The 'Real Quality Assurance' Manifesto: are you ready for the next generation QA?

Presenter: Tom Gilb, Gilb.com
BCS SIGIST Conference - 'Motivating Testers' –
Thursday 10th December 2010
Keynote 09.30 – 10.30
London
Quality Manifesto/Declaration
(1 page overview, see following slides for detail)

- System Quality can be viewed as a set of quantifiable performance attributes, that describe how well a system performs for stakeholders, under defined conditions, and at a given time.
- System Stakeholders judge past, present, and future quality levels; in relationship to own their perceived needs/values.
- System Engineers can analyze necessary, and desirable, quality levels; and plan, and manage to deliver, a set of those quality levels, within given constraints, and available resources.
- Quality Management is responsible for prioritizing the use of resources, to give a satisfactory fit, for the prioritized levels of quality: and for trying to manage the delivery of a set of qualities - that maximize value for cost - to defined stakeholders.
1. System Quality

- can be viewed as a set of quantifiable ‘performance attributes’,
- that describe how well a system performs for stakeholders,
  - under defined conditions, and at a given time.
Multiple Required Performance and Cost Attributes are the basis for architecture selection and evaluation
2. System Stakeholders

- judge
  - past, present, and future quality levels;
  - in relationship to own their perceived needs/values.
Stakeholder Map

Figure 1: A Stakeholder Map for the Library Loans project

Copyright The Atlantic Systems Guild, Used with Kind Permission.

Suzanne Robertson & James Robertson
3. System Engineers

• can analyze
  – necessary, and desirable, quality levels;
  – and plan, and manage to deliver,
    • a set of those quality levels,
    • within given constraints,
      – and available resources.

Example of an Impact Estimation table

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Designs</th>
<th>An Organic Apple</th>
<th>An Organic Orange</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Performance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eater Acceptance</td>
<td>70%</td>
<td>85% (80%)</td>
<td></td>
</tr>
<tr>
<td>From 50% to 80% of People</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pesticide Measurement</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Reduce from 0.005% to 0.001%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shelf-Life</td>
<td>70%</td>
<td>200% (100%)</td>
<td></td>
</tr>
<tr>
<td>Increase from 1 week to 1 month</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vitamin C from fruit</td>
<td>20%</td>
<td>140% (100%)</td>
<td></td>
</tr>
<tr>
<td>Increase from 50mg to 100mg per day</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carbohydrates from fruit</td>
<td>40%</td>
<td>30%</td>
<td></td>
</tr>
<tr>
<td>Increase from 50g to 100g per day</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sum of Performance</td>
<td>300%</td>
<td>410%</td>
<td></td>
</tr>
</tbody>
</table>

- Costs
  Relative Cost (Local currency)

| Performance to Cost Ratio          | 909                      | 1025             |

or ‘Benefit to Cost Ratio’

Lindsey Brodie
Figure 1: Real (NON-CONFIDENTIAL version) example of an initial draft of setting the objectives that engineering processes must meet.
Strategy Impact Estimation:
for a $100,000,000 Organizational Improvement Investment

Objectives

Defined
In earlier slide

"Benefits"

Technical Strategies

Strategy Impacts on Objectives

Cost

358!
4. Quality Management

• is responsible for prioritizing the use of resources,
• to give a satisfactory fit,
• for the prioritized levels of quality:
• and for trying to manage the delivery of a set of qualities -
• that maximize value for cost - to defined stakeholders.
Stakeholders:
How to find out about, and confirm, their requirements (a process of managing quality iteratively)

1. Identify all critical and profitable STAKE-HOLDERS
2. Identify All critical and profitable stakeholder REQUIREMENTS
3. Detail and clarify requirements (Scale + Benchmarks + Targets)
4. Validate and agree these requirements with stakeholders
5. Select most profitable requirements to deliver first (Evolutionary delivery)
6. Learn new requirements evolutionarily as result of experience feedback and time (new technology, markets and cost levels)
My Quality Principles (overview)


1. Quality Design: Ambitious Quality Levels are designed in, not tested in. This applies to work processes and work products.

