 9000 has a broad and general reach and can apply to all organizations striving to increase the quality of their products and services by applying these principles in their operations.
Development Models

- Most development models limited in scope to the product only from requirements to development, and testing.
- Almost all recognize that many defects stem from requirements.
Key Modeling Concepts

Take Aways

• Breakdown concept (quality) to components (levels of abstraction) to understand, evaluate, measure, and then improve
  – Hierarchical requirements tree
  – Measurements for each factor or element (characteristic)

• Relationships and dependency
  – Quality at one phase impacts quality at the next phase

• Current quality models
  – Don’t model outside of development and QA
Building a framework for Enterprise Quality

Components and levels of abstraction

Relationships and dependencies
End User Perception of Quality

• Dependent on good product, of course.
• But also influenced by many other factors outside of development and quality assurance.
• Other parts of the organization influence the quality of not only the product itself, but also the user’s perception of quality.
• Lot of info/research/discussion on metrics but ...
Common Product Lifecycle

Typical Simplified Product Lifecycle

- Sales
- Idea
- Design and Requirements
- Development and Production
- Maintenance and Support
Building a Quality Product or Service

• Using ISO 25010 modeling concepts of influence and dependency
  – Building a quality product or service recognizes that the quality of each phase has an influence on the quality of the next phase AND on the final result
Customer Perceived Quality

Diagram showing the flow from Sales through Product Management (Reqts.), Development, Quality Assurance, Customer Service-Tech Support, to Customer Perceived Quality. The diagram highlights the impact of defects at each stage and the feedback loop from customer interactions and product delivery.
Using the Framework

In progress – Future work
Measuring Quality

- Within each phase, we use the ISO 25010 abstraction model to understand and evaluate quality at that phase
- At the lower levels of abstraction, we can develop quality measures
- The lower level measurements will aggregate to higher cumulative measurements
- Aggregation can be done with various weighting schemes with flexibility according to the business of the organization
Each Component or Phase Has its Own Quality which aggregates and influences overall customer perception of quality.

Possible to have a high quality product, but with poor customer perceived quality! Here’s Why…

- Customer Perceived Quality
- Sales Quality
- Requirements Quality
- Product Development Quality
- Customer Service Quality
Each Component or Phase (Sales) Has its Own Quality Definition (model)

- Customer understanding
- Marketing material
- Sales process
- Customer relationship
- Customer expectations
Defects are introduced at all phases of the lifecycle of a product.

Sales:
- Wrong expectations
- Unfulfilled promises
- Misunderstanding of needs of customer/market
- Example – Website promises unknown to developers
Each Component or Phase (Product management – Requirements) Has its Own Quality Definition
Defects are introduced at all phases of the lifecycle of a product.

Requirements:
- Vague
- Incomplete
- Poorly Written
- Lack of understanding of underlying needs
Each Component or Phase (Development and QA) Has its Own Quality Definition

- Development Quality
  - Code
  - Re-work
  - Adherence to standards
  - Defects
  - Test ware
  - Validation and verification
Defects are introduced at all phases of the lifecycle of a product.

**Development:**
- Coding errors
- Non conformance to corporate coding best practices
- Not extensible
- Poor performance
XBO’s View of the Quality LifeCycle

Goal: Improving Quality

Customer Interactions

Defects are introduced at all phases of the lifecycle of a product.

Quality Assurance:

- No test cases
- Inaccurate test cases, incomplete
- Defects not written understandably
Each Component or Phase (Customer Service-Technical Support) Has its Own Quality Definition
Defects are introduced at all phases of the lifecycle of a product.

Customer Service Representative:
- Not communicating product/customer back
- Lack of good reports
- Not tracking metrics
- Long wait times for customers
- Inaccurate information given to customers
- Can not solve customer problems/answer questions
Evaluate Different Types of Defects

- Customer Service Quality
  - Call quality
  - Service Ticket quality
  - Service reporting
  - Re-calls
Getting Started with Measurement

• What quality attributes are important and fit your organization?
  – Develop a model

• What data can you collect/Which technique can you use
  – Maybe some elements of the model drop out can’t be measured that easily

• Start collecting and developing benchmarks
  – Not absolute numbers but relative or over time
Case Study

• Healthcare industry - EHR
• Company poised for huge growth, but is worried that their product Quality is lacking.
• 50 SW engineers, 50 CSRs and product managers
• Asked XBOSoft for a Quality Process Assessment
Case Study

- A week of onsite and offsite assessment and analysis.
- Discovered immediate problems, underlying problems and root causes.

Examples of problems found:

- Excessive re-work at all phases
- Low customer perception of quality
- Unclear product requirements

Studies show software specialists spend about **40-50%** of their time on avoidable rework rather than on value – added work.

- IEEE Spectrum Magazine
Case Study

• XBO provided company recommendations to solve the issues:

Examples of recommendations

- Standards and best practices
- Inspections THROUGHOUT
- Training program

• XBO delivered a roadmap to implement recommendations and continuously improve
Summary
The Customer’s View of Quality

Modeling Quality for your Organization

Customer Perception of Quality

User Satisfaction
Product Quality
Thanks

Questions and Answers

www.xbosoft.com
408-350-0508

Philip Lew
philip.lew@xbosoft.com