2. Software Environment: “Software” Quality is totally dependent on its resident system quality, and does not exist alone; ‘software qualities’ are dependent on a defined system’s qualities – including stakeholder perceptions and values.

3. Quality Entropy: Existing or planned quality levels will deteriorate in time, under the pressure of other prioritized requirements, and through lack of persistent attention.

4. Quality Management: Quality levels can be systematically managed to support a given quality policy. Example: “Value for money first”, or “Most competitive World Class Quality Levels”.

5. Quality Engineering: A set of quality levels can be technically engineered, to meet stakeholder ambitions, within defined constraints, and priorities.

6. Quality Perception: Quality is in the eyes of the beholder: objective system quality levels may be simultaneously valued as great for some stakeholders, and terrible for others.

7. Design Impact on Quality: any system design component, whatever its intent, will likely have unpredictable main effects, and side effects, on many other quality levels, many constraints, and many resources.

8. Real Design Impacts: you cannot be sure of the totality of effects, of a design for quality, on a system, except by measuring them in practice; and even then, you cannot be sure the measure is general, or will persist.

9. Design Independence: Quality levels can be measured, and specified, independently of the means (or designs) needed to achieve them

10. Complex Qualities: many qualities are best defined as a subjective, but useful, set of elementary quality dimensions; this depends on the degree of control you want over the separate quality dimensions.
1. Quality Design Principle

- **Ambitious Quality** Levels are designed in, not tested in.
- **This applies to work processes and work products.**

Testers Note: Testing cannot deliver quality!
EVO Plan Confrmit 8.5

4 product areas were attacked in all: 25 Qualities concurrently, one quarter of a year. Total development staff = 13

<table>
<thead>
<tr>
<th>Current Status</th>
<th>Improvements</th>
<th>Reportal - E-SAT features</th>
<th>Survey Engine .NET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Units</td>
<td>Units</td>
<td>%</td>
<td>Past</td>
</tr>
<tr>
<td>75.0</td>
<td>25.0</td>
<td>62.5</td>
<td>56</td>
</tr>
<tr>
<td>14.0</td>
<td>14.0</td>
<td>100.0</td>
<td>0</td>
</tr>
<tr>
<td>15.0</td>
<td>15.0</td>
<td>107.1</td>
<td>8</td>
</tr>
<tr>
<td>5.0</td>
<td>5.0</td>
<td>95.2</td>
<td>2</td>
</tr>
<tr>
<td>3.0</td>
<td>2.0</td>
<td>66.7</td>
<td>1</td>
</tr>
<tr>
<td>1.0</td>
<td>2.0</td>
<td>95.7</td>
<td>0</td>
</tr>
<tr>
<td>4.0</td>
<td>5.0</td>
<td>100.0</td>
<td>5</td>
</tr>
<tr>
<td>1.0</td>
<td>12.0</td>
<td>150.0</td>
<td>15</td>
</tr>
<tr>
<td>1.0</td>
<td>14.0</td>
<td>109.0</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Development resources</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Current Status</th>
<th>Improvements</th>
<th>Reportal - MR Features</th>
<th>XML Web Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Units</td>
<td>Units</td>
<td>%</td>
<td>Past</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>59.0</td>
<td>14</td>
</tr>
<tr>
<td>25.0</td>
<td>45.0</td>
<td>112.5</td>
<td>8</td>
</tr>
<tr>
<td>4.4</td>
<td>4.4</td>
<td>38.7</td>
<td>3</td>
</tr>
<tr>
<td>101.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

| Development resources | 0 | 0 |

Trond Johansen
Value Management

Management Cycle (about 1-3 weeks)

Development Cycle (about 1-3 weeks)

Stakeholder Vision  Prioritization  Product Vision  Prioritization

Verify Product Vision  Verify Stakeholder Vision

Value Management  Scrum  Value Management
# Value Decision Tables

<table>
<thead>
<tr>
<th>Stakeholder Value</th>
<th>Stakeholder Value 1</th>
<th>Stakeholder Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Value 1</td>
<td>-10%</td>
<td>40%</td>
</tr>
<tr>
<td>Business Value 2</td>
<td>50%</td>
<td>10%</td>
</tr>
<tr>
<td>Resources</td>
<td>20%</td>
<td>10%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stakeholder Value</th>
<th>Product Value 1</th>
<th>Find.Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stakeholder Value 1</td>
<td>-10%</td>
<td>50%</td>
</tr>
<tr>
<td>Stakeholder Value 2</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>Resources</td>
<td>2%</td>
<td>5%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solution 1</th>
<th>Service Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find.Fast</td>
<td>-10%</td>
</tr>
<tr>
<td>Product Value 2</td>
<td>50%</td>
</tr>
<tr>
<td>Resources</td>
<td>1%</td>
</tr>
</tbody>
</table>

**Prioritized List**

1. Service Guide
2. Solution 9
3. Solution 7

**Scrum Develop**

We measure improvements

Learn and Repeat
2. Software Environment Principle

- “Software” Quality is
  - totally dependent on its resident system quality,
    - and does not exist alone;
  - ‘software’ qualities’ are dependent
  - on a defined system’s qualities – (think, hardware)
    - including stakeholder perceptions and values

Testers Note: Software Testing cannot measure system qualities
Component Quality depends on Related Components
3. **Quality Entropy** Principle

- Existing or planned quality levels will
  - *deteriorate* in time,
  - under the pressure of other prioritized requirements,
  - and through lack of persistent attention

**Testers Note:** Software Testing cannot measure future long term qualities, they change.
Engineering “Maintainability”: Green Week
Weekly ‘Refactoring’ at Confirmit

<table>
<thead>
<tr>
<th>Current Status</th>
<th>Improvement</th>
<th>Goals</th>
<th>Step 6 (week 14)</th>
<th>Step 7 (week 15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Units</td>
<td>100.0</td>
<td>100.0</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>Speed</td>
<td>100.0</td>
<td>100.0</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>Maintainability Doc.Code</td>
<td>100.0</td>
<td>100.0</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>Interviewer.Console</td>
<td>100.0</td>
<td>100.0</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>NilUnitTests</td>
<td>0.0</td>
<td>0.0</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>PeerTests</td>
<td>100.0</td>
<td>100.0</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>FeCap</td>
<td>0.0</td>
<td>10.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TestDirectorTests</td>
<td>100.0</td>
<td>100.0</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>Robustness.Correctness</td>
<td>2.0</td>
<td>2.0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Robustness.BoundaryConditions</td>
<td>0.0</td>
<td>0.0</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>Speed</td>
<td>0.0</td>
<td>0.0</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>ResourceUsage.CPU</td>
<td>100.0</td>
<td>100.0</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>Maintainability.Doc.Code</td>
<td>100.0</td>
<td>100.0</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>SynchronizationStatus</td>
<td>100.0</td>
<td>100.0</td>
<td>80</td>
<td>80</td>
</tr>
</tbody>
</table>

Speed
Maintainability
Nunit Tests
PeerTests
TestDirectorTests
Robustness.Correctness
Robustness.Boundary Conditions
ResourceUsage.CPU
Maintainability.DocCode
SynchronizationStatus

POT-SHOTS — Brilliant Thoughts in 17 words or less

SOMETHING'S WRONG WITH MY LIFE ~
SHOULD I TRY TO FIX IT, OR WAIT UNTIL I GET ANOTHER?

December 15, 2009
© Ashleigh Brilliant  www.ashleighbrilliant.com
Multiple Required Performance and Cost Attributes are the basis for architecture selection and evaluation
4. Quality Management Principle

- Quality levels can be systematically managed
  - to support a given quality policy.

- Example:
  - “Value for money first”,
  - or
  - “Most competitive World Class Quality Levels”.

Testers Note: Software Testing cannot manage multiple quality/performance/cost levels.
Gilb’s Value Manifesto: A Management Policy?

1. **Really useful value, for real stakeholders** will be defined measurably. No nice-sounding emotive words please.

2. **Value** will be seen in light of total long term costs as a decent return on investment.

3. **Powerful management devices**, like motivation and follow-up, will make sure the value for money is really delivered – or that the failure is punished, and the success is rewarded.

4. **The value will be delivered evolutionarily** – not all at the end.

5. **That is**, we will create a stream of prioritized value delivery to stakeholders, at the **beginning** of our value delivery projects; and continue as long as the real return on investment is suitably large.

6. **The CEO is primarily responsible for making all this happen effectively.**
   1. **The CFO** will be charged with tracking all value to cost progress.
   2. **The CTO and CIO** will be charged with formulating all their efforts in terms of measurable value for resources.

---

Source “Value Delivery in Systems Engineering” available at www.gilb.com

December 15, 2009 © Gilb.com
The Value Principles:

1. Value can always be articulated quantitatively, so that we can understand it, agree to it, track it, contract for it and understand it in relation to costs.
2. Value is a result, delivered to a real set of stakeholders.
3. Value must be seen in light of lifetime total cost aspects, and must be as profitable as alternative investments.
4. Value occurs through time, as a stakeholder experience: It is not delivered when a system to enable it is delivered – only when that system is successfully used to extract the value.
5. Value can be delivered early, and for part of one stakeholder’s domain. This proves the value potential, and actually improves the real organization.
6. There is never a really sufficient reason to put off value delivery until large-scale long-term investments are made. This is just a common excuse from the many weak, Ignorant, cowards who would like to spend a lot of money before being held to account.
7. People who cannot deliver a little value early, in practice, cannot be entrusted to deliver a lot of value for a larger investment.
8. The top management must be primarily responsible for making value delivery happen in their organization. The specialist managers will never in practice take the responsibility, unless they are aiming to take over the top job.
9. Value is a multiplicity of improvements, and certainly not all related to money or savings – but we still need to quantify the value proposition in order to understand it, and manage it.
10. If we prioritize highest value for money first, then we should normally experience an immediate and continuous flow of dramatic results, that the entire organization can value and relate to. Be deeply suspicious of long-term visions with no short-term proof.


“Value Delivery in Systems Engineering”
5. Quality Engineering Principle

• A *set* of quality levels can be technically engineered,
  – to meet stakeholder ambitions,
  – within defined constraints, and priorities.

*Testing Note: testing cannot find and evaluate designs to deliver quality levels*
How do we evaluate, for a single quality dimension, the impact of a design?

- We must estimate
- (or measure)
- the numeric cumulative impact
- of the design
  - on a defined Scale (units),
  - using a defined Meter (test process),
  - with respect to requirement levels.
6. Quality Perception Principle

• Quality is in the eyes of the beholder:
  – objective system quality levels may be simultaneously valued as
    • great for some stakeholders,
    • and terrible for others.

Testing Note: testing cannot decide on right quality levels for stakeholders
Usability level good enough for ‘Management’ is not necessarily good enough for the ‘Operator’
Design Impact on Quality Principle

- any system design component,
  - whatever its intent,
  - will likely have unpredictable main effects,
    - and side effects,
    - on many other quality levels,
    - many constraints, and
    - many resources.

Testing Note: testing can give us measurable feedback on current incremental levels of quality.
## DoDef. Persinscom Impact Estimation Table:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements</strong></td>
<td>50% 10% 5% 5% 5% 60%</td>
<td>185%</td>
<td>265%</td>
<td>130%</td>
<td>180%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Availability</td>
<td>50% 5% 5–10% 0% 0% 200%</td>
<td>265%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90% &lt;&gt; 99.5% Up time</td>
<td>50% 5–10% 5–10% 50% 0% 10%</td>
<td>130%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td>50% 10% 90% 25% 5% 50%</td>
<td>180%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200 &lt;&gt; 60 Requests by Users</td>
<td>50% 10% 90% 25% 5% 50%</td>
<td>180%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responsiveness</td>
<td>45% 100% 10% 15% 53% 303%</td>
<td>251%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70% &lt;&gt; ECP's on time</td>
<td>45% 100% 10% 15% 53% 303%</td>
<td>251%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Productivity</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:1 Return on Investment</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Morale</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72 &lt;&gt; 60 per month on Sick Leave</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Integrity</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>88% &lt;&gt; 97% Data Error %</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technology Adaptability</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75% Adapt Technology</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement Adaptability</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>? &lt;&gt; 2.6% Adapt to Change</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource Adaptability</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1M &lt;&gt; ? Resource Change</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost Reduction</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FADS &lt;&gt; 30% Total Funding</td>
<td>42% 10% 25% 5% 70% 25%</td>
<td>177%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sum of Performance</strong></td>
<td>482% 280% 305% 390% 315% 649%</td>
<td>649%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Money % of total budget</td>
<td>15% 4% 3% 4% 6% 4%</td>
<td>36%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time % total work months/year</td>
<td>15% 15% 20% 10% 20% 18%</td>
<td>98%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sum of Costs</strong></td>
<td>30 19 23 14 26 22</td>
<td>120%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8. Real Design Impacts Principle: (tricky) (so you need frequent feedback!)

- you cannot be sure of the totality of effects, of a design for quality,
  - on a system,
    - except by measuring them in practice;
    - and even then, you cannot be sure the measure is general,
    - or will persist.

Test Note: Quality Tests must be frequent, to discover problems early
Real Relevant Measurement Frequently for Multiple Critical Factors, Is the only reliable way to measure and control many simultaneous effects of incremental design implementation

- 2 Kinds of Feedback from Stakeholders, when value increment is *really* exploited in practice after delivery.
- Combined with other information from the relevant environment. Like budget, deadline, technology, politics, laws, marketing changes.
The Simplest and Best Agile Project Method; ‘Evo’

- **Process Description**
  - 1. Gather from all the key stakeholders the top few (5 to 20) most critical goals that the project needs to deliver.
    - Give each goal a reference name (a tag).
  - 2. For each goal, define a scale of measure and a ‘final’ goal level.
    - For example: **Reliable**: Scale: **Mean Time Before Failure**, Goal: **1 month**.
  - 3. Define approximately 4 budgets for your most limited resources
    - (for example, time, people, money, and equipment).
  - 4. Write up these plans for the goals and budgets
    - *(Try to ensure this is kept to only one page).*
  - 5. Negotiate with the key stakeholders to formally agree the goals and budgets.
  - 6. Plan to deliver some benefit
    - (that is, progress towards the goals)
    - in **weekly** (or shorter) increments (Evo steps).
  - 7. Implement the project in Evo steps.
    - **Report** to project sponsors after each Evo step (weekly, or shorter) with your best available estimates or measures, for each performance goal and each resource budget.
    - **On a single page**, summarize the **progress to date** towards achieving the goals and the costs incurred.
  - 8. When all Goals are reached: ‘Claim success and move on’
    - a. Free remaining resources for more profitable ventures
9. Design Independence Principle

• Quality levels can be
  – measured,
  – and specified,
  – independently of the means (or designs) needed to achieve them

Test Note: testers of critical qualities must expect numeric specifications to test against.

What does the Lord say about specification and measurement?
THE PRINCIPLE OF 'QUALITY QUANTIFICATION'

All qualities can be expressed quantitatively, 'qualitative' does *not* mean unmeasurable.

"In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it.

I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be."

*Lord Kelvin, 1893*

From [http://zapatopi.net/kelvin/quotes.html](http://zapatopi.net/kelvin/quotes.html)
Google any Metric you need

**Gilb Measurability Principle**
Now the accuracy of this measurement is an attribute, X1, of the system. ... the complexity of a particular unit of software: Understandability and Testability. ... My performance metric is the number of peer-reviewed papers published. ...
c2.com/cgi/wiki?GilbMeasurabilityPrinciple - Cached - Similar

**Software Testing**
Software Testing is the process of executing a program or system with the intent of .... for testability is also an important design rule for software development. ..... Also testing can be served as a metric for software reliability. ...
www.ece.cmu.edu/~koopman/das_s69/sw_testing/ - Cached - Similar

**[PDF] Improving testability of object-oriented systems**
File Format: PDF/Adobe Acrobat

by S Jungmayr - 2004 - Cited by 6 - Related articles

**Metric System - Math Diagnostic Test**
Metric System - Math Diagnostic Test. Quiz. Show all questions. <-- -->. The metre is a measure of ... In a metric ton, there are ...
www.pedagonet.com/maths/metric.htm - Cached - Similar

**Emerald FullText Article : Measuring the quality of computer ...**
For each quality attribute and variable, a metric or series of metrics can be ..... Testability is a factor affecting the effort required to test a system, .... Gilb, T. (1977), Software Metrics, Winthrop Publishers, Cambridge, MA, ... xtra.emeraldinsight.com/Insight/viewContentItem.do?... - Similar
10. Complex Qualities Principle

- many qualities are best defined as a *subjective*, but *useful*, set of elementary quality dimensions;
- this depends on the degree of control you want over the *separate* quality dimensions.

Testers Note: every one of the defined sub-dimensions should be testable economically in practice frequently
Quantifying ‘Usability’ (ErieyeC&C System)

SIMPLIFIED PLANNING LANGUAGE: ‘PLANGUAGE’

**QUALITY**

- **USABILITY**
  - **Intuitiveness**
    - Ambition: Great intuitive capability
    - SCALE: Probability that intuitive guess right.
    - METER: <100 observations.>
    - PAST [GRAPES] 80% <-LN
    - RECORD [MAC] 9%?<-TG
    - Fail [TRAINED, RARE] 50-90%
    - Goal [TASKS] 99% <-LN

- **AVAILABILITY**

- **ADAPTABILITY**

- **WORK-CAPACITY**

- **INTELLIGIBILITY**

  - **Intelligibility**
    - Ambition: Super ease of immediate understanding
    - SCALE: % OK interpretations.
    - METER: 10 ops., 100 infos, 15 mins.
    - P: PAST [20 ops., 300 info, 30 min.] 99%
    - RECORD [P] 99.0%
    - Fail [DELIVERY[1]] 99.0% <- MAB
    - ACCEPTANCE 99.5%
    - Goal [M1] 99.9% <- LN

**DEFINED:**
- TRAINED: C&C operator, approved course, 200 hours duration.
- RARE: types of tasks performed less than once a week per op.
- TASKS: onboard operator distinct tasks carried out.
- ACCEPTANCE: formal acceptance testing via customer contract.
- DELIVERY: Evolutionary delivery cycle, integrated and useful.
A Generic Set of Performance measures – including several related to ‘change’
References

Any rule, if it is to be practicable, must be simple in administration.
8 June 1991

Dear Tom,

You may quote paragraphs from my book, *OUT OF THE CRISIS* (MIT, CAES, 1986). I know that you will be careful not to quote out of context. I send best greetings, remaining

Sincerely yours,

W. Edwards Deming

To Dr. Tom Gilb
Holters vei 2
N-1410 Kolbotn
Norway
Inspect Early, Don’t Test

- <Testing> is .... unreliable, costly, ineffective.
- It is important to carry out <QC> at the right point for minimum cost.
- You cannot <QC> quality into a product <-Harold F. Dodge
Details
Tom Gilb

- Tom Gilb (born 1940, California) has lived in UK since 1956, and Norway since 1958.
- He has taught and consulted world-wide for decades, including having direct corporate methods-change influence at major corporations such as Intel, HP, IBM, Nokia.
- He has had documented his founding influence in Agile Culture, especially with the key common idea of iterative development.
- He coined the term 'Software Metrics' with his 1976 book of that title.
- He is co-author with Dorothy Graham of the static testing method book 'Software Inspection' (1993).
- He is known for his stimulating and advanced presentations, and for consistently avoiding the oversimplified pop culture that regularly entices immature programmers to waste time and fail on their projects.
- More detail at www.Gilb.com
Questions: now, briefly
After lecture
By Email: tomsgilb at gmail.com

Copy of these slides will be on SIGIST conference site
And in Downloads: www.gilb.com
– Library